

## Capítulo

# 1

## Desenvolvendo ChatBots com o Dialogflow

Joeckson Correa, Davi Viana, Ariel Teles

### *Abstract*

*ChatBots are increasingly present in people's lives, whether in the financial area, customer service, education, and also in the health area. ChatBots bring benefits, for example, to the healthcare area, being able to assist patients and healthcare professionals in administrative tasks of appointment scheduling, in the area of education, and can answer students' questions about re-enrollment. This book chapter aims to present the ChatBot concepts, the origin of the term ChatBot and its evolution and architecture, and how Machine Learning and Natural Language Processing are used in ChatBots. In particular, it introduces the Dialogflow platform. Dialogflow is a web platform that facilitates the ChatBot development process. By using Dialogflow, the developer can create ChatBots that respond to users through text or speech. Finally, this chapter presents the step by step of how to create a ChatBot using Dialogflow to schedule medical appointments.*

### *Resumo*

*Os ChatBots estão cada vez mais presentes na vida das pessoas, seja na área financeira, atendimento ao cliente, educação, e também na área da saúde. Os ChatBots trazem benefícios, por exemplo, para área da saúde, podendo auxiliar pacientes e profissionais de saúde em tarefas administrativas de agendamento de consulta, na área da educação, podendo tirar dúvidas de alunos sobre matrícula. Esse capítulo de livro tem como objetivo apresentar os conceitos de ChatBots, a origem do termo ChatBot e sua evolução, arquitetura de um ChatBot, e de que forma o Aprendizado de Máquina e o Processamento de Linguagem Natural são utilizados em ChatBots. Em particular, ele apresenta a plataforma Dialogflow. O Dialogflow é uma plataforma web que facilita a criação de ChatBots. Ao utilizar o Dialogflow, o desenvolvedor pode criar ChatBots que possam responder aos usuários através de texto ou fala. Ao final, o capítulo apresenta o passo a passo de como criar um ChatBot, utilizando o Dialogflow, para agendamento de consultas médica.*

## 1.1. Introdução

Um ChatBot é um agente de conversação que usa Processamento de Linguagem Natural (PLN) para se comunicar com os usuários [Pérez-Soler et al. 2021]. Existem vários tipos de ChatBots sendo utilizados em diversos domínios. Este capítulo explora os conceitos, as tecnologias, as possibilidades de uso dos ChatBots em vários domínios de aplicação, e apresenta uma prática com a plataforma de desenvolvimento de ChatBots Dialogflow.

### 1.1.1. Origem do termo ChatBot

O conceito de ChatBot, ou robô de conversação, não é novo [Weizenbaum 1966]. Nos últimos anos, os ChatBots foram aplicados a diversos domínios, tais como saúde, educação, área financeira, varejo, dentre outras. Na década de 1950, Alan Turing publicou o artigo *Computing Machinery and Intelligence* [TURING 1950], que adquiriu grande notoriedade, em que foi apresentada e discutida a questão “As máquinas seriam capazes de pensar?”. Para responder esse questionamento, Alan Turing desenvolveu um método, chamado Teste de Turing, com a finalidade de identificar a capacidade de uma máquina em ter o comportamento semelhante a um comportamento humano. Esse teste impulsionou os primeiros passos para o surgimento dos ChatBots. O Teste de Turing definia o computador como inteligente quando ele conseguisse dialogar com uma pessoa através de um *chat*, sem que ela percebesse estar conversando com uma máquina.

Os primeiros Chatbots foram projetados para imitar o comportamento humano, como a “ELIZA ” [Weizenbaum 1966], em uma conversa baseada em textos, realizando ações específicas e delimitadas dentro de um escopo controlado. O ChatBot ELIZA é um robô conversacional com a finalidade de simular um psicólogo virtual, o qual utiliza a reformulação de trechos das frases que são capturados das entradas dos usuários, fazendo parecer que possui um vasto vocabulário. Tendo em vista ser a primeira tentativa de criar um software que pudesse passar no Teste de Turing, ELIZA é considerada a “mãe dos ChatBots” [Weizenbaum 1966].

O PARRY [Leptourgos and Corlett 2020] foi construído pelo psiquiatra americano Kenneth Colby em 1972. Este ChatBot tinha a finalidade de imitar o comportamento de um paciente com esquizofrenia. O PARRY tinha a característica de parecer estar adotando falsas crenças de ser assediado, subjugado, perseguido, acusado, maltratado, injustiçado, atormentado e depreciado. Enquanto ELIZA foi considerada uma simulação de um terapeuta, PARRY simulava um paciente esquizofrênico. No *script* utilizado pelo PARRY durante o diálogo, foram desenvolvidos diferentes tipos de comportamentos que simulavam o diálogo de forma mais profunda. As conversas não eram baseadas em perguntas e respostas, mas de forma fluida.

Outro ChatBot é ALICE, um acrônimo que significa *Artificial Linguistic Internet Computer Entity*. A primeira edição de ALICE<sup>1</sup> foi implementada em 1995 e o programa ganhou o “Loebner Prize” nos anos de 2000, 2001 e 2004. ALICE possui uma base de conhecimento constituída por centenas de fatos, citações e ideias de seu criador. Esse ChatBot apresenta um vocabulário de mais de 5.000 palavras, sendo programado para dar muitas informações a respeito do serviço que disponibiliza. O projeto ALICE<sup>2</sup>

---

<sup>1</sup><https://alicebot.org>

<sup>2</sup><https://www.pandorabots.com/pandora/pics/wallaceaimltutorial.html>

é composto por uma base de conhecimento desenvolvida na linguagem AIML (acrônimo para Linguagem de Marcação de Inteligência Artificial) que descreve o comportamento do ChatBot, e também por um módulo chamado interpretador, que manipula a base de conhecimento e é disponibilizado em diferentes linguagens de programação. O projeto ALICE é disponibilizado através da licença pública GNU e dividido em três módulos: especificações técnicas da linguagem AIML, conjunto de interpretadores AIML, e a base de conhecimento [Kraus 2007].

### 1.1.2. Conceitos

Os Chatbots [Softić et al. 2021] são soluções de software que podem interagir com seres humanos por meio de uma interface de *chat*, podendo ser em forma textual ou por voz. Eles são também conhecidos como *talkbots*, *smartbots*, *bots*, *chatterbots* ou agentes conversacionais. Os ChatBots tendem a se comunicar com o usuário e se comportar como um ser humano. Para isso ocorrer, precisam estar conectados a serviços de mensagem (e.g., Facebook Messenger, Telegram, WhatsApp), páginas web ou aplicativos móveis. Os ChatBots são classificados em dois tipos: baseados em regras e baseados em Inteligência Artificial (IA), apresentados a seguir.

### 1.1.3. Arquitetura de ChatBots

A arquitetura de um ChatBot baseado em IA, conforme mostra a Figura 1.1, pode ser dividida em três subsistemas: pré-processamento, classificação de intenção, e utilização de contexto e geração de respostas. Inicialmente, o usuário envia uma mensagem de entrada que a entende e produz uma resposta. O pré-processamento executa um conjunto de operações na mensagem para operações adicionais, tais como, tokenização, normalização, eliminação de palavras irrelevantes, identificação das partes do discurso e entidades, stemming, lemmatisation [Karve et al. 2018], conceitos explicados a seguir.

A classificação de intenção tem a finalidade de identificar a intenção da mensagem do usuário. As intenções são rótulos que representam um significado das mensagens de entrada dos usuários. Por exemplo, a mensagem do usuário “Olá, bom dia!” pode ser classificada como “intenção de saudação”. Em relação a utilização de contexto e geração de respostas, tem-se a implementação da pilha para tratamento de contextos. Toda resposta está associada a um determinado contexto. Assim, quando uma resposta é gerada, o contexto correspondente é enviado para a pilha. A qualquer momento, a parte superior da pilha indicará o contexto atual da conversa. Esse contexto atual é utilizado como um filtro para selecionar a resposta correspondente à intenção da mensagem do usuário. Caso nenhuma resposta corresponda ao filtro de contexto, a pilha de contexto é exibida para recuperar o contexto anterior. Então as respostas são pesquisadas com o contexto como filtro. Esse processo é repetido até que uma resposta apropriada seja encontrada [Bocklisch et al. 2017].

### 1.1.4. ChatBot baseado em Regras

A abordagem baseada em regras utiliza o conceito de máquina de estado, a qual consiste em regras que determinam o conjunto de entrada necessário para a transição de um estado para outro [Mellado-Silva et al. 2020]. Isso significa que o ChatBot conduz o usuário com perguntas para chegar à resolução correta. As estruturas das perguntas e respostas

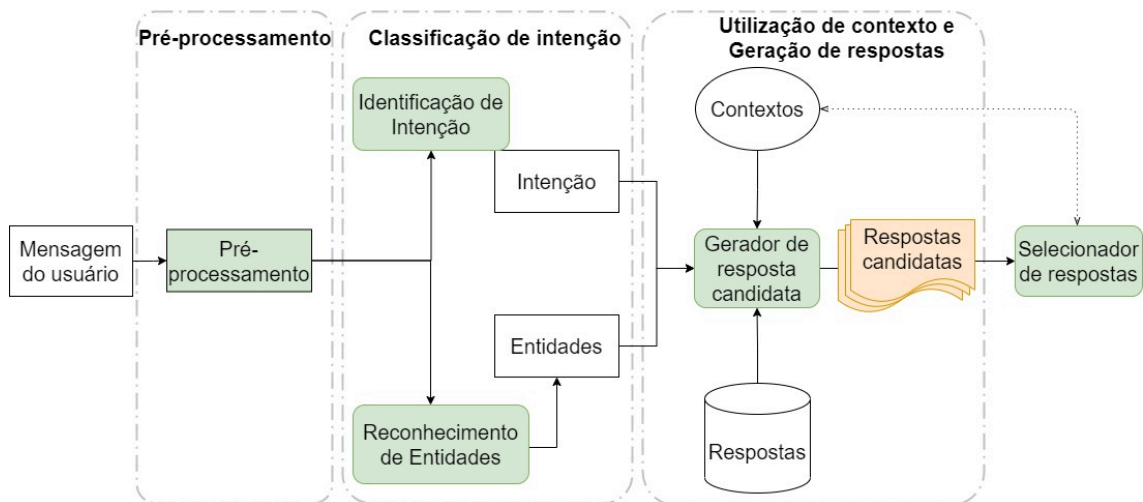


Figura 1.1. Arquitetura de um ChatBot. Adaptado de [Bocklisch et al. 2017].

são pré-definidas para que o ChatBot tenha o controle da conversa. Os ChatBots baseados em regras podem ser importantes para situações em que o objetivo de sua utilização seja pra realizar tarefas específicas, como requisitar segunda via de boleto ou acompanhar a situação de uma encomenda. Nesses casos, não se faz necessário a utilização de abordagens que fazem uso de IA. Os ChatBots baseados em regras realizam ações pré-estabelecidas, deixando claro suas limitações por não terem autonomia suficiente para resolver problemas mais elaborados [Kar and Haldar 2016a].

Como exemplo, considere um Chatbot baseado em regras que seja criado para uma instituição de ensino. Quando um usuário entra em contato com um ChatBot, é comum que ele responda com uma mensagem automática, por exemplo “Em que posso te ajudar?”. Podem ocorrer situações de buscar saber qual é o período de matrícula, documentação para rematrícula ou informação a respeito do contato telefônico da biblioteca. Caso o usuário responda algo diferente do que foi anteriormente sugerido pelo ChatBot, tal como “Quero saber minha nota de Matemática”, o ChatBot pode não conseguir entender a solicitação e provavelmente repetirá a mensagem automática.

### 1.1.5. ChatBot baseado em Inteligência Artificial

Os ChatBots baseados em IA [Kar and Haldar 2016b] são capazes de entender a linguagem natural e não somente os comandos predefinidos, desenvolvem inteligência à medida que interagem com os usuários. Além disso, eles conseguem manter diferentes contextos de conversas e fornecer ao usuário conversas mais ricas e engajadas. Em razão disso, surgiram conceitos de agentes virtuais e técnicas de reconhecimento de fala utilizadas em agentes virtuais, como por exemplo, Apple Siri, Amazon Alexa e Google Assistant. Certos elementos são extremamente importantes para o funcionamento dos ChatBots baseados em IA. Esses elementos são os classificadores de textos, algoritmos de aprendizado de máquina, redes neurais artificiais e processamento de linguagem natural. Como exemplo, os Chatbots baseados em IA podem realizar uma reserva de hotel após seguir solicitações do usuário ou realizar uma comparação de preços de reserva de hotel de vários sites ao mesmo tempo.

## 1.2. Processamento de Linguagem Natural

Nesta seção é abordado o PLN. É explicado como os ChatBots utilizam os principais componentes do PLN, tais como preparação dos dados, reconhecimento de entidades nomeadas, classificação de entidades nomeadas, compreensão de linguagem natural, extração de entidades e classificação de intenções.

### 1.2.1. Conceitos de Linguagem Natural

O PLN surgiu devido à necessidade de compreensão e comunicação automática do ser humano com o computador. O PLN é um mecanismo desenvolvido para extrair informações de textos, facilitar a entrada de dados nos sistemas e a estruturação de dados [Santos et al. 2014]. O PLN é uma área da Ciência da Computação e da Linguística que estuda os métodos formais para analisar textos e gerar frases em um idioma, através do uso de software [Aranha 2017].

Existem quatro etapas no PLN [BULEGON and MORO 2010]: análises morfológica, sintática, semântica e pragmática. A análise morfológica é o estudo de cada palavra presente no texto de forma independente. Essa análise tem a responsabilidade de definir artigos, substantivos, verbos e adjetivos armazenados em um dicionário de palavras com significados semelhantes, dentro de um domínio específico de conhecimento. A análise sintática faz uso do dicionário, primeiramente identificando possíveis relações entre as palavras. Em um segundo momento, ela identifica o sujeito, predicado, complementos nominais e verbais, adjuntos e apostos. Na análise semântica, ocorrem as relações dos termos ambíguos, de sufixos e afixos, ou seja, questões sobre significados associados aos componentes de uma palavra, o sentido real da frase ou palavra. Para a junção e visualização de todas as etapas, a análise pragmática faz a conexão de todo o processo e mostra visualmente o resultado.

### 1.2.2. Pré-processamento

Para modelar o texto apresentado pelo usuário, possibilitando que a máquina o entenda, é necessário o seu pré-processamento, o qual abstrai e estrutura do texto, deixando apenas o que é informação relevante. Esse pré-processamento reduz o vocabulário do texto e torna os dados menos esparsos, uma característica importante para o processamento computacional [Aranha 2017].

A **tokenização** (do inglês, *tokenization*) é o início do pré-processamento de texto. Ela ocorre quando uma sequência de caracteres é dividida e quando for delimitada por espaço em branco, vírgula, ponto ou outro delimitador. Cada divisão estabelecida é chamada de *token*.

O processo de **remoção de stopwords** envolve a eliminação de palavras que não devem ser mantidas no texto. As *stopwords* [Morais and Ambrósio 2007] são conhecidas por serem palavras não importantes na análise do texto. As preposições, pronomes, artigos, advérbios, e outras classes de palavras auxiliares são geralmente classificadas como *stopwords*.

A **normalização morfológica** é uma forma de aumentar a memorização dos diversos significados de um mesmo conceito. Ela visa evitar a repetição das representações de

uma palavra a um mesmo conceito [BULEGON and MORO 2010]. Por exemplo, do conceito de “objeto físico que consiste em um número de páginas atadas juntamente” tem-se a palavra “livro” com as seguintes representações “livro” e “livros”. O processo de normalização propõe que essas duas formas sejam agrupadas em apenas uma, indicando que elas têm o mesmo significado.

*Stemming* faz parte do processo de normalização, podendo ser classificado por radicalização inflexional ou radicalização para a raiz. A radicalização inflexional tem como característica a utilização das flexões verbais, fazendo truncamentos que tornam as palavras, na maioria das vezes, de difícil compreensão. Por exemplo, “livro”, “livros”, “livreto” são substituídas pelo radical da palavra “livr” [BULEGON and MORO 2010].

A *lemmatization* faz parte do processo de normalização. Ele tem o objetivo de fazer a substituição das diversas formas de representação da palavra pela forma primitiva. As formas “livro”, “livros” e “livraria” são todas transformadas para sua forma primitiva “livro” [BULEGON and MORO 2010].

### 1.2.3. Reconhecimento de Entidades Nomeadas

As entidades são utilizadas para representar pessoas, lugares, instituições, acontecimento, tempo, dentre outras. Entretanto, para reconhecer essas entidades, é necessário o reconhecimento dos objetos no texto [BULEGON and MORO 2010]. O Reconhecimento de Entidades Nomeadas (REN) tem como objetivo identificar as entidades nomeadas e classificá-las em categorias pré-definidas, tais como: Pessoa, Organização, Local [Light 1998]. Por exemplo:

“Gabriel Barbosa reside em São Luis e estuda na UFMA (Universidade Federal do Maranhão)”

Realizando o REN do exemplo, temos: [Gabriel Barbosa], [São Luis], [UFMA] e [Universidade Federal do Maranhão], respectivamente, entidades cujas categorias são: Pessoa, Local, Organização e Organização.

### 1.2.4. Compreensão de Linguagem Natural

A Compreensão de Linguagem Natural (em inglês, *Natural Language Understanding*) é uma subárea do PLN que usa a análise sintática e semântica do texto ou da fala para classificar o significado de uma frase. A sintaxe se refere à estrutura gramatical de uma frase, enquanto a semântica diz respeito ao seu significado. Na arquitetura de ChatBots, a NLU é um dos principais componentes, torna o ChatBot completo e reduz as chances do ChatBot não compreender o que o usuário deseja. As principais funcionalidade da NLU em um ChatBot são a classificação de intenções e extração das entidades [Bocklisch et al. 2017].

A intenção é uma determinada mensagem do usuário, aquilo que o usuário está tentando transmitir ou realizar durante a conversa. Por exemplo, se um usuário deseja consultar a previsão do tempo, ele pode dizer: “consultar a previsão do tempo”, ou pode simplesmente escrever “previsão do tempo”. Em ambos os casos, o ChatBot não vai se concentrar na frase completa. A ideia é que o ChatBot possa identificar qual a intenção e, assim, consiga dar uma resposta que faça sentido ao usuário [Gupta et al. 2021].

As entidades<sup>3</sup> são palavras-chave extraídas de uma mensagem do usuário. Por exemplo, o número do telefone. As entidades são os elementos sobre os quais a intenção está se referindo. Por exemplo, na frase “Quero um suco de uva”, a entidade é a palavra “uva”.

### 1.3. Aprendizado de Máquina

O AM é uma subárea da IA que permite criar programas de computador com a capacidade de aprender e executar tarefas [Forgy 1965]. O que torna o programa capaz de aprender por si só, usando um conjunto de dados que passam a representar experiências passadas. O AM é composto por diferentes áreas de pesquisa, tais como IA, probabilidade e estatística, teoria da complexidade computacional, teoria da informação, filosofia, psicologia, neurobiologia, dentre outros. Tem-se como tarefas do AM a classificação, regressão, agrupamento de dados, previsão de séries temporais, dentre outras. A utilização de AM tem crescido na construção de modelos para resolução de problemas em diversos domínios de aplicação, como visão computacional, reconhecimento de fala, e compreensão de texto [von Rueden et al. 2021].

Como exemplo de solução baseada em AM, considere um programa de computador que deve executar uma tarefa simples, como distinguir entre três variedades diferentes de flor de uma mesma espécie. Em vez de codificar um programa utilizando todo o conhecimento acerca das variedades da flor em questão, algumas características botânicas das três flores são apresentadas a um programa. Ele implementa um algoritmo de AM, que, através de um processo de treinamento, aprende a caracterizar uma flor baseado em suas características. Assim como os seres humanos aprendem a diferenciar as variedades de flores observando suas características, o programa de AM também pode aprender a realizar essa tarefa por meio da análise das características.

Existem tarefas descritivas e tarefas preditivas em AM. Nas tarefas descritivas, busca-se o desenvolvimento de algoritmos e modelos (i.e., um algoritmo treinado) que descreve os dados. Entre as tarefas descritivas, uma das principais é o agrupamento de dados [Forgy 1965], que busca separar os dados de maneira que dados semelhantes fiquem em um mesmo grupo. Um exemplo da aplicação de agrupamento de dados é o agrupamento de textos. Nesse caso, o algoritmo procura agrupar textos que abordem o mesmo assunto e separar em grupos diferentes os textos que abordam assuntos diferentes.

As tarefas preditivas podem ser divididas em tarefas de classificação e regressão. Nas tarefas de classificação, busca-se atribuir categorias predefinidas a exemplos de entrada. Por exemplo, um banco pode desenvolver um sistema para a classificação de seus clientes em duas categorias para fornecimento de empréstimo: SIM e NÃO. Por meio do histórico de crédito dos clientes, e também de dados como salário e tempo de emprego, o sistema pode aprender a distinguir os clientes para os quais o banco deve (SIM) ou não deve (NÃO) fornecer um empréstimo. Assim, tem-se um sistema de recomendação de crédito, cujas categorias a serem preditas para um novo cliente são SIM e NÃO (atributo de saída). Nas tarefas de regressão, objetiva-se prever o valor de uma variável numérica (atributo de saída, i.e., variável dependente), dadas outras variáveis (atributos de entrada, i.e., variáveis independentes). Assim, em vez de encontrar uma classe associada, como

---

<sup>3</sup><https://rasa.com/docs/rasa/nlu-training-data/>

na classificação, deve-se encontrar uma função que mapeie um exemplo para um número.

O AM pode ser utilizado em ChatBots para ajudar a diagnosticar doenças. Em [\[Mathew et al. 2019\]](#) é proposto um ChatBot que utiliza um modelo de AM baseado no algoritmo *K Nearest Neighbor* (KNN). Esse modelo é treinado com um conjunto de dados de doenças e sintomas. Durante a conversa entre o ChatBot e o usuário, os sintomas são identificados pelo ChatBot, o qual consegue reconhecer a doença e recomenda o tratamento adequado.

### 1.3.1. Classificação de Intenções

A classificação de intenções é a categorização de forma automática de dados de texto com base nos objetivos do usuário. Um classificador de intenções analisa os textos e os categoriza em intenções. Isso é útil para entender as intenções por trás das mensagens do usuário, automatizar processos e obter informações úteis. Toda interação com o usuário tem um propósito, objetivo ou intenção. Quer eles queiram, por exemplo, fazer uma compra, solicitar mais informações, ou cancelar a assinatura de internet [\[Ali 2020\]](#).

A classificação de intenção pode usar AM e PLN para fazer a associação de palavras ou expressões de forma automática a uma determinada intenção do usuário. Por exemplo, um modelo de AM pode realizar um agendamento de uma consulta médica através das sentenças, “quero consultar” ou “consulta médica” que estão associadas à intenção “consultas paciente”. Para que isso ocorra, o classificador de intenções precisa ser treinado com dados de exemplos, conhecidos como dados de treinamento ou frases de treinamento. Por exemplo, se o ChatBot for destinado a realizar tarefas de vendas de lanches, deve-se escolher sentenças como: “quero comprar lanche” ou “quero lanche”. Após a definição das *tags*, pode ser dado início ao processo de treinamento do classificador de intenções, repassando os exemplos de textos ou frases de treinamento para cada sentença. Por exemplo, “Quero comprar um suco de goiaba”.

### 1.3.2. Deep Learning e ChatBots

*Deep Learning* (DL), ou Aprendizado Profundo, é uma sub-área de ML. Mais especificamente, trata de Redes Neurais Artificiais, uma área que busca simular computacionalmente o cérebro enquanto máquina de aprendizado [\[Pacheco and Pereira 2018\]](#). Atualmente é uma área de pesquisa extremamente ativa, que pode ser aplicada em diversas áreas, tais como reconhecimento de fala, visão computacional, e também ChatBots. Atualmente existem ChatBots baseados em DL que podem ser divididos em duas categorias: baseados no modelo de recuperação, e gerativo [\[Csaky 2019\]](#). Essa subseção será focada em ChatBots no modelo gerativo.

O ChatBot gerativo é uma abordagem mais inovadora, e foi possibilitada pelo surgimento do DL. Os modelos gerativos foram desenvolvidos para resolver problemas em que o ChatBot não consegue responder uma resposta predefinida. ChatBots gerativos podem trabalhar com casos novos de mensagens de usuários porque não precisam contar com respostas predefinidas. Para que isso ocorra, o modelo cria sua própria resposta, através da transformação dos dados de entrada do usuário utilizando a multiplicação de matrizes e funções não lineares que podem conter milhões de parâmetros. Os modelos gerativos baseados em DL conseguem dar aos usuários a sensação de es-



tar conversando com um ser humano real, por não haver respostas predefinidas. Esses modelos precisam de uma grande base de dados para aprenderem a construir suas respostas [Nguyen and Shcherbakov 2018]. O modelo *Seq2seq* [Csaky 2019] é um dos principais modelos de DL utilizados na abordagem de construção de ChatBots gerativos. A rede *Seq2seq* é um modelo de rede neural recorrente (do inglês, *Recurrent Neural Network* - RNNs) que tem sido bastante utilizada em diversas tarefas de PLN, como a tradução e a sumarização de textos.

## 1.4. Exemplos de ChatBots

### 1.4.1. SERMO

O SERMO [Denecke et al. 2021] é um ChatBot com funcionalidades que realizam intervenções baseadas em Terapia Cognitivo Comportamental (TCC). O SERMO se destaca dos demais ChatBots por utilizar técnicas de PLN e métodos de análise de emoções dos usuários a partir do diálogo. O SERMO fornece quatro módulos:

- Módulo de interação, que realiza o diálogo com o usuário para coletar informações sobre alguma situação que impactou o usuário, bem como a emoção associada à situação;
- Módulo de fornecimento de atividades e exercícios que disponibilizam aos usuários uma lista de atividades prazerosas divididas em três categorias: exercícios de *mindfulness*, exercícios de relaxamento e atividades de lazer;
- Módulo diário de eventos com emoções associadas, que registra o humor por meio de um controle deslizante com cinco *smileys* diferentes, de bons a ruins. Além da possibilidade de registrar vários humores em um dia;
- Módulo de fornecimento de informação disponibiliza explicações sobre as funcionalidades do SERMO, os princípios básicos da TCC e as emoções: medo, nojo, raiva, alegria, tristeza, culpa e vergonha.

### 1.4.2. Jamura

O Chatbot Jamura [Salvi et al. 2019] foi proposto para fornecer a capacidade de monitorar e controlar dispositivos domésticos conectados à Internet e responder a perguntas sobre a casa e o jardim do usuário. As informações da residência (e.g., temperatura, umidade, intensidade de luz, estado operacional das luzes, o estado operacional dos ventiladores em cada cômodo, a umidade do solo do jardim, estado da bomba d'água) podem ser acessadas e controladas por meio de comandos de linguagem natural usando o Jamura. O sistema de automação residencial proposto usa os conceitos da Internet das Coisas (IoT) para tornar os serviços, dispositivos e dados relacionados transparentes e acessíveis. O Jamura foi construído utilizando o Dialogflow e disponibilizado na plataforma Telegram.

### 1.4.3. EMMA

EMMA [Ghandeharioun et al. 2019] é um Chatbot pessoal de bem-estar emocional inteligente e tem o objetivo de fornecer sugestões de bem-estar, através de atividades relevantes

individuais ou sociais, que se enquadram em uma das categorias de psicoterapia, sendo psicologia positiva, intervenções cognitivo-comportamentais, metacognitivas ou somáticas. Como exemplo de atividades tem-se “Divulgar a alegria ligando para um amigo e transmitindo sua energia positiva” ou “Escrever um comentário positivo para a boa postagem de algum amigo”.

## 1.5. Plataformas de Desenvolvimento de ChatBots

Atualmente existem diversas plataformas de desenvolvimento de ChatBots. Por exemplo, *Wit.ai* que é uma plataforma que implementa uma “Interface de Programação de Aplicações” (API), com instância pública e privada sem limites de requisições e pertence ao “Facebook” [Agarwala et al. 2019]. O *Botpress* que é plataforma de código aberto para criar ChatBots de alta qualidade, considerada entre os desenvolvedores a melhor plataforma para criar ChatBots em português [Sabharwal et al. 2019]. O *Botkit* que é um *framework* de código aberto escrito em *JavaScript*, com ele é possível fazer integração com diversas plataformas de mensagens, como *Slack*, *Messenger*, *Hangouts*, *Twilio*, *Webex* e *Facebook*, fornece também integração com API de PLN<sup>4</sup>. Nesta seção será dada uma visão geral nas plataformas “RASA” e “Dialogflow”.

### 1.5.1. Rasa

O RASA [Mellado-Silva et al. 2020] é uma plataforma de desenvolvimento de ChatBots de código aberto, baseada em PLN e AM. Ela é comumente utilizada na comunidade de pesquisa, embora não ofereça uma infraestrutura em nuvem, como hospedagem gerenciada e escalabilidade. A plataforma possui dois componentes principais (RASA NLU e RASA Core) e ambos podem ser utilizados separadamente.

#### 1.5.1.1. RASA NLU

O RASA NLU [Bocklisch et al. 2017] é responsável pelo PLN, fazendo o reconhecimento da entrada do usuário, categorizando a entidade e classificando a intenção. A plataforma dá flexibilidade para utilizar diferentes analisadores de linguagem natural que são baseados nas bibliotecas *spaCy*, *NLTK*, *CoreNLP*, e *BERT*.

As *pipelines* definem as ordens e as ações a serem executadas para o ChatBots entenderem corretamente as entradas dos usuários. O RASA, por ser de código aberto, oferece diversas formas de configurar suas *pipelines*, como, por exemplo, adicionando ou removendo etapas de processamento. A plataforma recomenda a utilização de duas *pipelines*: *spaCy* e *Tensorflow* [Bocklisch et al. 2017]. Uma das principais diferenças entre elas é o uso de vetores de palavras pré-treinadas pelo *spaCy*. Por exemplo, os dados de treinamento contêm a palavra “vinho” categorizada como bebida, mas não “champanhe”. Caso a entrada do usuário seja “Eu quero champanhe”, o *spaCy* consegue reconhecer que “vinho” e “champanhe” são palavras semelhantes. Isso faz com que o RASA tenha uma maior confiança de que champanhe é uma bebida, tendo como resultado o direcionamento do usuário para a intenção correta.

---

<sup>4</sup><https://botkit.ai/docs/v4/>

### 1.5.1.2. RASA Core

O RASA Core [Bocklisch et al. 2017] é responsável por receber todas as intenções e entidades do usuário (i.e., saída do RASA NLU ou outra ferramenta de extração de entidades e classificação de intenções), e direciona uma ação a ser realizada através de um modelo de AM. Ele leva em consideração o histórico das conversas e os dados de treinamento. A plataforma atribui uma pontuação a todas as ações possíveis e executa aquela com maior valor.

A Figura 1.2 mostra a arquitetura de alto nível do RASA Core e, em seguida, ela é explicada.

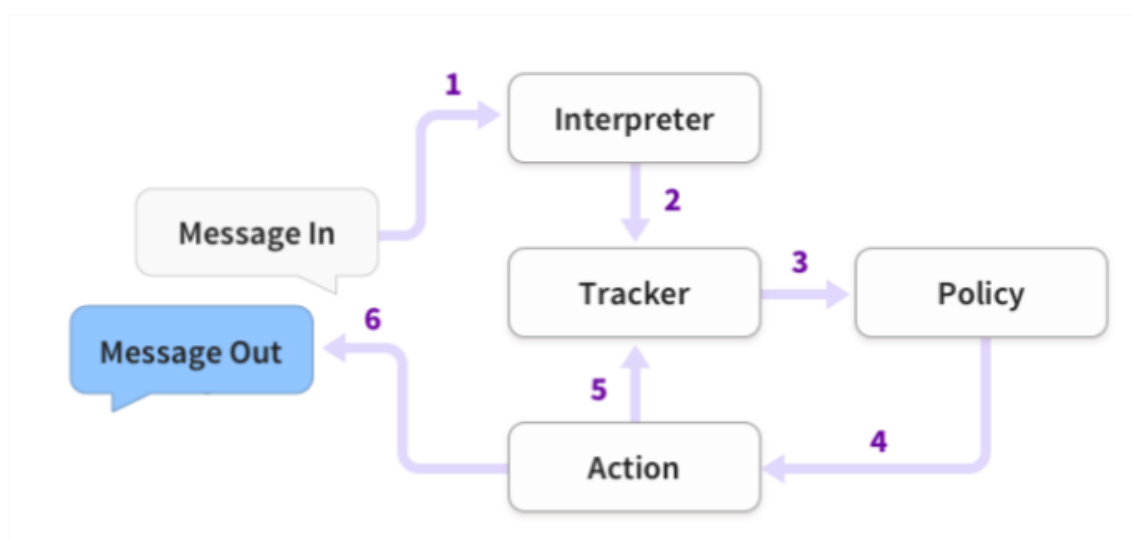


Figura 1.2. Arquitetura de alto nível do RASA Core [Bocklisch et al. 2017].

1. As mensagens de entrada do usuário são recebidas e repassadas para o módulo de interpretação (*Interpreter*), que converte todo o conteúdo em um dicionário constituído por texto original, intenções e entidades. O RASA Core não consegue interpretar as entradas dos usuário, logo é necessário um modelo de NLU (RASA NLU) para analisar as entradas.
2. O *Tracker* gerencia o estado da conversa com um usuário. Ele tem a responsabilidade de receber uma notificação caso uma nova mensagem da conversa chegue, armazenar o estado da conversa com um único usuário, e recuperar a mensagem da conversa armazenada.
3. A *Policy* recebe o estado atual do *Tracker*.
4. A *Policy* define qual a próxima ação a ser tomada nas etapas da conversa.
5. A *Action* escolhida é registrada pelo *Tracker*.
6. Por fim, ocorre o envio da resposta ao usuário.

### 1.5.2. Dialogflow

O Dialogflow é uma plataforma de PLN que facilita o design e a integração de uma interface do usuário conversacional com aplicações para dispositivos móveis, aplicações web, dispositivos, bots, sistemas interativos de resposta de voz, dentre outras.

O Dialogflow é uma plataforma de desenvolvimento de ChatBots que faz utilização de PLN. A plataforma suporta algumas tecnologias, linguagem de programação e bibliotecas, tais como Android, iOS, Webkit HTML5, JavaScript, Node.js, e Python. Os principais conceitos para o entendimento da plataforma Dialogflow são: agentes, intenções, entidades, contextos e *fullfilment* [Maldonado and Cuadra 2019a].

### 1.5.3. Agentes

Os agentes são assistentes virtuais que administram conversas com usuários finais. O agente pode ser considerado um módulo de PLN que entende as entradas dos usuários e pode ser integrado a outras aplicações ou canais de atendimento, tais como Facebook, WhatsApp e Telegram [Maldonado and Cuadra 2019a]. O Dialogflow estrutura um conjunto de intenções, as quais são cruciais para a definição do mapeamento de entradas para um conjunto de uma ou mais ações de respostas correspondentes.

### 1.5.4. Intenções

As intenções [Maldonado and Cuadra 2019b] servem para fazer um mapeamento entre a entrada do usuário e qual ação o ChatBot deve realizar. A intenção é composta por quatro componentes principais [Maldonado and Cuadra 2019b]: frases de treinamento, resposta, *ações* e *parâmetros*.

- **Frases de treinamento.** Para melhorar a forma de reconhecimento das entradas dos usuários, podem ser criadas frases de treinamento pré-definidas. Dessa forma, o ChatBot consegue entender as entradas dos usuários utilizando técnicas de PLN [Muhammad et al. 2020a].
- **Respostas.** São as possíveis saídas que o ChatBot pode retornar para o usuário, podendo ser textos pré-definidos ou ações processadas em serviços externos [Maldonado and Cuadra 2019a], por exemplo, consultar a previsão do tempo ou consultar a cotação do dólar.
- **ções e Parâmetros.** O Dialogflow não consegue trabalhar apenas com a correspondência de palavras no sentido literal fornecidas pelo usuário. É então necessária a utilização de intenções que utilizem entidades, tanto para entrada dos usuários quando na resposta dos ChatBots. Ao definir entidades nas frases de treinamento, a plataforma, automaticamente, transforma-as em *parâmetro* de uma *ação*.

### 1.5.5. Contextos

Os contextos são strings que representam o contexto atual da conversa do usuário. Eles são úteis para diferenciar frases que podem ser ambíguas e ter significados diferentes dependendo do que foi escrito anteriormente [Muhammad et al. 2020b].

### 1.5.6. *Fullfilment*

O Dialogflow permite utilizar dois tipos de resposta aos usuários: de forma estática ou processada através do componente chamado *fullfilment*. O *fullfilment* é uma lógica opcional usada pela plataforma para retornar respostas mais apropriadas e inteligentes de acordo com os dados extraídos na intenção do usuário [Muhammad et al. 2020b].

## 1.6. Desenvolvendo um ChatBot com o Dialogflow

Esta seção apresenta, em formato de tutorial passo a passo, o processo de criação de um ChatBot no Dialogflow para agendamento de consulta. A versão detalhada do desenvolvimento do ChatBot pode ser encontrada em: <https://github.com/josantosc/enucompi2021-minicurso-chatbot>.

### 1.6.1. Fluxo de Conversa

Para que o ChatBot consiga ter um diálogo eficiente com o usuário, é preciso projetar o fluxo do diálogo cuidadosamente. Para criar um fluxo de conversa, o primeiro passo é determinar o tema geral da conversa, depois determinar cada uma de suas cenas, bem como o personagem para ilustrar cada cena. Por último, criar um exemplo de conversa com a resposta esperada [Muhammad et al. 2020b].

O fluxo de conversa (Figura 1.3) é utilizado para desenvolver o ChatBot de agendamento de consulta. As intenções (representadas na cor cinza) utilizadas no fluxo são baseadas em correspondência de padrões. Assim, é disponibilizado a opção de utilizar um menu. Os menus estão disponíveis no momento que o usuário fornece uma entrada (representada na cor verde) durante o decorrer da conversa. O menu disponibilizado ao usuário tem o formato numérico para a seleção dos conteúdos desejados. Embora as entradas do usuário esperem uma resposta numérica, também existe a utilização de textos.

### 1.6.2. Acesso ao Dialogflow

O Dialogflow fornece ao usuário uma interface web chamada *Console Dialogflow*. Este console é utilizado para criar agentes, intenções, entidades e contextos. Para criar uma conta no Dialogflow é requerido uma conta do Google. Portanto, para realizar login na plataforma, caso não tenha uma conta, é necessário criar. Siga os seguintes passos para acessar a plataforma:

1. Acesse <https://Dialogflow.cloud.google.com/login>;
2. Clicar no botão *Sign-in with Google*;
3. Selecione a conta de acesso.

### 1.6.3. Criando o Primeiro Agente

Para criar o agente, é preciso seguir os passos abaixo:

1. Acessando o Console do Dialogflow, clique em “Criar Agente” (Figura 1.4) no menu à esquerda;

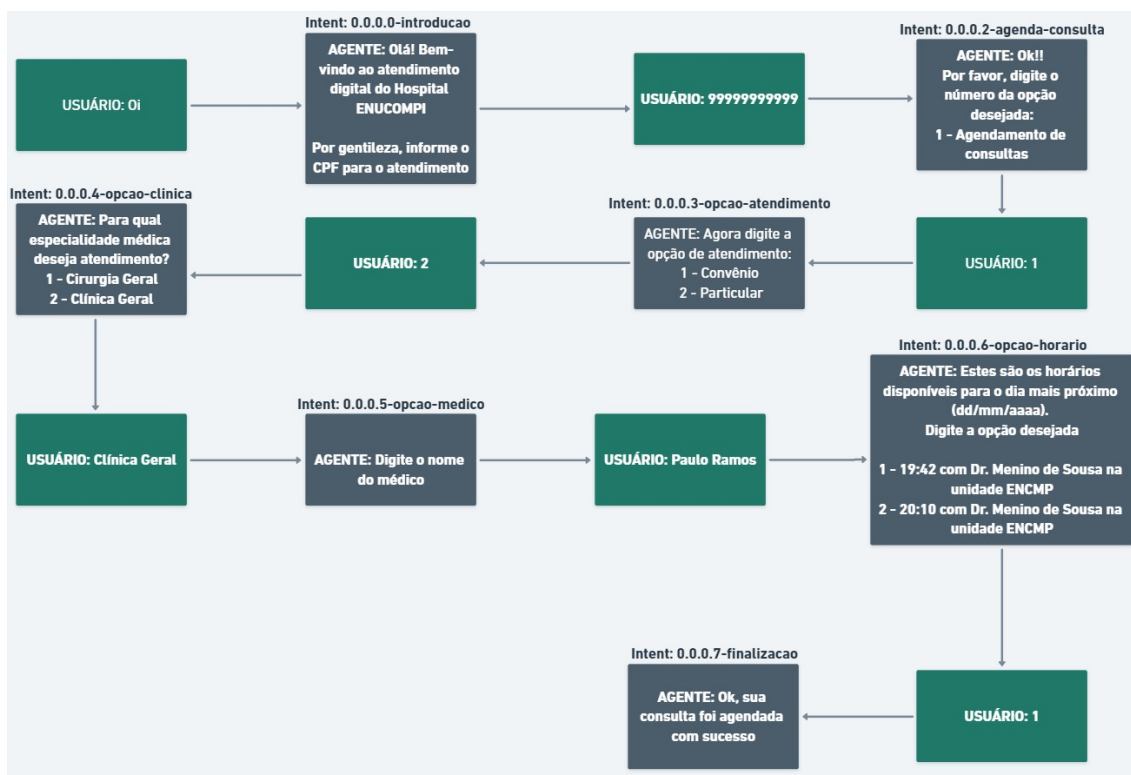


Figura 1.3. Fluxo de conversa de agendamento de consulta.

2. Digite o nome do agente, o idioma e fuso horário padrão;
3. Depois clique no botão Criar (Figura 1.5). É possível alterar a edição do agente após a criação.

#### 1.6.4. Criação de Intenções

Nesta etapa, serão criadas as intenções do ChatBot. Para isso, siga os seguintes passos, considerando a Figura 1.6

1. Clique no botão de adição + ao lado de *Intents* no menu da barra lateral à esquerda;
2. Insira um nome para a *intent*. O nome da *intent* precisa representar as expressões de usuário final que ela reconhece. Para o ChatBot que está sendo criado, a primeira intenção será chamada de “0.0.0.0-introducao”;

Siga os passos a seguir para adicionar frases de treinamento, considerando a Figura 1.7).

1. Na seção *Training phrases*, clique em *Add user expression*;
2. Digite as frases de treinamento correspondentes a intenção em questão, e pressione “Enter” após cada entrada.

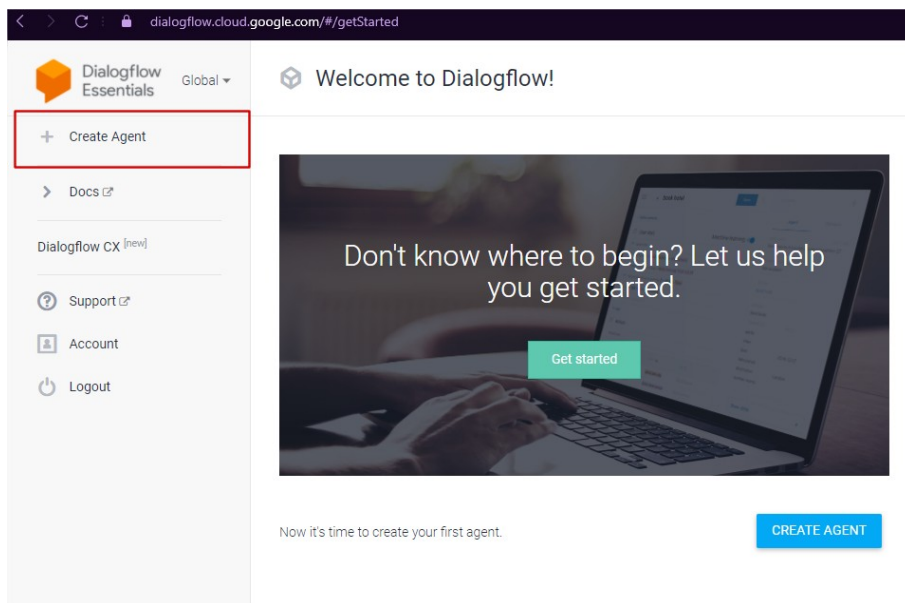


Figura 1.4. Tela inicial do Dialogflow para criar agente.

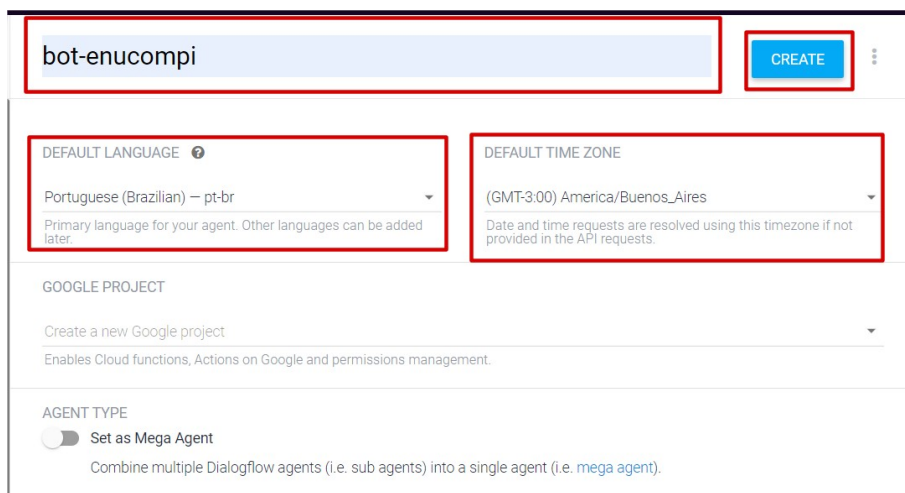


Figura 1.5. Criando agente no Dialogflow.

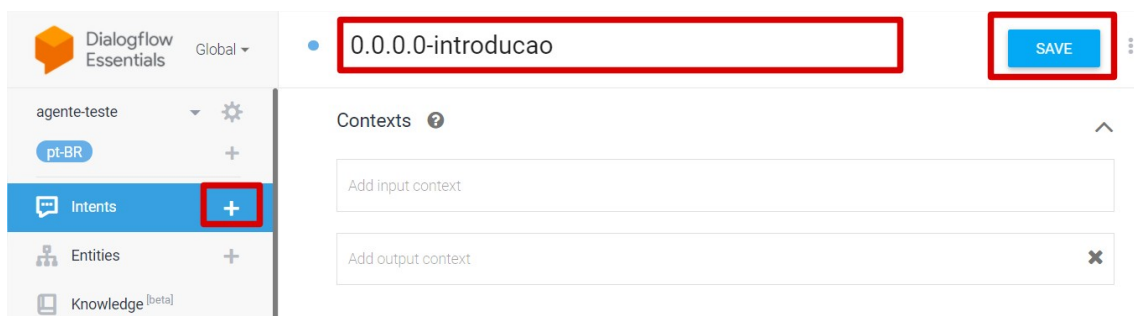
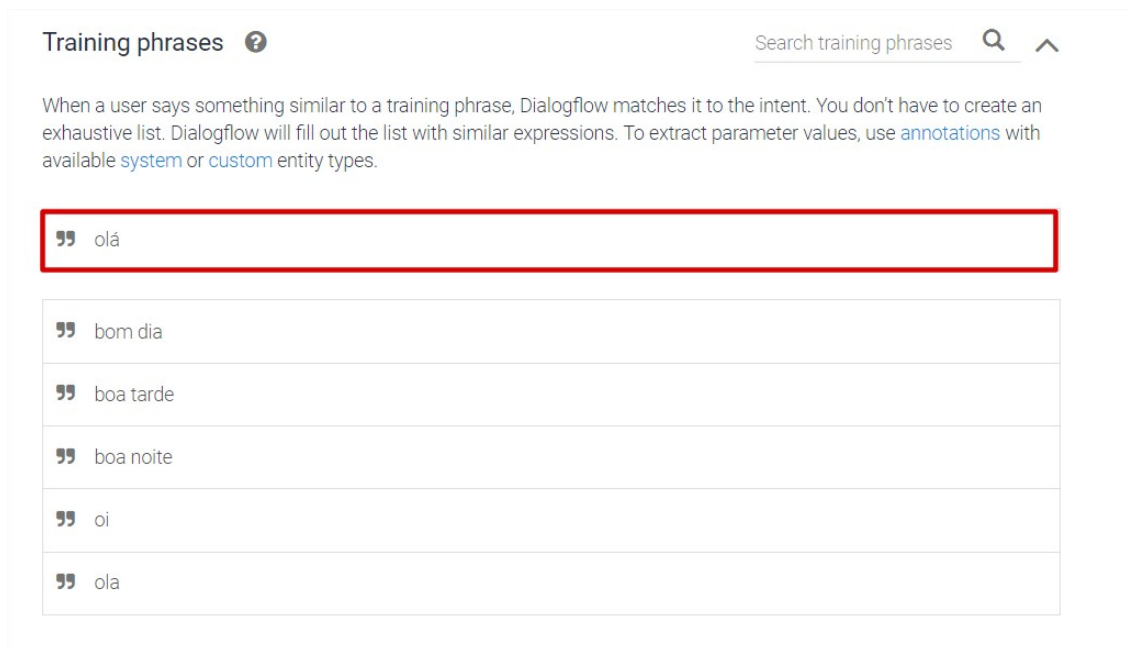


Figura 1.6. Criação de intenção no Dialogflow.



**Figura 1.7. Adicionando frases de treinamento no Dialogflow.**

A seguir são apresentados os passos para adicionar respostas às intenções, conforme a Figura 1.8.

1. Na seção *Responses*, digite a resposta da intenção na seção *Text Response*;
2. Para adicionar mais de uma resposta, clique no botão *ADD RESPONSES*, e adicione as respostas desejadas;
3. Após o preenchimento das respostas, clique em *Save*.

O processo de criação de intenção precisa ser feito para todas as intenções do fluxo apresentando na Figura 1.3.

### 1.6.5. Contextos

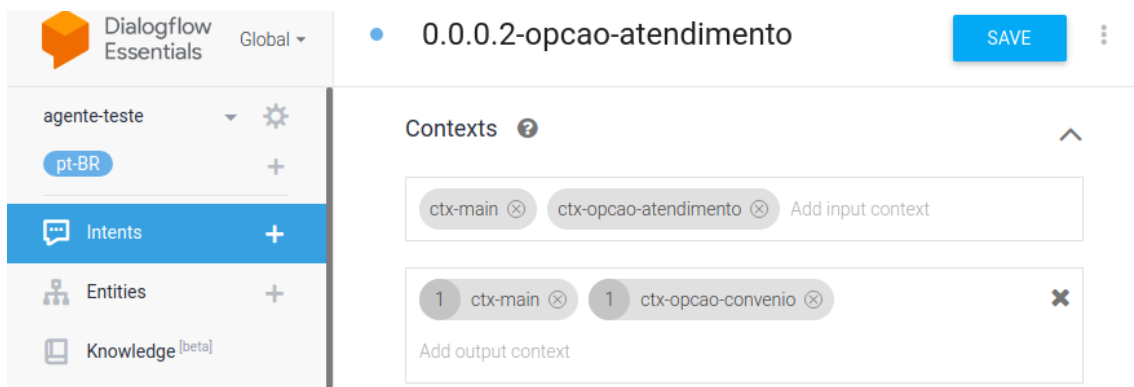
Esta seção explica como adicionar contextos nas intenções do fluxo para acontecer o encaixe da conversa. Primeiramente, siga os seguintes passos para adicionar contextos a uma intenção, conforme ilustrado na Figura 1.9.

1. Selecione uma *intent*;
2. Selecione a opção *Contexts*;
3. Clique na opção *ADD CONTEXT*;
4. No campo *Add input context*, adicione o contexto de entrada;
5. No campo *Add output context*, adicione o contexto de saída;





**Figura 1.8. Adicionando respostas às intenções no Dialogflow.**



**Figura 1.9. Adicionado contextos a uma intenção no Dialogflow.**

6. Clique no botão “Salvar”.

Por fim, siga os passos a seguir para realizar o encadeamento de conversa, considerando as Figuras [1.10](#) e [1.11](#)).

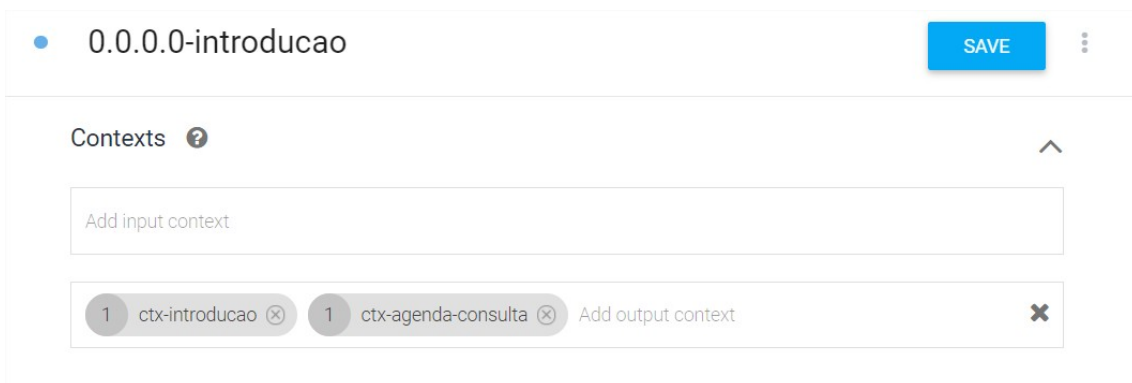
1. Selecione a *intent* “0.0.0.0-introducao”.

- (a) Selecione a opção *Contexts*;
- (b) No campo *Add output context*, adicione os contextos de saída “ctx-introducao” e “ctx-agenda-consulta”;
- (c) Clique no botão “Salvar”.

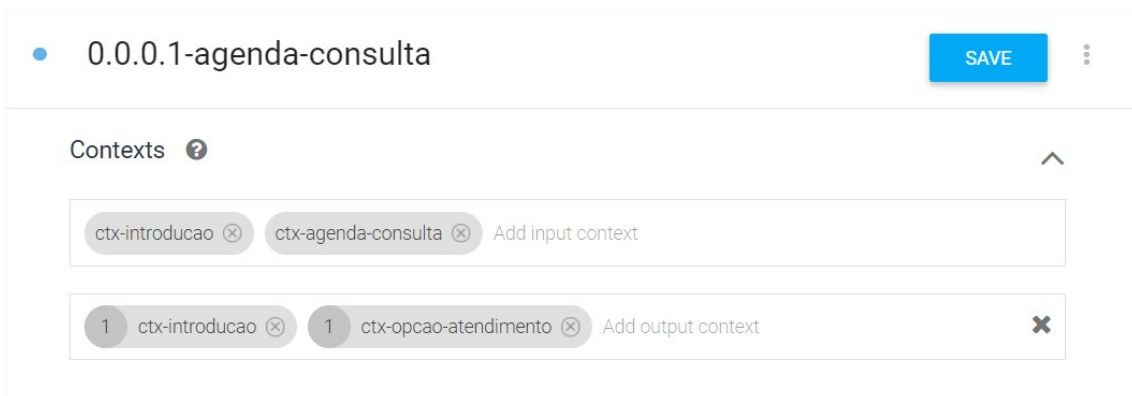
2. Selecione a *intent* “0.0.0.1-agenda-consulta”.

- No campo *Add input context*, adicione os contextos de entrada “ctx-introducao” e “ctx-agenda-consulta”;
- No campo *Add output context*, adicione os contextos de saída “ctx-introducao” e “ctx-opcao-atendimento”.
- Clique no botão “Salvar”.

O processo de encadeamento de conversa precisa ser configurado para todas as intenções do fluxo.



**Figura 1.10. Encadeamento de conversa no Dialogflow.**

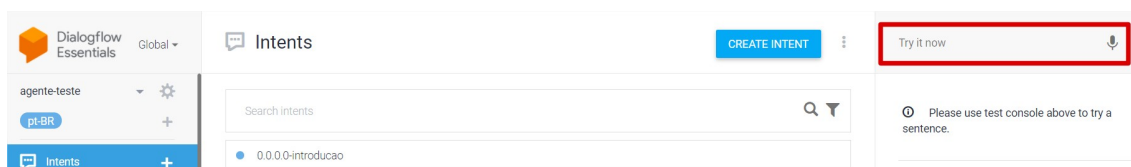


**Figura 1.11. Encadeamento de conversa (agendamento).**

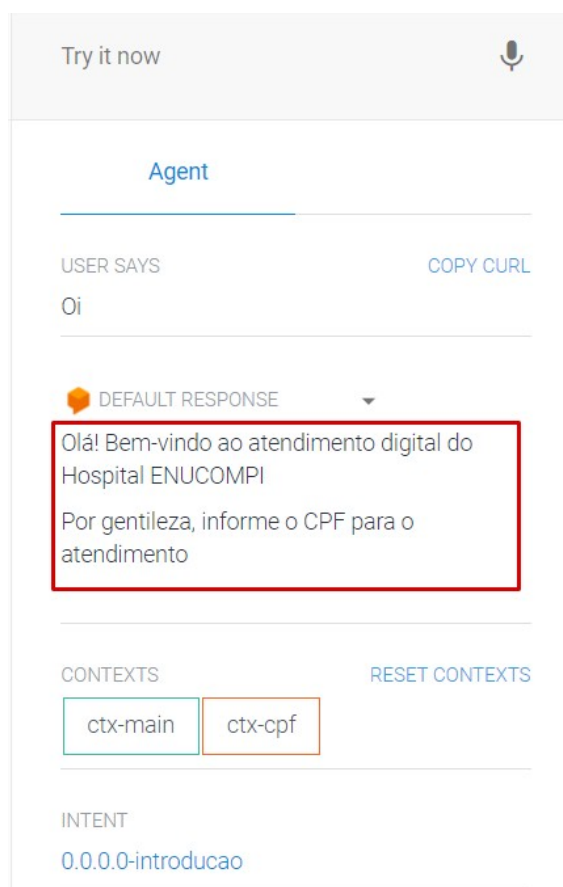
### 1.6.6. Testando o ChatBot

Para realizar o teste do ChatBot desenvolvido, siga os passos a seguir, levando em consideração a Figura [1.12](#).

- Clique na campo *Try it now*;
- Digite uma frase de entrada, por exemplo “Oi” e aperte a tecla “Enter”;
- Como resultado, será apresentado a frase de resposta cadastrada na intenção “0.0.0.0-introducao”(Figura [1.13](#)).



**Figura 1.12. Teste do agente no Dialogflow.**



**Figura 1.13. Teste do agente (resultado).**

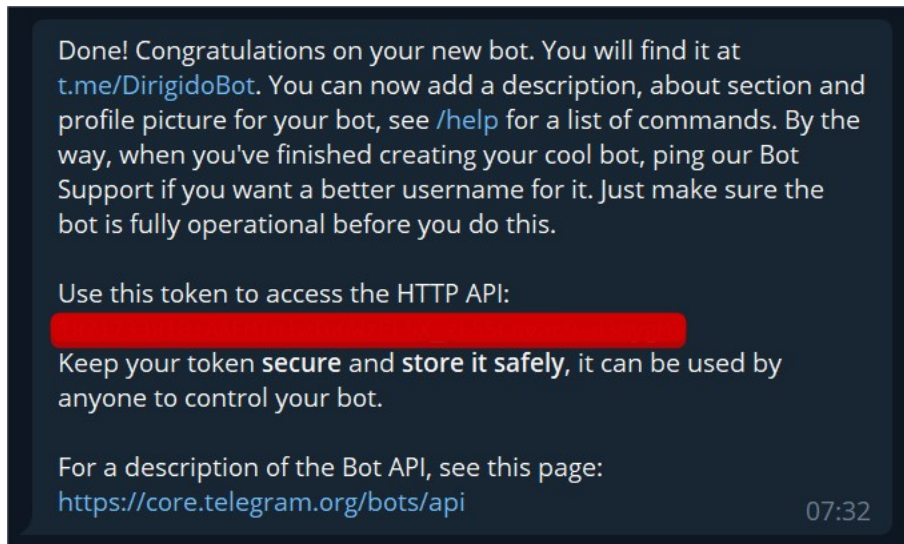
Caso deseje, é possível conectar o ChatBot com a plataforma Telegram. A seguir são apresentados os passos para isso.

1. Primeiro será necessário criar um Bot no “Telegram”, o processo de criação do Bot pode ser consultado na documentação do “Telegram”<sup>5</sup>;
2. Copie o token gerado ao criar o Bot no “Telegram”(Figura 1.14);
3. No Dialogflow, clique no menu *Integrations*;
4. Selecione a opção “Telegram”;

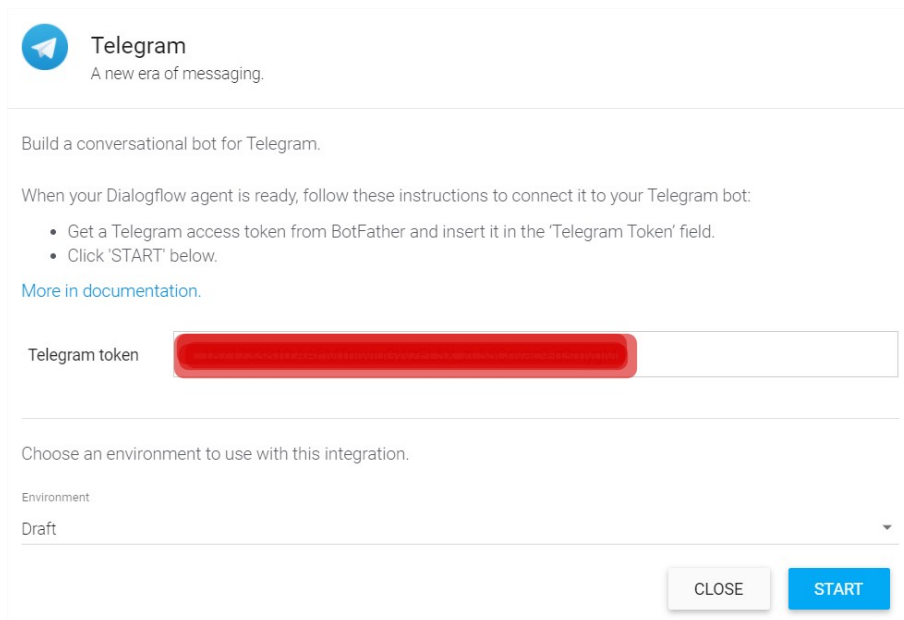
<sup>5</sup><https://core.telegram.org/bots#3-how-do-i-create-a-bot>

5. Cole o token no campo “Telegram token” (Figura 1.15);
6. Por fim, clique no botão *START*.

Após seguir todo o passo a passo, o ChatBot está pronto para ser utilizado, como ilustrado na Figura 1.16.



**Figura 1.14. Token Telegram.**



**Figura 1.15. Integração com o Telegram no Dialogflow.**

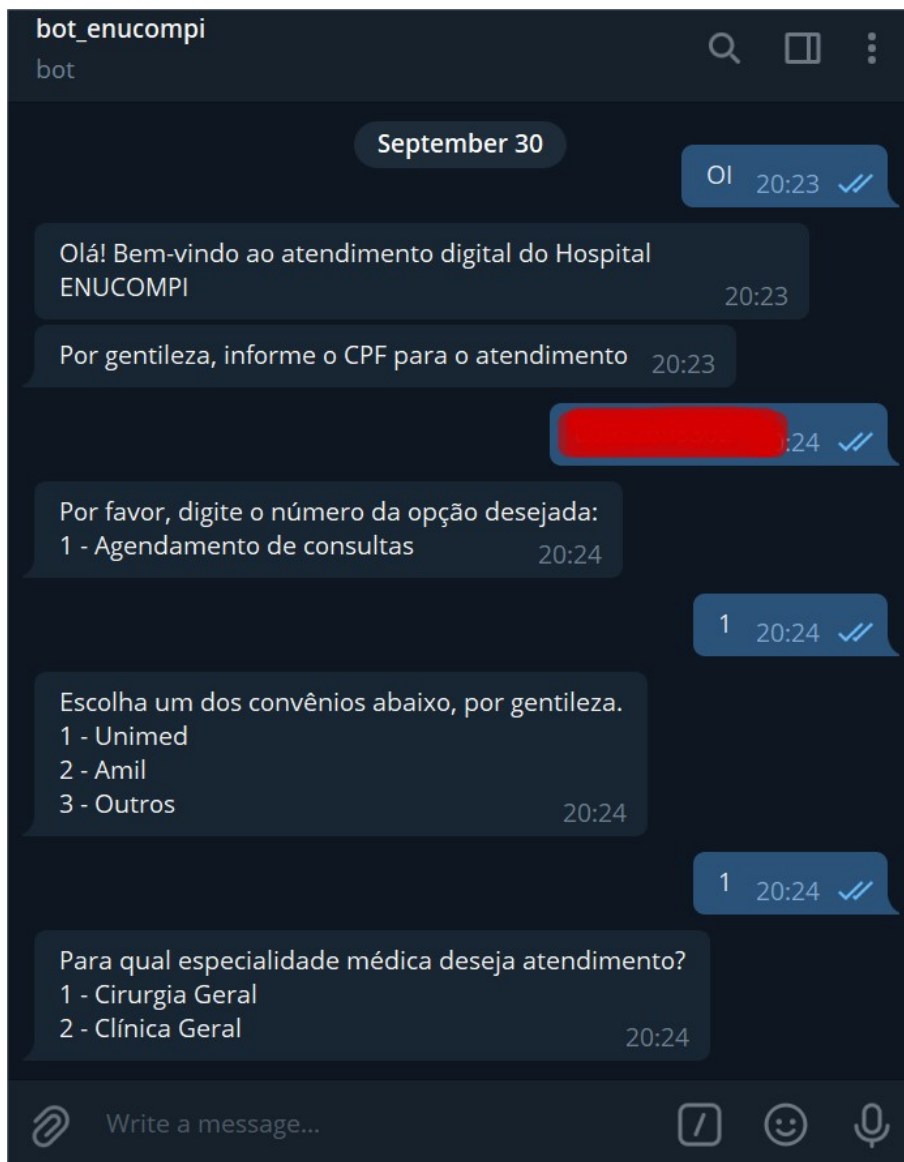


Figura 1.16. Demonstração de conversa com o ChatBot criado.

## 1.7. Considerações Finais

Os ChatBots não são novos no mercado, e suas tecnologias e formas de implementação estão em constante evolução. Isso ocorre na medida em que os avanços nas áreas de processamento de linguagem natural e aprendizado de máquina evoluem. Este capítulo mostrou que as técnicas de PLN e AM disponíveis na plataforma Dialogflow se mostraram eficaz e fáceis de utilização, até mesmo por desenvolvedores não experientes.

No capítulo, foram apresentados os conceitos de ChatBots e as tecnologias utilizadas em seu desenvolvimento. O capítulo deu foco em mostrar como desenvolver um ChatBot para realizar agendamento de consultas médicas com base na plataforma Dialogflow, e integrado ao Telegram. Esse ChatBot pode ser melhorado, como, por exemplo, adicionado integrações com serviços externos (e.g., *Google Calendar*, *API Speech-to-Text*) por

meio de *WebHooks*, e utilizando banco de dados para persistir e consultar informações. Além disso, o Dialogflow fornece suporte para várias integrações, tais como Telegram, Skype, Messenger fom Facebook, Slack, Line, Twilio, Twitter, Viber.

## Referências

- [Agarwala et al. 2019] Agarwala, H., Becker, R., Fatima, M., and Riediger, L. (2019). Development of an artificial conversational entity (ace) for continuous learning and adaptation to user’s preferences and behavior. *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*.
- [Ali 2020] Ali, N. (2020). Chatbot: A conversational agent employed with named entity recognition model using artificial neural network. *CoRR*, abs/2007.04248.
- [Aranha 2017] Aranha, C. N. (2017). Uma abordagem de pré-processamento automático para mineração de textos em português: Sob o enfoque da inteligência computacional.
- [Bocklisch et al. 2017] Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *CoRR*, abs/1712.05181.
- [BULEGON and MORO 2010] BULEGON, H. and MORO, C. M. C. (2010). Mineração de texto e o processamento de linguagem natural em sumários de alta hospitalar.
- [Csaky 2019] Csaky, R. (2019). Deep learning based chatbot models. *CoRR*, abs/1908.08835.
- [Denecke et al. 2021] Denecke, K., Vaaheesan, S., and Arulnathan, A. (2021). A mental health chatbot for regulating emotions (sermo) - concept and usability test. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1170–1182.
- [Forgy 1965] Forgy, E. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–780.
- [Ghandeharioun et al. 2019] Ghandeharioun, A., McDuff, D., Czerwinski, M., and Rowan, K. (2019). Emma: An emotion-aware wellbeing chatbot. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–7.
- [Gupta et al. 2021] Gupta, J., Singh, V., and Kumar, I. (2021). Florence- a health care chatbot. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 504–508.
- [Kar and Haldar 2016a] Kar, R. and Haldar, R. (2016a). Applying chatbots to the internet of things: Opportunities and architectural elements. *International Journal of Advanced Computer Science and Applications*, 7(11).
- [Kar and Haldar 2016b] Kar, R. and Haldar, R. (2016b). Applying chatbots to the internet of things: Opportunities and architectural elements. *CoRR*, abs/1611.03799.

- [Karve et al. 2018] Karve, S., Nagmal, A., Papalkar, S., and Deshpande, S. A. (2018). Context sensitive conversational agent using dnn. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 475–478.
- [Kraus 2007] Kraus, H. (2007). Protótipo de um chatterbot para área imobiliária integrado a tecnologia de reacionínio baseado emcasos.
- [Leptourgos and Corlett 2020] Leptourgos, P. and Corlett, P. R. (2020). Embodied predictions, agence, and psychosis.
- [Light 1998] Light, M. (1998). *Journal of Logic, Language, and Information*, 7(1):111–114.
- [Maldonado and Cuadra 2019a] Maldonado, J. A. V. and Cuadra, J. A. G. (2019a). Natural language interface to database using the dialogflow voice recognition and text conversion api. In *2019 8th International Conference On Software Process Improvement (CIMPS)*, pages 1–10.
- [Maldonado and Cuadra 2019b] Maldonado, J. A. V. and Cuadra, J. A. G. (2019b). Natural language interface to database using the dialogflow voice recognition and text conversion api. In *2019 8th International Conference On Software Process Improvement (CIMPS)*, pages 1–10.
- [Mathew et al. 2019] Mathew, R. B., Varghese, S., Joy, S. E., and Alex, S. S. (2019). Chatbot for disease prediction and treatment recommendation using machine learning. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 851–856.
- [Mellado-Silva et al. 2020] Mellado-Silva, R., Faúndez-Ugalde, A., and Lobos, M. B. (2020). Learning tax regulations through rules-based chatbots using decision trees: a case study at the time of covid-19. In *2020 39th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–8.
- [Morais and Ambrósio 2007] Morais, E. A. M. and Ambrósio, A. P. (2007). Mineração de textos.
- [Muhammad et al. 2020a] Muhammad, A. F., Susanto, D., Alimudin, A., Adila, F., As-sidiqi, M. H., and Nabhan, S. (2020a). Developing english conversation chatbot using dialogflow. In *International Electronics Symposium (IES)*, pages 468–475.
- [Muhammad et al. 2020b] Muhammad, A. F., Susanto, D., Alimudin, A., Adila, F., As-sidiqi, M. H., and Nabhan, S. (2020b). Developing english conversation chatbot using dialogflow. In *2020 International Electronics Symposium (IES)*, pages 468–475.
- [Nguyen and Shcherbakov 2018] Nguyen, T. and Shcherbakov, M. (2018). A neural network based vietnamese chatbot. In *2018 International Conference on System Modeling Advancement in Research Trends (SMART)*, pages 147–149.

- [Pacheco and Pereira 2018] Pacheco, C. and Pereira, N. (2018). Deep learning conceitos e utilização nas diversas Áreas do conhecimento. 2:34–49.
- [Pérez-Soler et al. 2021] Pérez-Soler, S., Guerra, E., and de Lara, J. (2021). Creating and migrating chatbots with conga. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 37–40.
- [Sabharwal et al. 2019] Sabharwal, N., Barua, S., Anand, N., and Aggarwal, P. (2019). *Developing Cognitive Bots Using the IBM Watson Engine*.
- [Salvi et al. 2019] Salvi, S., Geetha, V., and Sowmya Kamath, S. (2019). Jamura: A conversational smart home assistant built on telegram and google dialogflow. In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pages 1564–1571.
- [Santos et al. 2014] Santos, R. E. S., Neto, J. S. C., Souza, E. P. R., Magalhães, C. V. C., and Vilar, G. (2014). Técnicas de processamento de linguagem natural aplicadas ao processamento de mineração de textos: Resultados preliminares de mapeamento sistemático. *revista de sistemas e computação*.
- [Softić et al. 2021] Softić, A., Husić, J. B., Softić, A., and Baraković, S. (2021). Health chatbot: Design, implementation, acceptance and usage motivation. In *2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–6.
- [TURING 1950] TURING, A. M. (1950). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.
- [von Rueden et al. 2021] von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Walczak, M., Pfrommer, J., Pick, A., Ramamurthy, R., Garcke, J., Bauckhage, C., and Schuecker, J. (2021). Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.
- [Weizenbaum 1966] Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.