



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIA E SAÚDE - CAMPUS ARARANGUÁ  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Rafael Selau M. Rocho

**Uma proposta de integração entre tecnologias chatbots e LLMs**

Araranguá  
2023

Rafael Selau M. Rocho

**Uma proposta de integração entre tecnologias chatbots e LLMs**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologia e Saúde - Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Alison R. Panisson, Dr.

Araranguá  
2023

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rochó, Rafael Selau Martins

Uma proposta de integração entre tecnologias chatbots e LLMs Araranguá / Rafael Selau Martins Rochó ; orientador, Alison Roberto Panisson, 2023.

24 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2023.

Inclui referências.

1. Engenharia de Computação. 2. Chatbots. 3. Modelos de Linguagem. 4. Lei de trânsito.. I. Panisson, Alison Roberto. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Rafael Selau M. Rocho

**Uma proposta de integração entre tecnologias chatbots e LLMs**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Computação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 30 de Novembro de 2023.

---

Prof. Jim Lau, Dr.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Alison R. Panisson, Dr.  
Orientador

---

Prof. Giovanni Parente Farias, Dr.  
Avaliador  
Instituição FURG

---

Prof. Anderson Luiz Fernandes Perez, Dr.  
Avaliador  
Instituição UFSC

# Uma proposta de integração entre tecnologias chatbots e LLMs

Rafael Selau M. Rocho\*

Alison R. Panisson†

2023, December

## Resumo

O presente trabalho tem como objetivo propor uma integração híbrida de tecnologias para o desenvolvimento de chatbots. A integração proposta incorpora framework de desenvolvimento de chatbots baseados em regras e LLMs. Como estudo de caso da integração proposta, foi desenvolvida uma aplicação voltada para a legislação do trânsito brasileira Lei nº 9.503 de 23 de setembro de 1997, Capítulo XV - Infrações. O estudo destaca não apenas a viabilidade técnica da integração proposta, mas também ressalta uma série de desafios relacionados a desenvolvimentos similares. Por fim, o trabalho também destaca áreas promissoras para futuras pesquisas, como a adaptação do chatbot desenvolvido a diferentes legislações, a melhoria da acessibilidade, etc. De forma geral, o estudo demonstra o potencial da integração entre framework para desenvolvimento de chatbots e LLMs, permitindo o desenvolvimento de chatbots inteligentes e personalizados.

**Palavras-chaves:** Chatbots, Modelos de Linguagem, Lei de trânsito.

---

\*rafaelrocho\_@hotmail.com

†alison.panisson@ufsc.br

# An Approach for Integrating Chatbots and LLMs Technologies

Rafael Selau M. Rocho\*

Alison R. Panisson†

2023, December

## Abstract

The present work aims to propose an integration of hybrid technologies for chatbot development. The proposed integration incorporates rule-based chatbot development framework and LLMs (Large Language Models). As a case study for the proposed integration, was developed an application focused on Brazilian traffic legislation Law No. 9,503 of September 23, 1997, Chapter XV - Infractions. The study highlights not only the technical feasibility of the proposed integration but also emphasizes a series of challenges related to similar developments. Finally, the paper also highlights promising areas for future research, such as adapting the developed chatbot to different legislations, improving accessibility, etc. Overall, the study demonstrates the potential of integration between framework for chatbot development and LLMs, enabling the development of intelligent and personalized chatbots.

**Key-words:** Rule-based Chatbots, Large Language Models, Traffic Law.

---

\*rafaelrocho\_@hotmail.com

†alison.panisson@ufsc.br

# 1 Introdução

Nos últimos anos, a automação e a inteligência artificial têm desempenhado um papel significativo na simplificação de processos e na melhoria da interação entre sistemas computacionais e seres humanos. Uma aplicação notável deste avanço tecnológico é a construção de chatbots, que são programas de computador projetados para realizar conversas de forma autônoma e interativa, como por exemplo, o Bard da Google e o ChatGPT da OpenAI. Conforme [Pieraccini \(2012\)](#) observou, a criação de uma máquina capaz de falar, entender a fala e exibir comportamento inteligente era um objetivo compartilhado muito antes de 1700, no entanto, as tentativas científicas para replicar e compreender a fala humana só começaram no final da década de 1930 e nos anos 1950.

Este trabalho tem como objetivo principal explorar a integração entre o Rasa, uma plataforma de desenvolvimento de chatbots altamente personalizável e versátil, com LLMs (*Large Language Models*), em particular, com a API da OpenAI. A proposta deste trabalho segue apontamentos recentes, que destacam o uso de LLMs como uma ferramenta complementar, e que seu uso pode melhorar a qualidade das respostas em aplicações chatbot ([MCTEAR; MAROKKIE; BI, 2023](#)). A integração proposta representa um avanço significativo na capacidade de criar chatbots sofisticados e inteligentes, destacando as oportunidades e desafios que essa integração oferece. Como forma de avaliar a proposta de integração apresentada nesse trabalho, foi desenvolvido um chatbot. O desenvolvimento implementa uma aplicação na área de legislação de trânsito brasileira - Lei nº 9.503 ([BRASIL, 1997](#)), com o foco principal em seu Capítulo XV - Infrações.

As infrações cometidas por motoristas podem acarretar em diversas penalidades, como multas, suspensão do direito de dirigir e até mesmo processos judiciais. Além disso, segundo ([ABREU; SOUZA; MATHIAS, 2020](#)), os acidentes de trânsito são reconhecidos como uma questão relevante na área da saúde, acarretando índices elevados de fatalidade e uma quantidade significativa de indivíduos lesionados, diariamente, tanto em zonas urbanas quanto nas rodovias. De acordo com ([HENRIQUE, 2020](#)), os acidentes envolvendo o transporte terrestre no Brasil são responsáveis por cerca de quarenta e cinco mil mortes por ano, sendo um dos países com maior número de mortes no trânsito, ficando atrás de países como Índia e China por exemplo ([ABREU; SOUZA; MATHIAS, 2020](#)). No entanto, a complexidade dessas leis e a falta de conhecimento sobre as regras específicas muitas vezes dificultam a compreensão por parte dos motoristas e pedestres, o que acaba acarretando em acidentes, e conseqüentemente nos índices destacados.

Neste trabalho, serão explorados os benefícios e desafios envolvidos na criação de uma aplicação chatbot utilizando a integração das ferramentas do Rasa e a API da OpenAI, para a Lei nº 9.503/1997 ([BRASIL, 1997](#)) - CTB, Capítulo XV - Infrações, discutindo sua importância do conhecimento jurídico e os aspectos envolvidos em seu desenvolvimento. O chatbot desenvolvido servirá como um estudo de caso prático para ilustrar o potencial da integração proposta na construção de soluções de IA, visando fornecer uma visão sobre sua aplicabilidade no contexto da legislação de trânsito brasileira.

## 2 Tecnologias para o Desenvolvimento de Chatbots

Os chatbots desempenham um papel importante no campo da inteligência artificial e são projetados para simular a interação humana ([BANSAL; KHAN, 2018](#)). Eles têm sido cada vez mais utilizados em uma variedade de setores, incluindo atendimento ao cliente, suporte técnico, marketing e comércio eletrônico ([SHAWAR; ATWELL, 2007](#)).

O desenvolvimento de chatbots têm crescido significativamente nos últimos anos, impulsionado pelo aumento da adoção de tecnologias de inteligência artificial. Segundo o relatório do Grand View Research (RESEARCH, 2021), o mercado global de chatbots foi estimado em US\$ 5.132,8 milhões em 2022 e deve crescer a uma taxa de crescimento anual de 23,3% de 2023 a 2030.

De acordo com o mesmo relatório (RESEARCH, 2021), a crescente exigência dos clientes por operações de autoatendimento tem impulsionado a necessidade de disponibilizar serviços ao cliente 24 horas por dia, enquanto se busca reduzir os custos operacionais por meio da delegação de tarefas a chatbots. Esse aumento nas demandas dos clientes por autoatendimento tem sido um fator impulsionador para o aumento da procura de tais tecnologias.

## 2.1 Frameworks para o Desenvolvimento de Chatbots

Nesta seção serão apresentadas as tecnologias mais conhecidas no desenvolvimento de chatbots, destacadas pelo conjunto de ferramentas (*frameworks*), interfaces de desenvolvimento, ampla utilização comercial, etc.

### 2.1.1 Rasa

Rasa<sup>1</sup> é uma plataforma de código aberto para criar chatbots e assistentes virtuais personalizados. É uma solução completa que inclui ferramentas para treinar modelos de linguagem natural, gerenciar diálogos e integrar com outras ferramentas. A arquitetura do Rasa é composta por dois componentes principais: Rasa NLU e Rasa Core. O Rasa NLU entende a linguagem natural dos usuários e extrai informações relevantes, enquanto o Rasa Core controla a lógica das conversas e determina as respostas adequadas do chatbot.

Para treinar a NLU (*Natural Language Understanding*) do Rasa, pode-se fornecer exemplos de conversas e perguntas que os usuários poderiam fazer ao chatbot. Com base nesses exemplos, o Rasa NLU usa algoritmos de aprendizado de máquina para identificar padrões na linguagem e criar um modelo capaz de reconhecer as intenções dos usuários e as informações relevantes em suas solicitações.

O Rasa Core gerencia a lógica das conversas do chatbot. Ele utiliza algoritmos de aprendizado de máquina para decidir qual ação o chatbot deve tomar em resposta às solicitações do usuário. Por exemplo, se o usuário perguntar sobre o horário de funcionamento de uma empresa, o Rasa Core pode responder com os horários de funcionamento atuais ou pedir mais informações ao usuário se a solicitação não estiver clara o suficiente para gerar uma resposta. Além disso, o Rasa oferece várias ferramentas e recursos para auxiliar na construção e gerenciamento de chatbots. Isso inclui um kit de desenvolvimento de software (SDK) para Python, uma interface de linha de comando (CLI), entre outras. O Rasa também tem sido explorado em contextos acadêmicos, com integração a outras tecnologias de IA distribuída, como por exemplo sistemas multiagentes (CUSTÓDIO et al., 2022).

### 2.1.2 Dialogflow

Dialogflow<sup>2</sup> é uma plataforma desenvolvida pelo Google para criar chatbots e agentes de conversação. Ela utiliza tecnologia de processamento de linguagem natural

---

<sup>1</sup> <<https://rasa.com/>>

<sup>2</sup> <<https://cloud.google.com/dialogflow>>



(NLP) para permitir que os usuários criem chatbots capazes de se comunicar de forma natural. Essa plataforma permite criar chatbots em diversos canais, como aplicativos móveis, websites e assistentes de voz, como o Google Assistant. Além disso, ela oferece integração com outros serviços do Google, como o Google Cloud Speech-to-Text e o Google Cloud Text-to-Speech, o que possibilita que os chatbots interajam com os usuários em diferentes idiomas.

O Dialogflow utiliza várias ferramentas e recursos para ajudar os desenvolvedores a criar chatbots eficientes. Por exemplo, a plataforma oferece uma interface de usuário intuitiva baseada em arrastar e soltar, que permite criar fluxos de conversação rapidamente, tanto para responder a perguntas frequentes quanto para sequências de diálogo mais complexas. Além disso, o Dialogflow utiliza aprendizado de máquina para melhorar continuamente o desempenho dos chatbots. Dessa forma, eles aprendem com as interações anteriores e aprimoram sua capacidade de entender e responder aos usuários. Como o Rasa, o Dialogflow também tem sido estudado em contextos de integração com outras tecnologias de IA distribuída, como por exemplos sistemas multiagentes (ENGELMANN et al., 2021a), sendo aplicado no desenvolvimento de aplicações de IA distribuída em domínios diversos, como na saúde (ENGELMANN et al., 2021b; ENGELMANN et al., 2023).

### 2.1.3 Botpress

Botpress<sup>3</sup> é uma plataforma de desenvolvimento de chatbots de código aberto que permite criar chatbots personalizados, escaláveis e seguros, utilizando tecnologias avançadas de inteligência artificial e aprendizado de máquina. Essa plataforma oferece uma variedade de recursos, como análise de conversas, integração com várias plataformas de mensagens e gerenciamento de fluxo de conversas, entre outros.

O Botpress é baseado em tecnologias modernas de inteligência artificial, como processamento de linguagem natural e aprendizado de máquina. Além disso, ele possui uma interface gráfica de usuário intuitiva, o que facilita a construção e a manutenção dos chatbots. O Botpress também conta com recursos avançados de segurança, como criptografia de dados em repouso e em trânsito, autenticação de usuários e gerenciamento de acesso. Isso o torna uma opção segura para empresas que lidam com informações confidenciais.

### 2.1.4 ManyChat

ManyChat<sup>4</sup> é uma plataforma de chatbot e automação de marketing que permite criar chatbots para interagir com os usuários em várias plataformas de mensagens, como Facebook, WhatsApp, Instagram, entre outras. A plataforma possui uma interface amigável baseada em arrastar e soltar, que facilita a criação e personalização dos fluxos de conversação de forma intuitiva, sem a necessidade de habilidades avançadas de programação.

## 2.2 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural (PLN) é um campo da inteligência artificial que se concentra na interação entre computadores e linguagem humana. O destaque do PLN é permitir que os computadores entendam e respondam a declarações e textos escritos em uma linguagem natural. O PLN usa várias técnicas para interpretar a linguagem

---

<sup>3</sup> <<https://botpress.com/>>

<sup>4</sup> <<https://manychat.com/>>

humana, como a análise da estrutura gramatical das frases, a análise semântica e a análise pragmática, ou seja, como o contexto influencia o significado das frases.

A criação de programas que entendem uma linguagem natural é uma tarefa difícil (CHOWDHARY, 2020). A maioria das línguas naturais são grandes e possuem muita ambiguidade na linguagem natural, ou seja, palavras com vários significados.

### 2.3 Large Language Models (LLMs)

Os Large Language Models (LLMs), ou modelos de linguagem de grande escala, são uma classe de modelos de IA que têm a capacidade de entender e gerar texto de maneira muito semelhante à linguagem humana. Eles são treinados em grandes conjuntos de dados textuais para aprender padrões de linguagem, gramática, semântica e até mesmo o contexto cultural. À medida que o tamanho do parâmetro dos LLMs continua a aumentar, estudos recentes indicaram que os LLMs podem levar ao surgimento de capacidades notáveis (CHEN et al., 2023). Exemplos, como o GPT-3.5 e GPT-4 da OpenAI, Claude da Anthropic, e o Bard da Google, são destacados por sua capacidade de realizar uma variedade de tarefas de processamento de linguagem natural.

Os LLMs são compostos por redes neurais profundas que podem ser alimentadas com texto de entrada e, em seguida, gerar texto de saída que é coerente e relevante para a entrada fornecida. Segundo (FAN et al., 2023), estudos recentes indicaram que os LLMs exibem capacidades impressionantes de generalização e raciocínio, fazendo com que os LLMs generalizem melhor para uma variedade de tarefas e domínios não vistos, ou seja, em vez de exigir um ajuste em cada tarefa específica, os LLMs podem aplicar seu conhecimento aprendido e habilidades de raciocínio para se ajustar a novas tarefas simplesmente fornecendo instruções apropriadas ou algumas demonstrações de tarefas. Isso torna esses modelos incrivelmente versáteis para tarefas como (KASNECI et al., 2023):

- **Geração de texto:** um dos usos mais notáveis dos LLMs é a geração de texto coerente e natural. Eles podem criar histórias, poesias, código de programação e muito mais, de acordo com as entradas específicas.
- **Tarefas de Processamento de Linguagem Natural (PNL):** além da geração de texto, LLMs são usados em uma variedade de tarefas de processamento de linguagem natural, como tradução automática, respostas a perguntas, sumarização de texto, entre outros.
- **Aplicações práticas:** os LLMs têm uma ampla gama de aplicações no mundo real. Eles podem ser usados em chatbots, assistentes virtuais, corretores de textos, traduções, pesquisa inteligente, geração automática de texto e muito mais.

Os LLMs, mesmo se mostrados ferramentas eficazes na área de processamento de linguagem natural, também acabam enfrentando diversos desafios e dificuldades como por exemplo (KADDOUR et al., 2023):

- **Coerência e contexto:** podem gerar respostas que parecem plausíveis, mas podem carecer de coesão e contexto a longo prazo. Manter uma conversa coerente e relevante é uma dificuldade, especialmente em interações mais longas.

- **Ambiguidade:** modelos de linguagem muitas vezes têm dificuldade em lidar com ambiguidade na linguagem, o que pode levar a interpretações incorretas ou respostas ambíguas.
- **Conhecimento limitado:** os LLMs têm uma data de corte de conhecimento, ou seja, só têm informações disponíveis até uma determinada data, quando os mesmos foram gerados.
- **Desafios de privacidade e segurança:** o uso indiscriminado de LLMs para gerar conteúdo pode criar problemas de segurança, como a disseminação de informações falsas e ameaças à privacidade.
- **Robustez:** os modelos de linguagem são sensíveis a pequenas mudanças na entrada e podem gerar respostas inconsistentes ou incorretas sob variações mínimas de contexto.

### 2.3.1 OpenAI

A OpenAI<sup>5</sup> é uma empresa de pesquisa e desenvolvimento em inteligência artificial que se dedica a criar modelos de linguagem avançados. A API (Application Programming Interface) da OpenAI é uma interface que permite aos desenvolvedores integrar esses modelos em seus aplicativos, serviços e sistemas. A API da OpenAI fornece acesso a modelos de linguagem treinados pela OpenAI, incluindo o GPT-3.5 e GPT-4.

A API da OpenAI permite que desenvolvedores usem esses modelos para tarefas como geração de texto, tradução, resumo de texto, resposta a perguntas e entre outros. Os desenvolvedores podem fazer solicitações para a API com texto de entrada e receber respostas geradas pelo modelo, o que torna a tecnologia de IA acessível e eficaz para uma ampla variedade de aplicativos.

## 2.4 Retrieval Augmented Generation (RAG)

A *Retrieval Augmented Generation* (RAG) é uma abordagem inovadora e eficiente que implementa uma estratégia para melhorar a qualidade das respostas geradas por *Large Language Models* (LLMs) ao fundamentar o modelo em fontes externas de conhecimento para complementar a representação interna de informações do LLM (ZHAO et al., 2023).

Os LLMs podem ser inconsistentes, às vezes, eles acertam a resposta para perguntas, outras vezes lançam fatos aleatórios de seus dados de treinamento. Um dos motivos do por que não podemos dar aos LLMs toda a base de conhecimento é que os modelos têm limites integrados sobre a quantidade de texto que podem consumir por vez. Outra razão é o seu elevado custo de enviar uma grande quantidade de dados. Segundo Liu et al. (2023), o desempenho de um LLM é melhor quando a informação relevante ocorre no início ou no final do contexto de entrada, e piora quando os modelos devem submeter-se a informação relevante no meio de contextos longos. No mesmo artigo (LIU et al., 2023), o autor comenta que o desempenho diminui significativamente à medida que o contexto de entrada aumenta, mesmo para modelos explicitamente de contexto longo.

Como solução, os métodos RAG (LEWIS et al., 2020) oferecem uma solução promissora para os LLMs interagirem efetivamente. Em sua essência, o RAG busca aprimorar a geração de texto ao incorporar mecanismos de recuperação de informações. Ele opera com dois componentes principais: o modelo de geração de linguagem, responsável

---

<sup>5</sup> <<https://openai.com/>>

por criar texto coerente e significativo, e um modelo de recuperação de informações, que incorpora a capacidade de recuperar informações específicas de fontes externas (ZHAO et al., 2023).

O RAG representa uma fronteira interessante na inteligência artificial, prometendo avanços significativos na geração de texto de alta qualidade, enriquecida por dados e contextualmente relevantes. Essa abordagem tem o potencial de transformar a maneira como interagimos com a informação, criando conteúdo mais preciso e informativo em uma variedade de contextos, desde assistentes virtuais até geração de conteúdo para pesquisa e divulgação de conhecimento.

## 2.5 Código de Trânsito Brasileiro (CTB)

Nesta seção será contextualizado o estudo de caso utilizado neste trabalho. Onde, como forma de validar a proposta de integração de tecnologias, um chatbot para responder questões relacionadas a seção de infrações do Código de Trânsito Brasileiro (CTB) foi desenvolvida. O CTB é instituído pela Lei nº 9.503 (BRASIL, 1997), de 23 de setembro de 1997, e como outros documentos legais, possui estrutura documental desafiadora para a área de processamento de linguagem natural (KATZ et al., 2023).

### 2.5.1 Lei nº 9.503/1997

A legislação de trânsito brasileira é estabelecida pelo Código de Trânsito Brasileiro (CTB), instituído pela Lei nº 9.503/1997 (BRASIL, 1997). Essa lei é responsável por regulamentar todas as questões relacionadas ao tráfego de veículos e pedestres em território nacional, estabelecendo direitos, deveres e penalidades para os usuários das vias públicas. O CTB é um documento bastante abrangente, que trata de diversos aspectos do trânsito. Entre os principais pontos contemplados pela lei, destacam-se:

- **Condutores e veículos:** o CTB estabelece as regras para a habilitação de condutores, incluindo os requisitos para obtenção da Carteira Nacional de Habilitação (CNH) e as categorias de veículos. Além disso, define as responsabilidades e deveres dos condutores, como respeitar os limites de velocidade, não dirigir sob efeito de álcool ou substâncias psicoativas, usar o cinto de segurança, entre outros.
- **Infrações e penalidades:** o CTB define as infrações de trânsito, classificadas em quatro categorias: leves, médias, graves e gravíssimas. Cada infração possui uma penalidade correspondente, que pode variar desde uma advertência por escrito até a suspensão da CNH, além do pagamento de multas. O CTB também determina a pontuação que é atribuída ao condutor infrator, que pode resultar na suspensão ou cassação da habilitação.
- **Sinalização e regras de tráfego:** o CTB estabelece os padrões de sinalização das vias públicas, incluindo a sinalização com placas e marcas no asfalto. Além disso, define as regras de tráfego, como a preferência nos cruzamentos, a utilização das faixas exclusivas, o respeito à preferência dos pedestres, entre outros aspectos importantes para garantir a segurança no trânsito.
- **Transporte de passageiros:** o CTB aborda também as regras e condições para o transporte de passageiros, seja em veículos particulares, táxis ou transporte coletivo. São estabelecidas normas de segurança para o transporte de crianças, limites de

lotação dos veículos, além de determinar os direitos dos passageiros em casos de atrasos e irregularidades.

O CTB passou e passa por algumas alterações ao longo dos anos, buscando aprimorar a legislação e a segurança no trânsito. Portanto, é fundamental que os condutores estejam sempre atualizados sobre as leis vigentes.

### 2.5.2 Estrutura de um Documento de Lei Brasileira e seus Desafios

A estrutura dos documentos legais, segue um padrão estabelecido pela legislação nacional (BRASIL, 1998). Citando a seguir os principais:

- Capítulos: as leis são frequentemente divididas em capítulos, que organizam o conteúdo em seções. Cada capítulo é identificado por um número romano (I, II, III, ...) e um título correspondente, que no presente trabalho será abordado principalmente o Capítulo XV - Infrações.
- Artigos: compõem a matéria legislada. Ela é identificada através de numeração ordinal para o primeiro ao nono artigo (art. 1º, art. 2º, ..., art. 9º), e numeração cardinal a partir do artigo 10 (art. 10, art. 11, ...).
- Parágrafo: é a divisão do artigo e pode conter explicações ou modificações da proposição anterior. Os Parágrafos são representado pelo símbolo “§” e seguida pela numeração (§ 4º e § 15).
- Caput: vem do termo em latim que denota “cabeça”, é utilizado para se referir à parte principal de um artigo de lei, especialmente quando ele inclui incisos e/ou parágrafos. Essa parte inicial representa o enunciado fundamental do artigo.
- Inciso: de maneira geral, essa estrutura é utilizada para descrever situações, hipóteses ou listagens relacionadas à regra estabelecida no início do artigo (caput). Na escrita da lei são representados apenas por um número romano (I, II, III,...)
- Alínea: ela também tem o propósito de listar e detalhar situações, hipóteses, exceções e outros elementos. As Alíneas, por sua vez, são subdivisões dos incisos, nas quais são representadas por letras minúsculas (a, b, c, ...).

Uns dos principais desafios de se trabalhar com leis é a complexidade linguística que se utiliza de termos técnicos e jurídicos específicos, da atualizações e alterações legislativas que está em constante evolução e aprovação de novas leis, decretos, emendas constitucionais, entre outros.

## 3 Proposta de Integração de Tecnologias Chatbots e LLMs

Como apontado pela literatura, uma abordagem híbrida que combina chatbots baseados em regras e LLMs tem se mostrado bastante promissora no desenvolvimento de chatbots sofisticados, que atendem as necessidade de domínios específicos (MCTEAR; MAROKKIE; BI, 2023). Nesse contexto, propomos uma integração (Figura 1) de chatbots desenvolvidos com a *framework* Rasa e LLMs, em particular, utilizando uma abordagem de RAG.

Fonte: O Autor.



Figura 1 – Diagrama Ferramental.

A implementação do RAG em um sistema de resposta a perguntas baseado em LLM assegura que o modelo tenha acesso aos fatos mais recentes e confiáveis, especialmente no contexto jurídico, que está sujeito a mudanças constantes. Ao fundamentar um LLM em um conjunto de fatos externos e verificáveis, o modelo tem menos oportunidades de usar informações incorporadas em seus parâmetros. Isso reduz as chances de um LLM responder com informações incorretas ou enganosas (ZHAO et al., 2023).

Uma visão geral da abordagem baseada em RAG pode ser observada na Figura 2. O processo inicia com uma determinada entrada realizada pelo usuário, por exemplo, perguntando "Como funciona a...?". Em seguida, é disparado um processo de busca de informações (1), que se concentra em procurar a dúvida do usuário e localizar na base de conhecimento o conteúdo mais relevante (2) para respondê-la. Após extraído os dados relevantes da base de informação, é encaminhado esses dados juntamente com a questão do usuário ao modelo de linguagem de grande escala (3), o LLM. Então o LLM assimila as informações fornecidas e oferece a resposta à pergunta.

Fonte: O Autor.

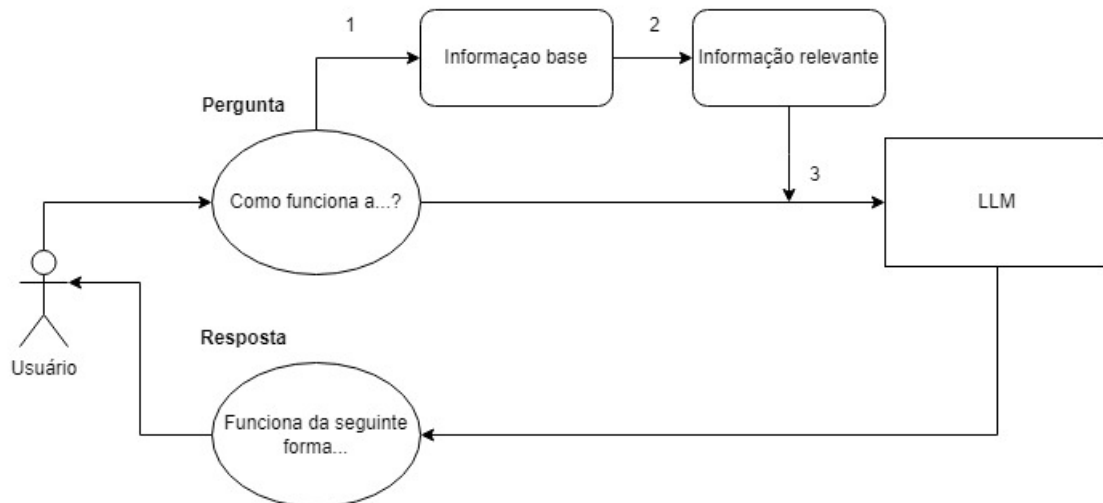


Figura 2 – Diagrama RAG.

Chatbots têm se tornado uma parte essencial das interações online, seja para

melhorar a experiência do cliente, automatizar tarefas ou fornecer suporte. Integrar o Rasa, um framework de desenvolvimento de chatbots popular, considerado um dos mais utilizados no mundo (SHARMA; JOSHI, 2020), com LLM da da OpenAI através da sua API, uma expressiva ferramenta de processamento de linguagem natural, permite a criação facilitada de chatbots, com características de inteligência e versatilidade. Os passos de desenvolvimento da integração proposta são enumerados abaixo:

1. Planejamento e Design: O primeiro passo é definir o propósito e os objetivos do chatbot. Qual problema ele resolverá? Quais perguntas ele responderá? Como ele se encaixará na estratégia de negócios? Além disso, é importante projetar a arquitetura do chatbot, incluindo os fluxos de conversação, as intenções do usuário e as respostas do chatbot. A seguir será explorado as etapas e o processo de desenvolvimento desse chatbot, junto com as ferramentas utilizadas.
2. Treinamento do Modelo Rasa: o Rasa utiliza aprendizado de máquina para entender as intenções do usuário e gerar respostas apropriadas. O modelo de linguagem é treinado com dados de treinamento, que incluem exemplos de diálogos e intenções do usuário. O Rasa fornece uma série de ferramentas para treinar, avaliar e ajustar o modelo até que ele atinja um desempenho satisfatório.
3. Integração com a API OpenAI: a API OpenAI é usada como interface para um LLM, o que possibilita enriquecer a capacidade do chatbot de processar a linguagem natural, podendo assim gerar respostas mais contextualmente relevantes e humanas. Além de deixar a parte da codificação mais simples.
4. Engenharia de Prompt: foi definido um prompt para acrescentar instruções a API OpenAI, utilizando a abordagem de RAG descrita, com o objetivo de atingir respostas mais satisfatórias.
5. Testes e Ajustes: após a integração da API OpenAI, é importante testar o chatbot. Isso envolve a simulação de interações do usuário, identificação de pontos fracos na compreensão da linguagem e correção de respostas inadequadas. Os ajustes finos são necessários para garantir que o chatbot forneça respostas úteis e coerentes.
6. Implementação: uma vez que o chatbot esteja funcionando de forma satisfatória, ele pode ser implementado em produção. Isso pode ser feito em várias plataformas, como sites, aplicativos móveis ou serviços de mensagens instantâneas, dependendo do público-alvo.

### 3.1 Chatbots Utilizando Rasa

A escolha por utilizar o Rasa, foi baseada na sua estrutura de código aberto para desenvolver assistentes de IA e chatbots, bem como a sua ampla utilização. O Rasa oferece uma plataforma versátil que permite aos desenvolvedores criarem chatbots altamente personalizados, capazes de lidar com a lógica de negócios específica da aplicação sendo desenvolvida. Nesta seção, descreveremos as ferramentas utilizadas do Rasa e a integração com o LLM selecionado.

Uma das principais características do Rasa é sua capacidade de lidar com a interação do usuário de maneira inteligente, devido a sua *Natural Language Understanding* (NLU). A NLU do Rasa é treinada para identificar as intenções do usuário e extrair elementos relevantes das mensagens. Isso permite que o chatbot compreenda as perguntas

e solicitações dos usuários, fornecendo respostas mais precisas e relevantes. Por exemplo, abaixo é apresentado um trecho de um dataset para treinamento da NLU, com sentenças utilizadas para classificação da intenção de cumprimento do usuário:

```
- intent: cumprimento
  examples: |
    - Olá
    - Boa tarde!
    - Bom dia!
    - Oi!
```

Pode-se também extrair informações relevantes das sentenças interagidas pelo usuário de acordo com a necessidade da aplicação, por exemplo, extrair o nome do usuário, através de uma entidade `nome`, por exemplo, após o chatbots solicitar essa informação através da interação "Para prosseguirmos, preciso saber seu nome. Qual é o seu nome?", seguindo com a intenção `informar_nome`:

```
- intent: informar_nome
  examples: |
    - Meu nome é [Jonas] (nome)
    - [Maria] (name)
    - Ah sim, meu nome é [Thiago] (nome)
    - sim, é [Alex] (nome)
```

Após o reconhecimento da intenção do usuário, e entidades associadas a interação recebida, por parte da NLU, o Rasa core processa essa informação, definindo o fluxo de conversação, selecionando as ações do chatbot em resposta aquela interação do usuário. Respostas do chatbot são codificadas no formato de ações, que podem ser simples, ou seja, um mapeamento direto para uma lista de textos que podem ser oferecidos como resposta, ou podem ser implementadas como métodos personalizáveis implementados pelo desenvolvedor do chatbot, os quais são implementados na linguagem de programação Python, oferecendo todos os recursos de bibliotecas externas, etc.

Para desenvolver este projeto, foram criadas duas ações personalizadas *ActionGetKeywords* e *ActionAnswerQuestion*. Essas ações personalizadas são utilizadas para implementar o método de RAG, ou seja, para montar o prompt usado para requisições na API OpenAI. *ActionGetKeywords* é usada para obter palavras-chave relevantes da pergunta do usuário. Enquanto a *ActionAnswerQuestion* é responsável por realizar a requisição para a API da OpenAI e formular respostas com base nas informações disponíveis.

A integração da API da OpenAI com o Rasa é um elemento fundamental deste projeto. Enquanto o Rasa lida com a interação do usuário e a coleta de palavras-chave relevantes, através do método de RAG para geração do prompt, a API da OpenAI entra em cena para gerar respostas baseadas nas informações disponíveis. Isso permite que o assistente forneça respostas de melhor qualidade, enriquecendo a experiência do usuário. Mais detalhes serão descritos sobre a construção do prompt e API na próxima seção, que descreve a integração do LLM e Rasa.

Em resumo, o uso do Rasa e a integração com a API OpenAI representam uma abordagem eficiente para a criação de chatbots inteligentes e assistentes de IA altamente



personalizados. O Rasa lida com a compreensão da linguagem natural e a coleta de informações relevantes, enquanto a API OpenAI complementa fornecendo respostas mais humanizadas. Essa combinação oferece um grande potencial, automatizando tarefas e fornecendo assistência personalizada de forma eficaz.

## 3.2 OpenAI API

Nesta seção serão descritos aspectos sobre a API OpenAI, que é uma ferramenta que permite a interação com modelos de linguagem avançados, como o GPT-3.5, para gerar respostas inteligentes a partir de prompts personalizados. A engenharia de prompt desempenha um papel crucial no aproveitamento dessa tecnologia, pois ajuda a orientar o modelo na geração de respostas relevantes e precisas.

Para fazer uso da API, o primeiro passo foi configurá-la, juntamente com o modelo a ser usado. A chave da API é necessária para autenticar as solicitações à API da OpenAI, já o modelo é a linguagem que será usado para gerar as respostas. Com o objetivo de integrar o Rasa e a API, foram desenvolvidas ações específicas dentro da estrutura do Rasa para realizar a montagem de prompt através do método RAG descrito, bem como consultar a API e montar a resposta ao usuário. A primeira das ações implementadas foi a ação para obter palavras-chave. A classe *ActionGetKeywords* define uma ação personalizada que extrai palavras-chave de uma pergunta feita pelo usuário. A pergunta é passada como um prompt para a API da OpenAI, que retorna uma lista de palavras-chave relevantes. Essas palavras-chave são então armazenadas em um slot chamado “*keywords*”. Então, foi implementada também a ação para responder à pergunta. A classe *ActionAnswerQuestion* define outra ação personalizada que usa as palavras-chave extraídas para responder à pergunta do usuário. A pergunta e o texto com as informações de domínio são passados como um prompt para a API da OpenAI, que retorna uma resposta baseada no texto.

O sistema projetado usa a API para realizar diferentes tarefas usando modelos de prompts pré-definidos que incluem instruções a serem seguidas para direcionar o LLM alcançar a resposta final e evitar respostas indesejadas ou incorretas. Junto disso foi criado um arquivo 'texto' diretamente no código, com informações do Código de Trânsito Brasileiro - Lei Nº 9.503 (BRASIL, 1997), onde a API irá extrair os dados e responder a pergunta do usuário. Com isso, foi definido algumas regras no prompt para direcionar o LLM em suas respostas. O modelo armazena palavras-chaves relevantes de acordo com a entrada do usuário, busca a(s) palavra(s)-chave(s) no arquivo 'texto', para assim retornar a resposta correspondente a pergunta realizada.

Outros pontos também foram abordados na montagem do prompt, como: (i) Respostas somente em português; (ii) Respostas objetivas; (iii) Sinônimos das palavras-chaves; entre outros. Uma ilustração da estratégia utilizada para montar o prompt da aplicação proposta neste trabalho pode ser observada na Figura 3. O intuito com essas instruções foi para corrigir e evitar respostas que estavam sendo geradas em inglês, respostas que não estavam de acordo com a pergunta ou estavam sendo interpretadas de forma errada pela API. Dessa forma, a engenharia de prompt baseada em RAG permite melhorar o desempenho no uso de um LLM. Ela permite que os desenvolvedores orientem o modelo de forma eficaz e criem sistemas inteligentes capazes de fornecer respostas precisas e contextualmente relevantes para uma variedade de aplicações.

Fonte: O Autor.

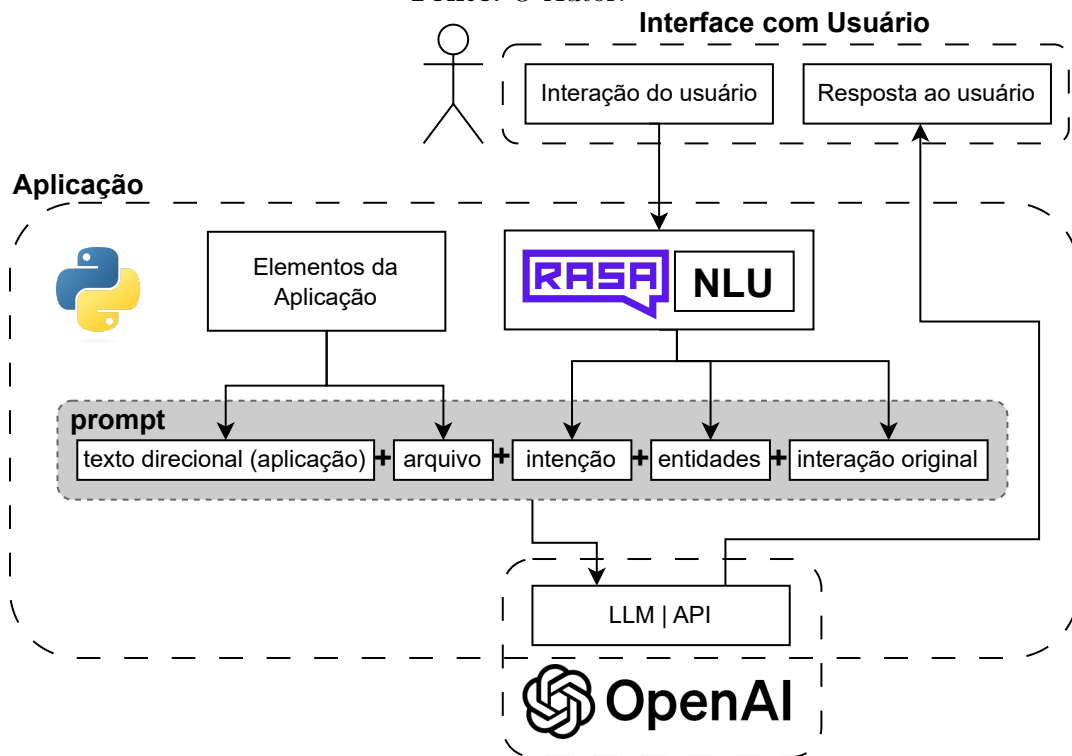


Figura 3 – Diagrama de Construção do Prompt.

### 3.3 Avaliação no domínio do CTB

Nesta seção será apresentada a avaliação realizada sobre o domínio do CTB, exemplificando a integração proposta através de um estudo de caso. A avaliação segue uma apresentação das intenções utilizadas para treinar a NLU do Rasa, seguida pela entrada fornecida pelo usuário em interações de teste, com a respectiva resposta fornecida pelo chatbot.

#### 3.3.1 Exemplo 1

Neste primeiro exemplo, o usuário consulta o chatbot em relação a gravidade associada à dirigir sem cinto de segurança. O trecho abaixo apresenta alguns dos exemplos fornecidos à NLU do Rasa para reconhecer essa intenção do usuário:

```
- intent: gravidade_cinto
examples: |
  - Qual a gravidade por dirigir sem cinto de segurança?
  - Quais são as penalidades por não usar o cinto de segurança?
  - Diga qual a punição por não utilizar o cinto?
  ...
  - Qual a punição por não usar o cinto de segurança?
  - Qual a multa por dirigir sem cinto de segurança?
```

No primeiro exemplo de interação, o usuário realiza a seguinte interação com chatbot, que em seguida já fornece a resposta a interação do usuário:

Your input -> Qual a gravidade por dirigir sem cinto de segurança?

A gravidade por dirigir sem cinto de segurança é grave. A penalidade é uma multa. Medida administrativa é a retenção do veículo até a colocação do cinto pelo infrator.

### 3.3.2 Exemplo 2

No segundo exemplo, temos a mesma intenção sendo classificada pela NLU do rasa, seguindo a modelagem que foi apresentada no exemplo anterior. Porém, a segunda pergunta dá um foco no aspecto de penalidade por dirigir sem cinto de segurança, como segue:

Your input -> Diga qual seria a penalidade por não usar o cinto?

A penalidade por não usar o cinto é grave. A multa é de R\$ 195,23 e 5 pontos na CNH. O veículo será retido até que o cinto seja colocado pelo infrator.

### 3.3.3 Exemplo 3

No terceiro exemplo apresentado, a interação foi mantida com foco na intenção relacionada a gravidades que a lei prevê para a não utilização do cinto. Porém, nessa interação a pergunta apresenta erros de escrita, como segue:

Your input -> Qula seria a puniciao por dirigir sem cinnto de segurança?

A punição por dirigir sem cinto de segurança é considerada grave. A penalidade é uma multa. O veículo pode ser retido até que o cinto seja colocado pelo infrator. É importante seguir as normas de segurança para garantir a segurança de todos.

Percebe-se que nos três exemplos acima, mesmo com uma diversidade de interações, incluindo erros de escrita, o chatbot consegue identificar a pergunta e responder corretamente, com precisão.

### 3.3.4 Exemplo 4

Como no anterior, o quarto exemplo segue a mesma premissa mas com uma mudança pontual na escrita, como é demonstrado a seguir:

Your input -> Qual a gravidade por dirigir sem sinto de segurança?

A infração por dirigir sem sentido de segurança é gravíssima. A penalidade é multa (três vezes) - R\$ 293,47 e 7 pontos na CNH. Além disso, há medidas administrativas como a retenção do veículo até a apresentação de um condutor habilitado

Neste caso, foi colocado a palavra "cinto" com a letra "s" (sinto), ou seja, houve uma confusão por parte do bot entre a palavra "cinto" e "sinto", resultando em uma resposta incorreta. Embora no contexto da pergunta, o termo "sinto" esteja errada, a palavra em si, que vem do verbo "sentir", não está. Como mencionado anteriormente na Seção 2.3, apesar de serem ferramentas eficazes no campo do processamento de linguagem natural, os LLMs podem enfrentar dificuldades em certos casos, levando a respostas incorretas.

### 3.3.5 Exemplo 5

Neste exemplo, é apresentada uma demonstração de interação relacionada a parar o carro sobre a faixa de segurança. Abaixo é apresentada parte do código da NLU, com exemplos de interações que reconhecem essa intenção do usuário em relação a sua interação:

```
- intent: gravidade_fs
examples: |
- Qual a gravidade por parar em cima da faixa de segurança?
- Qual a gravidade por parar em cima da faixa de pedestres?
- Qual a multa por parar o carro em cima da faixa de segurança?
- ...
- Qual a punição por deixar o carro em cima da faixa de pedestres?
- O que acontece se eu deixar o carro sobre a faixa de pedestres?
```

Abaixo é apresentado um exemplo de pergunta interagida com o chatbot, no contexto da intenção acima, e a resposta apresentada pelo chatbot:

```
Your input -> Estou dirigindo e acabo parando o carro em cima da faixa de pedestres, tem alguma punição para isso?
```

```
Ou seja, você está parando seu carro na faixa de pedestres, que é uma infração média. A punição por isso é uma multa de R$130,16 e 4 pontos na CNH. É importante seguir as regras de trânsito para garantir a segurança de todos.
```

### 3.3.6 Exemplo 6

Abaixo é apresentado um trecho do arquivo de treinamento da NLU, utilizado para reconhecer a intenção do usuário em questionar sobre corridas de carros:

```
- intent: gravidade_corrida
examples: |
- Qual a gravidade por disputar corrida?
- Quais são as penalidades por disputar corrida?
- Qual a multa por disputar uma corrida na estrada?
...
- Qual a punição por fazer corrida na estrada?
- O que acontece se eu disputar uma corrida na estrada?
```

Neste exemplo, o usuário questiona sobre ser permitido disputar corridas de carro, recebendo a seguinte resposta do chatbot:

Your input -> Posso disputar uma corrida na estrada?

Você pode disputar uma corrida na estrada, mas é importante lembrar que isso é ilegal e pode ser consequências graves. A infração é considerada "gravíssima" e pode resultar em uma multa de 10 vezes o valor máximo, suspensão do direito de dirigir e apreensão do veículo. Além disso, a aplicação da medida administrativa de recolhimento do documento de habilitação e remoção do veículo pode ocorrer. Em caso de reincidência no período de 12 meses, a multa será duplicada.

### 3.3.7 Exemplo 7

Neste exemplo, é demonstrado a intenção do usuário em obter informações relacionadas a avançar o sinal vermelho.

```
- intent: gravidade_sv
examples: |
  - Qual a gravidade por ultrapassar o sinal vermelho?
  - Quais são as penalidades por ultrapassar o sinal vermelho?
  - Se eu passar no sinal vermelho?
  ...
  - Qual a punição por ultrapassar o sinal vermelho?
  - Multa por ultrapassar o sinal vermelho?
```

Abaixo é apresentado um exemplo de interação realizada com o chatbot e sua respectiva resposta:

Your input -> Qual a multa por ultrapassar o sinal vermelho?

A multa por ultrapassar o sinal vermelho é gravíssima, com uma multa no valor de R\$293,47 e 7 pontos na CNH. É importante seguir as regras de trânsito e respeitar os sinais para garantir a segurança de todos os viajantes.

Em suma, a avaliação no domínio do CTB evidencia a eficácia do chatbot em compreender uma variedade de interações dos usuários, mesmo diante de desafios como erros de escrita em alguns casos. O chatbot no geral, demonstrou consistência ao fornecer respostas precisas e contextualmente adequadas, capaz de recuperar as informações corretas do CTB, elaborando uma resposta bastante humanizada.

## 3.4 Avaliação Comparativa com o ChatGPT 3.5

Nesta seção, serão apresentadas as respostas geradas pela aplicação ChatGPT 3.5 <sup>6</sup>, utilizando algumas das perguntas anteriormente empregadas na seção anterior, para meios de comparação.

---

<sup>6</sup> <<https://chat.openai.com/>>

### 3.4.1 Exemplo 1

No exemplo 1, a entrada do usuário é referente a pergunta sobre dirigir sem cinto de segurança.

Entrada -> Qual a gravidade por dirigir sem cinto de segurança?

ChatGPT -> Dirigir sem cinto de segurança é extremamente perigoso e pode ter consequências graves em caso de acidente. O cinto de segurança é um dispositivo fundamental de segurança veicular projetado para proteger os ocupantes do veículo em colisões e paradas abruptas.

Ao não utilizar o cinto de segurança, os ocupantes ficam mais suscetíveis a lesões sérias ou fatais em caso de acidente. Em uma colisão, o cinto de segurança ajuda a manter os ocupantes no lugar, impedindo que sejam lançados para fora do veículo ou colidam com as partes internas do carro.

As leis de trânsito em muitos países exigem o uso do cinto de segurança, e não utilizá-lo pode resultar em multas e penalidades. Além disso, é importante reconhecer a responsabilidade pessoal em relação à segurança no trânsito e tomar medidas para proteger a si mesmo e aos outros.

Em resumo, a gravidade de dirigir sem cinto de segurança está relacionada ao risco significativamente aumentado de lesões graves ou fatais em caso de acidente. Portanto, é crucial usar o cinto de segurança em todos os momentos enquanto estiver em um veículo.

### 3.4.2 Exemplo 2

No exemplo 2, é referente sobre a parar o carro em cima da faixa de pedestres.

Entrada -> Estou dirigindo e acabo parando o carro em cima da faixa de pedestres, tem alguma punição para isso?

ChatGPT -> Sim, parar o carro em cima da faixa de pedestres é uma infração de trânsito em muitos lugares e pode resultar em multas e outras penalidades. Essas regras variam de acordo com a legislação de trânsito de cada país ou região, por isso é importante verificar as leis locais específicas.

Normalmente, estacionar ou parar em cima da faixa de pedestres é considerado uma infração grave, pois compromete a segurança dos pedestres. As penalidades podem incluir multas financeiras, pontos na carteira de motorista, além de possíveis medidas educativas, como a participação em cursos de reciclagem.

Para evitar infrações e garantir a segurança no trânsito, é fundamental estar ciente e respeitar as regras de trânsito locais, incluindo aquelas relacionadas ao estacionamento e parada em áreas destinadas aos pedestres.

Como se pode observar, o ChatGPT produz respostas significativamente mais extensas em comparação com a abordagem proposta neste trabalho, o qual passou por um processo de tratamento de informações para gerar respostas mais curtas e objetivas.

### 3.4.3 Exemplo 3

Neste exemplo, será adotada uma abordagem diferente, proporcionando ao ChatGPT algumas instruções adicionais, conforme utilizado no presente trabalho.

Entrada -> Responda de forma objetiva, em português, lei brasileira sobre: Qual a multa por ultrapassar o sinal vermelho?

ChatGPT -> Ultrapassar o sinal vermelho é uma infração de trânsito considerada gravíssima, de acordo com o Código de Trânsito Brasileiro (Lei 9.503/1997). A multa para essa infração é multiplicada por três, resultando em uma penalidade financeira mais elevada. Além disso, são aplicados sete pontos na Carteira Nacional de Habilitação (CNH), e pode haver a suspensão do direito de dirigir.

Ao receber informações adicionais, observou-se que o ChatGPT conseguiu gerar respostas mais curtas e objetivas, embora ainda com dificuldades para encontrar certas informações que foram solicitadas, como o valor da multa. Além disso, a resposta do ChatGPT apresentou uma imprecisão ao afirmar que "A multa para essa infração é multiplicada por três". O Art. 208 (BRASIL, 1997) que trata sobre ultrapassar o sinal vermelho, não prevê um fator multiplicador no valor da multa.

## 4 Trabalhos Relacionados

No trabalho apresentado em (TRAD, 2023), os autores propõem um sistema de diálogo conversacional para consultas nutricionais que integra ferramentas do Rasa, ChatGPT e API Edamam<sup>7</sup> (uma plataforma que oferece informações relacionadas à alimentação), que segundo o autor, tem como objetivo criar um assistente de conversação que oferece uma abordagem flexível, modular e que possa fornecer recomendações personalizadas, lidar com diálogos e oferecer respostas conscientes do contexto.

No trabalho apresentado em (VALADÉS, 2023), os autores apresentam um agente conversacional integrando o Rasa, o modelo GPT-3.5-turbo da OpenAI e a ferramenta Docker-Compose<sup>8</sup> (ferramenta que serve para facilitar o processo de definição, configuração e execução de aplicativos multi-container em ambientes). O trabalho visa explorar tarefas específicas de Geração de Linguagem Natural (NLG). Segundo o próprio autor, o trabalho entrega um agente conversacional de suporte ao aprendizado, para os cursos de Engenharia da Computação e Matemática da Universidade de Barcelona, que projeta auxílio ao aluno com dúvidas sobre o plano de ensino, disciplinas, conceitos gerais e também entrega exercícios para os alunos resolverem.

No trabalho apresentado em (GALLO; MALIZIA; PATERNÒ, 2023), os autores apresentam uma abordagem de um agente conversacional para a criação de regras de

---

<sup>7</sup> <<https://www.edamam.com/>>

<sup>8</sup> <<https://docs.docker.com/compose/>>

gatilho-ação e controle de objetos inteligentes em ambientes inteligentes, como por exemplo uma casa inteligente. O trabalho utiliza a integração do ChatGPT para geração de diálogo, com o Rasa para lidar com intenções e entidades.

## 5 Conclusão

Em conclusão, este trabalho destaca a crescente importância da integração de tecnologias chatbots e LLMs, como a API da OpenAI, na criação de soluções de interação com usuários. O chatbot desenvolvido com a integração do Rasa e da API OpenAI apresenta um potencial significativo para melhorar a interação entre usuários e a tecnologia, oferecendo respostas mais precisas e humanizadas. A proposta descrita nesse documento ainda fornece alternativas para o desenvolvimento de tecnologias que podem suportar diferentes tipos de negócios.

Na abordagem proposta, o Rasa lida com a compreensão da linguagem natural, identificando as intenções dos usuários e extraíndo informações relevantes, enquanto a API da OpenAI entra em cena para gerar respostas mais humanizadas e informativas. Essa combinação oferece um potencial significativo para melhorar a interação entre a tecnologia e usuários, podendo ser utilizado para implementar soluções com o objetivo de automatizar tarefas e fornecer assistência personalizada de forma eficaz.

Além da proposta de integração de ferramentas entre chatbots e LLMs, também foi proposto a aplicação de um chatbot baseado na legislação brasileira, especificamente no CTB - Lei nº 9.503/1997. O estudo foi desafiador devido às complexidades de documentos de leis. Em termos da aplicação, ela poderia, futuramente, ser utilizada como meio informativo, o que seria de grande importância, fornecendo conhecimento sobre as leis de trânsito para motoristas e pedestres. Nesse contexto, os acidentes de trânsito são uma questão relevante na área da saúde, resultando em um número alarmante de fatalidades e lesões. Portanto, um chatbot que possa esclarecer dúvidas sobre as infrações de trânsito ou em qualquer outra área jurídica brasileira, pode desempenhar um papel fundamental na disseminação do conhecimento das leis do país.

No entanto, a criação de um chatbot como esse não está isenta de desafios. A integração de tecnologias, o treinamento de modelos de linguagem natural e a garantia de que o chatbot forneça informações precisas são desafios que precisam ser superados. Além disso, a manutenção contínua e a atualização do chatbot para refletir mudanças na legislação são igualmente importantes. O chatbot precisa ser preciso e informativo, fornecendo respostas que estejam alinhadas com a legislação vigente. Isso demonstra como a tecnologia pode ser usada para disseminar conhecimento e simplificar o acesso à informação legal.

Em resumo, a integração entre o Rasa e a API da OpenAI para criar um chatbot com foco na legislação de trânsito brasileira é um exemplo notável de como a tecnologia pode ser usada para simplificar a interação entre sistemas computacionais e seres humanos. Essa abordagem abre portas para o desenvolvimento de chatbots eficazes em diferentes áreas e demonstra o potencial da inteligência artificial na simplificação de processos e na melhoria da qualidade de interação com os usuários. É importante que os desenvolvedores e pesquisadores continuem a explorar essas tecnologias e a aprimorá-las para beneficiar a sociedade como um todo.



## 5.1 Trabalhos Futuros

O desenvolvimento e a integração de chatbots com tecnologias de processamento de linguagem natural, como o Rasa e LLMs, apresentam um campo fértil para trabalhos futuros e aprimoramentos. O chatbot focado na legislação de trânsito brasileira, explorado neste trabalho, é apenas um exemplo do potencial que essa combinação de ferramentas oferece. À medida que a evolução na compreensão e aplicação da inteligência artificial continua, algumas áreas específicas merecem atenção para futuras pesquisas e desenvolvimentos:

- **Aprimoramento da integração Rasa e OpenAI:** a integração do Rasa com a API da OpenAI pode ser aprimorada para garantir respostas ainda mais precisas e contextualmente relevantes. Isso pode envolver a otimização dos modelos de linguagem, a análise de contexto mais aprofundada e o treinamento adicional para melhor compreender as nuances das perguntas dos usuários.
- **Adaptação à diferentes legislações:** a criação de chatbots voltados para a legislação não se limita ao trânsito brasileiro. Essa abordagem pode ser expandida para abranger diferentes conjuntos de leis em diferentes países. O desenvolvimento de chatbots que se adaptem automaticamente às regulamentações locais é um desafio significativo, mas também uma oportunidade valiosa para fornecer informações precisas e personalizadas.
- **Acessibilidade e multilinguismo:** garantir que os chatbots sejam acessíveis e capazes de atender a uma variedade de idiomas é fundamental. A inclusão de funcionalidades de tradução e adaptação de idiomas pode ampliar a utilidade desses assistentes virtuais, tornando-os relevantes para audiências globais.
- **Educação e conscientização:** os chatbots podem ser usados não apenas para responder a perguntas, mas também para educar os usuários sobre as leis e regulamentos. Trabalhos futuros podem explorar como esses assistentes virtuais podem ser utilizados para aumentar a conscientização sobre questões legais e melhorar o entendimento das leis.
- **Avaliação da eficácia:** é importante conduzir estudos e pesquisas para avaliar a eficácia desses chatbots na disseminação do conhecimento legal e na prevenção de infrações. Isso envolve a coleta de dados sobre o uso, a compreensão do usuário e a análise de métricas de impacto. A coleta de *feedback* dos usuários é fundamental para a melhoria contínua dos chatbots. A implementação de sistemas de *feedback* e análise dessas informações pode levar a aprimoramentos significativos na interação e no desempenho do chatbot.
- **Aplicação em outros setores jurídicos:** além do trânsito, chatbots podem ser desenvolvidos para fornecer suporte em outros setores legais, como direito civil, direito trabalhista, direito criminal e muitos outros. Cada área possui suas próprias complexidades, tornando essa uma área vasta para pesquisas futuras.

Por fim, destaca-se que a integração entre os frameworks para desenvolvimento de chatbots baseados em regras e LLMs oferece muitas oportunidades. Esses assistentes virtuais têm o potencial de simplificar a compreensão de documentos complexos, como por exemplo documentos legais e regulatórios. Podendo ainda melhorar a interação entre sistemas e usuários. À medida que essas tecnologias continuem a ser exploradas, é importante manter

o foco na melhoria da precisão, na segurança e na acessibilidade, para que eles possam ser eficazes em uma variedade de contextos, incluindo culturais. O campo dos chatbots e da inteligência artificial continua a evoluir, e os trabalhos futuros desempenharão um papel vital na realização desse potencial.

## Referências

- ABREU, D. R. d. O. M.; SOUZA, E. M. d.; MATHIAS, T. A. d. F. Impacto do código de trânsito brasileiro e da lei seca na mortalidade por acidentes de trânsito. CSP, 2020. Citado na página [3].
- BANSAL, H.; KHAN, R. A review paper on human computer interaction. *International Journal of Advanced Research in Computer Science and Software Engineering*, v. 8, n. 53, 2018. Citado na página [3].
- BRASIL. *LEI Nº 9.503, DE 23 DE SETEMBRO DE 1997*: Institui o código de trânsito brasileiro. 1997. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/L9503.htm](http://www.planalto.gov.br/ccivil_03/leis/L9503.htm)>. Acesso em: Novembro de 2023. Citado (4) vezes nas páginas [3, 8, 13 e 19].
- BRASIL. *Lei complementar Nº 95, de 26 de fevereiro de 1998*: Dispõe sobre a elaboração, a redação, a alteração e a consolidação das leis, conforme determina o parágrafo único do art. 59 da constituição federal, e estabelece normas para a consolidação dos atos normativos que menciona. 1998. Disponível em: <[https://www.planalto.gov.br/ccivil\\_03/leis/lcp/lcp95.htm](https://www.planalto.gov.br/ccivil_03/leis/lcp/lcp95.htm)>. Acesso em: Novembro de 2023. Citado na página [9].
- CHEN, Z. et al. Exploring the potential of large language models (llms) in learning on graphs. *arXiv preprint arXiv:2307.03393*, 2023. Citado na página [6].
- CHOWDHARY, K. Natural language processing. *Fundamentals of artificial intelligence*, Springer, p. 603–649, 2020. Citado na página [6].
- CUSTÓDIO, M. d. S. et al. Rasa4jaca: Uma interface entre sistemas multiagentes e tecnologias chatbots open source. *12th Brazilian Conference on Intelligent Systems*, 2022. Citado na página [4].
- ENGELMANN, D. et al. Dial4jaca—a communication interface between multi-agent systems and chatbots. *International conference on practical applications of agents and multi-agent systems*, p. 77–88, 2021. Citado na página [5].
- ENGELMANN, D. C. et al. A conversational agent to support hospital bed allocation. *Brazilian Conference on Intelligent Systems*, p. 3–17, 2021. Citado na página [5].
- ENGELMANN, D. C. et al. Maids-a framework for the development of multi-agent intentional dialogue systems. *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, p. 1209–1217, 2023. Citado na página [5].
- FAN, W. et al. Recommender systems in the era of large language models (llms). *arxiv 2023. arXiv preprint arXiv:2307.02046*, 2023. Citado na página [6].
- GALLO, S.; MALIZIA, A.; PATERNÒ, F. Towards a chatbot for creating trigger-action rules based on chatgpt and rasa. *9th International Symposium on End-User Development*, 2023. Citado na página [19].
- HENRIQUE, C. R. d. C. Custos dos acidentes de trânsito no brasil: Estimativa simplificada com base na atualização das pesquisas do ipaea sobre custos de acidentes nos aglomerados urbanos e rodovias. IPEA, 2020. Citado na página [3].

- KADDOUR, J. et al. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023. Citado na página [6].
- KASNECI, E. et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, Elsevier, v. 103, p. 102274, 2023. Citado na página [6].
- KATZ, D. M. et al. Natural language processing in the legal domain. *arXiv preprint arXiv:2302.12039*, 2023. Citado na página [8].
- LEWIS, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, v. 33, p. 9459–9474, 2020. Citado na página [7].
- LIU, N. F. et al. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023. Citado na página [7].
- MCTEAR, M.; MAROKKIE, S. V.; BI, Y. A comparative study of chatbot response generation: Traditional approaches versus large language models. In: SPRINGER. *International Conference on Knowledge Science, Engineering and Management*. [S.l.], 2023. p. 70–79. Citado (2) vezes nas páginas [3 e 9].
- PIERACCINI, R. *The Voice in the Machine: Building Computers that Understand Speech*. [S.l.]: MIT Press, 2012. (Computer Science: Language). Citado na página [3].
- RESEARCH, G. V. *Chatbot Market Size, Share Trends, Analysis Report By Application (Customer Services, Branding & Advertising), By Type, By Vertical, By Region (North America, Europe, Asia Pacific, South America), And Segment Forecasts, 2023 - 2030*. 2021. Disponível em: <<https://www.grandviewresearch.com/industry-analysis/chatbot-market>>. Acesso em: Novembro de 2023. Citado na página [4].
- SHARMA, R. K.; JOSHI, M. An analytical study and review of open source chatbot framework, rasa. *International Journal of Engineering Research*, v. 9, n. 06, 2020. Citado na página [11].
- SHAWAR, B.; ATWELL, E. Chatbots: Are they really useful? *LDV Forum*, v. 22, p. 29–49, 2007. Citado na página [3].
- TRAD, A. A conversational dialogue system for nutrition and search queries. 2023. Citado na página [19].
- VALADÉS, P. S. Agent educatiu multiassignatura: disseny i implementació amb rasa i docker. 2023. Citado na página [19].
- ZHAO, R. et al. Retrieving multimodal information for augmented generation: A survey. *arXiv preprint arXiv:2303.10868*, 2023. Citado (3) vezes nas páginas [7, 8 e 10].