Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

DUBI: um framework para avaliação automática de chatbots

José Ronaldo Agra de Souza Filho

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Jacir Luiz Bordim

Brasília
2024

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

S729d

Souza Filho, José Ronaldo Agra de
DUBI: um framework para avaliação automática de chatbots / José Ronaldo Agra de Souza Filho; orientador Jacir Luiz Bordim. -- Brasília, 2024.
138 p.

Dissertação(Mestrado Profissional em Computação Aplicada) -- Universidade de Brasília, 2024.

1. qualidade de chatbot. 2. avaliação automática de chatbot. 3. avaliação estática. 4. avaliação interativa. 5. testes automatizados. I. Bordim, Jacir Luiz, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

DUBI: um framework para avaliação automática de chatbots

José Ronaldo Agra de Souza Filho

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Jacir Luiz Bordim (Orientador)
CIC/UnB

Prof. Dr. Marcelo Ladeira
CIC/UnB

Prof. Dr. Hilário Tomaz
IFES

Prof. Dr. Gladston Luiz da Silva
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 06 de agosto de 2024

Dedicatória

A Lívia, Maria Eduarda e Gabriel. O apoio incondicional, inspiração e amor de vocês foram fundamentais para que eu pudesse alcançar este sonho. Só cheguei até aqui porque estavam comigo. Este trabalho é dedicado a vocês, com todo meu amor e gratidão.

Agradecimentos

Ao final da jornada do mestrado, o título obtido é individual, mas ele é resultado de um trabalho em conjunto. Por isso, gostaria de expressar meus sinceros agradecimentos a todos que estiveram comigo nesse período.

Primeiramente, agradeço a Deus, pelas bênçãos derramadas sobre mim e meus próximos, proporcionando a paz necessária para a realização deste trabalho.

Agradeço à minha família, pelo apoio incondicional, compreensão e incentivo contínuo para a realização do mestrado. Um agradecimento especial aos meus filhos, que me motivam e inspiraram o nome DUBI, e a minha esposa, que esteve de mãos dadas comigo desde o primeiro momento dessa caminhada.

Ao Serpro, meu agradecimento por permitir que eu dedicasse preciosas horas de trabalho ao mestrado e por incentivar a geração de pesquisa aplicada no contexto da empresa.

Ao meu orientador, sou imensamente grato por toda a ajuda durante esta jornada, pelas valiosas conversas, dicas e sugestões de melhoria, e, principalmente, por me dar liberdade para conduzir a pesquisa conforme minhas ideias.

Agradeço também aos professores do programa de mestrado, por compartilharem um pouco de sua sabedoria e experiência, proporcionando uma base sólida e inspiração para seguir em frente.

Aos colegas da turma 01/2022 do PPCA, obrigado pelas conversas, incentivos e pelos almoços das sextas-feira, que ajudaram a tornar esse caminho mais leve.

E aos amigos do Serpro, agradeço pelo compartilhamento de conhecimento, apoio e motivação para realizar um trabalho aplicável ao nosso contexto de trabalho.

A todos, meu muito obrigado.

Resumo

A proliferação da inteligência artificial impulsiona a adoção de *chatbots*, sistemas conversacionais projetados para automatizar interações com usuários. No entanto, avaliá-los representa um desafio complexo e que frequentemente depende da intervenção humana, tornando-se impraticável em larga escala. Uma revisão do estado da arte indicou que duas abordagens de avaliação são utilizadas: estática e interativa. A primeira examina a modelagem do assistente virtual, enquanto a última interage com o sistema para avaliar seu desempenho. No entanto, foi observado que falta um método que combine ambas as avaliações, algo crucial para o diagnóstico completo do sistema. Nesse contexto, este estudo apresenta o *framework* DUBI, acrônimo para *Design Understanding* (DU) e *chatBot Intelligence* (BI), como um meio de avaliar automaticamente *chatbots*, cobrindo seus componentes estáticos e interativos. O DUBI é um avanço em comparação aos métodos existentes, pois permite a avaliação contínua do desempenho dos assistentes virtuais e fornece recomendações objetivas para aprimorar sua estrutura, que podem ser usadas como base para intervenções. O módulo de avaliação estática mede uma série de métricas e indica quais áreas exigem melhorias na modelagem do *chatbot*. A avaliação interativa utiliza grandes modelos de linguagem para criar casos de teste a partir do conteúdo de treinamento do *chatbot* e analisa seu desempenho após a execução desses testes. O procedimento automatizado é o diferencial do DUBI, pois reduz a variabilidade e o viés da avaliação humana, ao mesmo tempo em que economiza tempo e recursos. Um experimento com assistentes virtuais reais foi realizado para validar o DUBI. As descobertas evidenciaram que os aprimoramentos sugeridos pelo DUBI levou a avanços substanciais nas medidas de desempenho. Especificamente, um dos *chatbots* avaliados teve um aumento notável de 55% na acurácia e uma redução impressionante de 89% na taxa de *fallback*. Os resultados comprovam a eficácia do DUBI em identificar deficiências na modelagem e propor aprimoramentos tangíveis. Este trabalho contribui para a literatura ao integrar avaliações estáticas e interativas, fornecendo uma ferramenta para melhorar a qualidade de *chatbots*, o que possibilita reduzir riscos financeiros ou de reputação.

Palavras-chave: qualidade de *chatbot*, avaliação estática, avaliação interativa, testes automatizados

Abstract

The proliferation of artificial intelligence is driving the adoption of chatbots, which are conversational systems designed to automate user interactions. Nevertheless, evaluating chatbots poses an intricate difficulty that frequently depends on human intervention, rendering it impractical on a large scale. A review of the state of the art indicated that two evaluation approaches have been utilized: static and interactive. The former examines the structure and training content of the virtual assistant, while the latter engages with the system to assess its performance. However, it has been noted that there is a lack of a method that combines both evaluations, which are crucial for a thorough system diagnosis. Within this perspective, this study introduces the DUBI framework, an acronym for *Design Understanding* (DU) and *chatBot Intelligence* (BI), as a means to automatically assess chatbots, covering both their static and interactive components. DUBI offers a notable improvement compared to existing methods, since it enables ongoing assessment of virtual assistants' performance and provides objective recommendations for enhancing their structure, which can be used as a basis for interventions. The static assessment measures a range of metrics and provides feedback on areas that require improvement in the chatbot's modeling. The interactive assessment utilizes large language models to create test cases from the chatbot's training material and analyzes its performance after the execution of these tests. The automated procedure is a key feature of DUBI, since it reduces the variability and bias from human evaluation while saving time and resources. An experiment was done to authenticate DUBI by employing actual virtual assistants. Our findings demonstrated that implementing the enhancements suggested by DUBI led to substantial advancements in performance measures. Specifically, one of the assessed chatbots had a remarkable 55% increase in accuracy and an impressive 89% decrease in the fallback rate. The results clearly showcase the efficacy of DUBI in pinpointing shortcomings in modeling and proposing tangible enhancements. This work contributes to the literature by integrating static and interactive evaluations, providing a tool to improve chatbot quality and reduce financial or reputational risks.

Keywords: *chatbot quality, static evaluation, interactive evaluation, automated testing*

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AIML *Artificial Intelligence Mark-up Language.*

API *Application Programming Interface.*

BI *chatBot Intelligence.*

DU *Design Understanding.*

GPT *Generative Pre-Training.*

IA Inteligência Artificial.

JSON *JavaScript Object Notation.*

MAE *Mean Absolut Error.*

NLU *Natural Language Understanding.*

PLN Processamento de Linguagem Natural.

RI Recuperação de informação.

SERPRO Serviço Federal de Processamento de Dados.

XML *Extensible Markup Language.*

Lista de Símbolos

BET Balanceamento dos exemplos de treinamento.

CE_D Condição de entrada/ativação de um nó de diálogo em um *chatbot*.

D Nó de diálogo de um *chatbot*.

D_F Nó de diálogo em um *chatbot* com a condição de entrada/ativação sempre falsa.

D_T Nó de diálogo testado em um *chatbot*.

D_V Nó de diálogo do *chatbot* com a condição de entrada/ativação sempre verdadeira.

D_{SR} Nó de diálogo do *chatbot* que não possui texto de resposta.

E_T Exemplo de treinamento de uma intenção.

F Frase em um texto.

FK *Flesch–Kincaid grade level*.

FN Falsos negativos.

FP Falsos positivos.

$G - index$ *Gulpease Index*.

I Intenção de um *chatbot*.

i_f Interação onde o *chatbot* precisou utilizar uma estratégia de *fallback*.

I_O Intenção órfã em um *chatbot*.

I_T Intenção testada em um *chatbot*.

L Letra de um texto.

LET Legibilidade dos exemplos de treinamento.

LTR Legibilidade dos textos de resposta.

M_e Quantidade de mensagens com erros de grafia recebidas pelo *chatbot*.

M_r Quantidade de mensagens repetidas ou similares em uma conversa com o *chatbot*.

μ_{ET} Média da quantidade dos exemplos de treinamento de uma intenção.

P Palavra em um texto.

P_R Palavra representativa do idioma em um texto.

QCF Quantidade de ciclos nos fluxos de conversa.

QDF Quantidade de nós de diálogo com condição de entrada sempre falsa.

QDR Quantidade de nós de diálogo sem texto de resposta.

QDV Quantidade de nós de diálogo com condição de entrada sempre verdadeira.

QET Quantidade de exemplos de treinamento.

QID Quantidade de nós de diálogo com condições de entrada iguais.

QIO Quantidade de intenções órfãs.

R Resposta do *chatbot*.

R_c Quantidade de respostas corretas dadas às mensagens recebidas.

R_N Resposta do *chatbot* com sentimento negativo.

RET Representatividade dos exemplos de treinamento.

RTR Representatividade dos textos de resposta.

S Sílaba de uma palavra.

SID Similaridade entre intenções distintas.

STR Sentimento negativo dos textos das respostas.

T_c Taxa de compreensão.

T_CR Taxa de consistência das respostas.

T_f Taxa de *fallback*.

t_R Tempo de resposta de uma interação com o *chatbot*.

TET Tamanho dos exemplos de treinamento.

tm_R Tempo médio de resposta.

TTR Tamanho dos textos de resposta.

VN Verdadeiros negativos.

VP Verdadeiros positivos.

Capítulo 1

Introdução

A recente popularização do uso da Inteligência Artificial (IA), cada vez mais presente no cotidiano das pessoas, impulsiona a evolução dos sistemas capazes de interagir com usuários por meio de conversas automatizadas, conhecidos como *chatbots*. Projetados para oferecer uma experiência de conversação natural, sendo preparados para compreender as intenções dos usuários e responder adequadamente com informações relevantes, estes sistemas possibilitam atender a milhares de usuários ao mesmo tempo [1].

Aliado a esta eficiência para lidar com solicitações em larga escala, os *chatbots*, também chamados de assistentes virtuais, possibilitam um atendimento contínuo e de alta disponibilidade, característica especialmente significativa em um mundo cada vez mais conectado e com horários flexíveis. Esses atributos contribuem para a aplicação abrangente desses sistemas em diversos setores da economia, tais como saúde, educação, turismo e comércio, entre outros [2]. Independente do domínio de negócio, a implementação dos *chatbots* geralmente visa aprimorar o atendimento aos clientes e usuários de serviços [3].

No âmbito específico dos serviços públicos, essa realidade não é diferente. Por um lado, há a necessidade de que determinados serviços governamentais estejam disponíveis ininterruptamente à população, por outro, o Estado Brasileiro enfrenta uma histórica carência de servidores públicos para atender a essa demanda [4]. Diante disso, a utilização de assistentes virtuais nesses serviços pode desempenhar um papel relevante na melhoria do atendimento aos cidadãos, contribuindo para um Estado mais eficiente e para a redução de custos aos cofres públicos. Essa abordagem está em sintonia com a recente política de digitalização adotada pelo governo brasileiro, em especial à Lei 14.129/2021 [5], que dispõe sobre os princípios, regras e instrumentos para o governo digital, priorizando a prestação de serviços digitais por meio do autoatendimento pelo cidadão.

Alinhado a este cenário, o Serviço Federal de Processamento de Dados (SERPRO), empresa pública brasileira de tecnologia da informação, responsável por mais de 90% das soluções digitais do Estado Brasileiro [6], lançou a plataforma Serprobots visando

fornecer uma solução acessível e eficiente para a construção de *chatbots*, capacitando os entes governamentais a otimizarem suas operações e aprimorarem o atendimento dos cidadãos.

A plataforma Serprobots é uma solução tecnológica que visa simplificar a criação e gerenciamento de assistentes virtuais, desde os mais simples até os mais complexos, sem a necessidade de conhecimentos em programação [7]. Essa abordagem permite que equipes de desenvolvimento de *chatbots*, independentemente de suas habilidades de codificação, possam se concentrar nas atividades de modelagem dos fluxos conversacionais e curadoria das informações, cruciais para esses sistemas. No contexto do SERPRO, esta plataforma viabilizou a diminuição do tempo e do custo dos projetos de construção de assistentes virtuais, representando assim um progresso para a empresa no atendimento de demandas deste tipo.

Apesar dos significativos avanços na área de *chatbots*, independente da empresa ou da plataforma utilizada para tal, o desenvolvimento desse tipo de sistema enfrenta desafios diversos. Um dos principais diz respeito à avaliação da qualidade e do desempenho dos assistentes virtuais. Pois, avaliar automaticamente sistemas de diálogo conversacional é um processo complexo e um problema em aberto, dificultado pelas características inerentes a estes tipos de sistemas [8], como a necessidade de fornecer respostas corretas, coerentes e relevantes.

Devido a esta complexidade, tradicionalmente, os assistentes virtuais são avaliados manualmente, por meio de interações de especialistas ou usuários com o *chatbot*, os quais o julgam com base em critérios pré-definidos. Embora essa abordagem possa fornecer percepções valiosas, ela é trabalhosa, demorada e está sujeita a vieses e variabilidade de avaliação [1]. Além disso, com a crescente adoção de *chatbots* em larga escala, a avaliação manual se torna impraticável e insustentável em termos de tempo, custo e eficiência [9].

Visando superar as limitações da avaliação manual, diversos estudos têm buscado propor abordagens automatizadas para a avaliação de assistentes virtuais, como Bravo-Santos *et al.* (2020) [10], Gao *et al.* (2021) [11], Cañizares *et al.* (2022) [12] e Yang *et al.* (2022) [13]. Entretanto, apesar de serem mais eficientes em termos de tempo e recursos, quando comparadas à avaliação manual, as propostas automáticas ainda não possuem um padrão de procedimentos amplamente aceito na indústria e na academia [1]. Esta é uma das razões para alguns estudos focarem em avaliar apenas a estrutura do *chatbot*, enquanto outros priorizam avaliar a interação com o sistema. Por isso, conforme será discutido neste documento, ainda não há uma solução que proporcione uma avaliação abrangente e automatizada para assistentes virtuais.

Portanto, embora a plataforma Serprobots, e outras soluções similares, facilitem o desenvolvimento de *chatbots*, é fundamental reconhecer que a avaliação e otimização desses

sistemas representam desafios e, ao mesmo tempo, oportunidades a serem enfrentados e aproveitados.

Nesse contexto, o presente trabalho visa contribuir para a área de pesquisa relacionada à avaliação de assistentes virtuais, propondo uma abordagem abrangente e automatizada que considere os diversos aspectos envolvidos. O intuito é viabilizar uma forma de mensurar e aprimorar a qualidade e eficácia das interações desses assistentes com os usuários, bem como as estruturas dos fluxos conversacionais. Com essa proposta, busca-se promover uma evolução na plataforma Serprobots, adicionando esta avaliação abrangente e automática como uma funcionalidade disponível a todos os usuários da solução. Entende-se que essa contribuição não apenas beneficiará a empresa fornecedora da plataforma, o SERPRO, mas também enriquecerá a linha de pesquisa sobre avaliação de *chatbots*, preenchendo lacunas existentes, conforme detalhado na próxima seção.

1.1 Motivação

A crescente demanda por *chatbots* impulsionou os fornecedores de serviços de conversação baseados em inteligência artificial (*e.g.* IBM *Watson Assistant* [14], Google *DialogFlow* [15], Amazon *Lex* [16] e RASA NLU [17]) a evoluírem suas ferramentas. No entanto, esta evolução se concentrou em melhorar os respectivos motores de conversação na tarefa de compreender as intenções dos usuários dos assistentes virtuais. Como consequência dessa priorização, esses serviços apresentam limitações significativas em relação a ofertar suporte para garantir a qualidade dos *chatbots* [10, 12].

É interessante notar que algumas empresas desenvolvedoras de *software*, visando melhor atender a seus clientes e padronizar os projetos de assistentes virtuais, decidiram desenvolver suas respectivas plataformas de desenvolvimento de *chatbots* [10]. Estas plataformas, normalmente, utilizam os serviços de IA desses grandes fornecedores de tecnologia, agregando funcionalidades extras que facilitam e aumentam a produtividade dos projetos de desenvolvimento. É o caso da plataforma Serprobots, que, além de permitir a criação e gerenciamento de *chatbots* sem a necessidade de desenvolvimento de código, oferece benefícios como a capacidade de criar assistentes multicanais — que podem interagir com os usuários por meio da *Web*, *WhatsApp* [18], *Facebook Messenger* [19] ou *Twitter Direct Message* [20] — e multimotores, possibilitando a combinação de diferentes motores de conversação em um único *chatbot*.

Em termos de suporte à avaliação de qualidade dos *chatbots*, a Serprobots disponibiliza como funcionalidades padrões as seguintes:

- **Painel de curadoria**: trata-se de um painel pelo qual as equipes de desenvolvimento podem acompanhar o uso de seu assistente virtual pelos usuários, sendo possível extrair dele informações relevantes à curadoria da base de treinamento.
- **Componente de avaliação do *chatbot***: é um componente que pode ser habilitado pelas equipes de desenvolvimento, de acordo com seu interesse. Ao ser habilitado, o usuário tem à sua disposição uma forma de avaliar a conversa que teve com o assistente virtual, informando sua satisfação ou insatisfação com as respostas recebidas. Quando a avaliação é negativa, o *chatbot* solicita que o usuário informe o motivo daquela avaliação. Estas informações são disponibilizadas no painel de curadoria.
- **Testes automatizados**: funcionalidade para criar casos de testes, nos quais as equipes podem esquematizar um conjunto de mensagens a serem enviadas ao *chatbot* e quais são as respostas esperadas, sendo executados de forma automática.

Embora sejam avanços, quando comparadas às ofertas de garantia de qualidade dos serviços de conversação dos principais fornecedores desta tecnologia, estas funcionalidades ofertadas pela plataforma Serprobots ainda caminham na trilha da avaliação manual. A análise das avaliações feitas pelos usuários, através do painel de curadoria, necessita de um profissional para fazê-la. Além disso, apesar de valiosas, as avaliações dos usuários podem indicar tardiamente a presença de erros. Em relação aos testes automatizados, sua automação é parcial, referente a sua execução, mas estes ainda precisam ser criados manualmente por especialistas.

Essa abordagem, a de avaliação realizada por humanos, é a mais comum no cenário de desenvolvimento de *chatbots* [8], trazendo consigo as desvantagens da avaliação manual, como ser demorada, custosa e suscetível a viés. Estas características acarretam desaceleração do ciclo de desenvolvimento destes sistemas [21].

Dessa forma, é evidente a necessidade de uma solução abrangente e automática que possibilite a avaliação precoce dos assistentes virtuais, permitindo a identificação rápida de erros e reduzindo os custos de correção. Diante desse cenário, o SERPRO reconhece a importância da avaliação automática como parte fundamental da melhoria contínua da plataforma Serprobots. E, por isso, há interesse da empresa em ter alternativas para avaliar de forma adequada e célere a qualidade dos *chatbots* construídos na sua plataforma.

Este cenário motivou a proposição de desenvolver uma solução de avaliação automática e abrangente de *chatbots*, capaz de avaliar antecipadamente tanto a estrutura dos assistentes virtuais quanto sua assertividade na interação com os usuários. Essa solução será integrada à plataforma Serprobots, proporcionando recursos aos usuários não apenas para construir seus *chatbots*, mas também para avaliá-los de forma totalmente automá-

tica. Com essa abordagem, será possível identificar pontos de melhoria ainda durante o processo de desenvolvimento, resultando em *chatbots* de melhor qualidade e, consequentemente, em maior satisfação por parte dos usuários.

O sucesso desta proposição será alcançado através da realização dos objetivos descritos na Seção 1.3. Entende-se que esta proposta, além de ser valiosa ao Serviço Federal de Processamento de Dados, proporcionando-o ofertar assistentes virtuais ainda melhores aos seus clientes, será também uma contribuição à academia, preenchendo uma lacuna atualmente existente no estado da arte [1, 8, 22]: a ausência de uma solução que avalie automaticamente *chatbots*, considerando tanto aspectos estáticos quanto interativos.

1.2 Escopo do trabalho

De forma geral, os *chatbots* podem ser categorizados quanto ao domínio de conhecimento (aberto ou fechado) [23], em relação à abordagem de geração das respostas (generativos ou baseados em recuperação de informação) [1, 24] e pela forma como entendem a necessidade dos usuários (regras ou baseados em intenções) [25, 26].

O escopo deste trabalho é delimitado pelos *chatbots* de domínio fechado, baseados em intenções e que utilizam a abordagem de recuperação de informação. Esta delimitação se dá, em partes, em virtude de prioridades estabelecidas pelo SERPRO, financiador parcial deste trabalho. A empresa concentra esforços no aprimoramento e otimização de assistentes virtuais com essas características específicas, através da plataforma Serprobots, visando atender às demandas de seus clientes.

Além disso, conforme apresentado na Seção 1.3, a solução proposta neste trabalho tem em vista automatizar e objetivar a avaliação dos *chatbots*, considerando tanto a estrutura quanto a interação com eles. Para isso, é necessário ter conhecimento prévio dos componentes do assistente virtual, o que não é viável para os de domínio aberto e generativos, pois esses sistemas não possuem uma estrutura predefinida e suas respostas são geradas com base em modelos estatísticos complexos [27], ou seja, não há gabaritos para comparar as respostas.

1.3 Pergunta e objetivos da pesquisa

A presente pesquisa explora a suposição de que melhorias na estrutura e no conteúdo de treinamento de um *chatbot* têm um impacto positivo em seu desempenho durante as interações com os usuários. Além disso, também se especula que seja viável automatizar tanto a avaliação quanto a identificação de sugestão de aprimoramentos na modelagem do assistente virtual, reduzindo ou mesmo dispensando assim a intervenção humana.

Nesse contexto, o objetivo geral deste trabalho é desenvolver uma metodologia de avaliação automática para *chatbots* de domínio fechado e baseados em intenções e recuperação de informação, que englobe tanto aspectos estáticos quanto interativos. A proposta, chamada de DUBI, acrônimo para *Design Understanding* (DU) e *chatBot Intelligence* (BI), visa avaliar a estrutura, o conteúdo de treinamento e as respostas do *chatbot*, para obter métricas de qualidade e desempenho, além de identificar áreas que necessitam de melhorias. Com essa abordagem, pretende-se alcançar uma cobertura de avaliação abrangente, superando as limitações atuais das soluções disponíveis no estado da arte.

Como objetivos específicos, tem-se:

1. Comparar as soluções existentes na literatura com a proposta deste trabalho, a fim de validar se as lacunas em aberto identificadas são atendidas pela proposta DUBI.
2. Implementar um serviço que analise e avalie a estrutura e o conteúdo de treinamento de *chatbots* de forma estática, gerando como resultado um relatório que indique pontos de melhorias.
3. Implementar um serviço que gere automaticamente casos de testes e interaja com *chatbots*, simulando conversas de usuários, visando avaliar o desempenho do sistema quanto a sua assertividade das respostas.
4. Realizar a validação empírica do *framework* DUBI por meio de sua aplicação em *chatbots* reais e pré-existentes, permitindo a análise concreta dos resultados.

1.4 Contribuições do trabalho

Tendo os desenvolvedores de *chatbots* como público-alvo, este trabalho contribui com a literatura atual preenchendo lacunas existentes relacionadas à avaliação abrangente destes sistemas, conforme apresentado no Capítulo 3, realizando aferições de características estáticas e interativas, além de indicar objetivamente pontos de melhorias na modelagem dos *chatbots*.

Ao compartilhar o conhecimento com a comunidade científica por meio de publicações em conferências, este estudo proporciona uma contribuição para o campo de pesquisa sobre avaliação de assistentes virtuais. Neste cenário, destaca-se como marco importante do trabalho a submissão dos resultados preliminares do DUBI na 19ª Conferência Ibérica de Sistemas e Tecnologias de Informação. O artigo intitulado *"Chatbot Design Understanding: a framework for automating chatbot modeling quality assessment"* [28] foi aceito para apresentação, refletindo o reconhecimento da comunidade científica acerca da relevância e do potencial do estudo no domínio da avaliação de *chatbots*.

No âmbito específico do SERPRO, a integração do *framework* DUBI à plataforma Serprobots viabilizará a execução de avaliações automatizadas dos assistentes virtuais desenvolvidos pela empresa. Essa capacidade de avaliação contínua, ainda durante o desenvolvimento, permitirá o aprimoramento progressivo dos *chatbots* e a prevenção da implantação daqueles que ainda não atendam aos requisitos de qualidade estabelecidos.

1.5 Metodologia

Para cumprir com os objetivos descritos anteriormente, o desenvolvimento deste trabalho seguiu as seguintes etapas metodológicas:

1. **Revisão do estado da arte:** foi realizada uma pesquisa em bases de dados científicas para compreender estudos existentes sobre a avaliação de *chatbots*. Com base nesta revisão, foram identificas lacunas na avaliação de assistentes virtuais.
2. **Proposição de solução:** visando atender à necessidade do SERPRO de aprimorar a plataforma Serprobots para uma avaliação automatizada de seus *chatbots*, foi proposto o *framework* DUBI como solução de avaliação estática e interativa. Essa solução permite a coleta de métricas por meio da análise da estrutura dos assistentes virtuais e de simulações de conversas.
3. **Implementação da solução e validação experimental:** o *framework* DUBI foi plenamente implementado e sua eficácia foi comprovada por meio de uma validação experimental. No experimento, a solução foi aplicada em *chatbots* reais, disponíveis na plataforma Serprobots, permitindo que todos os aspectos do DUBI pudessem ser observados. Através deste procedimento, foi possível extrair e analisar as métricas propostas pelo *framework*, proporcionando uma discussão dos resultados alcançados.

1.6 Estrutura do documento

Visando facilitar a compreensão dos aspectos abordados nesta dissertação de mestrado, o presente documento foi assim estruturado. O Capítulo 2, intitulado "Fundamentação Teórica", apresenta os conceitos e teorias necessários para o entendimento da problemática abordada nesta pesquisa, bem como da solução proposta. No Capítulo 3, são discutidos temas relevantes relacionados aos *chatbots*, como definição, histórico, evolução, aplicações e desafios. Além disso, é apresentada a revisão da literatura realizada sobre o assunto. O Capítulo 4 é destinado à apresentação da solução desenvolvida neste mestrado, denominada "*Framework* DUBI". Neste capítulo, são detalhados os aspectos relevantes da arquitetura, componentes e funcionamento dessa solução. O Capítulo 5, por sua vez, descreve

o experimento realizado para validar a viabilidade técnica da proposta. Os resultados obtidos, bem como as respectivas análises, são apresentados no Capítulo 6. Por fim, no Capítulo 7, são expostas as conclusões relacionadas a este trabalho, além de indicações de pesquisas futuras que podem ampliar o conhecimento gerado nesta dissertação.

Além do corpo principal do texto, este documento apresenta quatro apêndices com o propósito de complementar as informações contidas nos capítulos. Os Apêndices A e B apresentam, respectivamente, os modelos dos relatórios gerados pelos módulos de avaliação estática e interativa do DUBI. O Apêndice C detalha o experimento conduzido para avaliar técnicas de similaridade de texto; enquanto o Apêndice D apresenta os relatórios de avaliação gerados durante a validação experimental do *framework* DUBI.

Capítulo 2

Fundamentação teórica

O presente capítulo apresenta a fundamentação teórica necessária ao entendimento da problemática abordada e da proposta de solução apresentadas neste documento, abordando conceitos relacionados ao desenvolvimento e à avaliação de *chatbots*. Serão explorados tópicos como processamento e compreensão da linguagem natural, identificação de intenções, recuperação de informação, inteligência artificial generativa, métricas específicas para avaliar a qualidade dos *chatbots*, dentre outros. Também será abordado o funcionamento de um *chatbot* na plataforma Serprobots, e quais os componentes ela disponibiliza. É importante ressaltar que essa fundamentação teórica é importante para o entendimento do trabalho apresentado nesta dissertação de mestrado, pois permitirá ao leitor melhor compreender como a eficiência de um assistente virtual pode ser avaliada.

2.1 Fundamentos de IA para *chatbots*

Os *chatbots*, ou assistentes virtuais, são sistemas projetados para interagir com os usuários de forma automatizada e conversacional, simulando uma conversa humana. Utilizando inteligência artificial, eles conseguem fornecer respostas relevantes e solucionar problemas expressos por seus usuários [1]. Atualmente, são aplicados em diversas áreas, como atendimento ao cliente, suporte técnico, vendas, assistentes pessoais, educação, entre outros.

2.1.1 Classes de *chatbots*

O Capítulo 3 deste trabalho detalhará conceito, histórico e taxonomia dos *chatbots*. Mas, em suma, eles podem ser classificados conforme:

- **o domínio em que atuam [22]:** aberto ou fechado. Os *chatbots* de domínio aberto conseguem responder a uma ampla variedade de perguntas e consultas, independen-

temente do tema. Enquanto os de domínio fechado são especializados e fornecem respostas sobre um determinado contexto.

- **a forma de geração de resposta [22, 24]:** recuperação de informação ou generativo. Os assistentes virtuais baseados em recuperação de informação respondem às perguntas recuperando respostas de bases de conhecimento pré-definidas [1]. Já os generativos conseguem gerar as respostas de forma autônoma, no contexto no qual foi treinado [24].
- **a maneira pela qual compreendem as necessidades dos usuários [2]:** regras ou intenções. Os *chatbots* mais simples são aqueles baseados em regras [25]. De maneira simplificada, estes sistemas utilizam um conjunto de instruções pré-definidas para identificar a necessidade do usuário. Já os *chatbots* baseados em intenções são mais sofisticados, e usam técnicas de Processamento de Linguagem Natural (PLN) para analisar semanticamente as mensagens e extrair delas a necessidade do usuário [26].

2.1.2 Processamento de linguagem natural

O Processamento de Linguagem Natural (PLN) é uma subárea da Inteligência Artificial (IA) e da linguística computacional que se concentra em tornar a linguagem humana mais acessível aos computadores [29]. E, de acordo com Chowdhary (2020) [30], trata-se de um conjunto de técnicas computacionais para análise e representação automática de linguagens humanas, visando compreender, interpretar e produzir texto em linguagem natural. Atualmente, muitas são as aplicações possíveis, como indexação e pesquisa de textos grandes, recuperação de informações, classificação de textos em categorias, extração de informações, tradução automática de idiomas, sumarização automática de textos, sistemas de perguntas e respostas, aquisição de conhecimento, geração de textos e diálogos.

2.1.3 Compreensão da linguagem natural

A compreensão da linguagem natural, do inglês *Natural Language Understanding* (NLU), é um importante subcampo do PLN que visa transformar a linguagem natural em uma representação estruturada e semanticamente significativa, que possa ser processada e compreendida por computadores [31]. A NLU possui diversas tarefas que permitem a identificação de palavras-chave, reconhecimento de contexto de conversa, entendimento de intenções e emoções dos usuários, além da extração de informações importantes de tex-

tos. Isso é possível graças a aplicação de técnicas de PLN, como as análises morfológica, sintática e semântica.

No desenvolvimento de um *chatbot*, é fundamental a utilização de tarefas e recursos de NLU, tendo em vista que facilitam a implementação destes sistemas, possibilitando extrair o contexto e o significado dos textos dos usuários em linguagem natural, bem como indicando como responder conforme a necessidade do usuário. Isto é feito a partir da detecção da intenção do usuário e da extração de entidades específicas do domínio da conversa [32]. Especificamente, uma intenção do usuário representa um mapeamento entre a expressão verbal dele e a ação correspondente que o *chatbot* deve realizar.

2.1.4 Recuperação de informação

Recuperação de informação (RI) é um campo da Ciência da Informação e Biblioteconomia que trata da busca e recuperação de informações relevantes a partir de um conjunto de documentos, como livros, artigos, páginas da web, entre outros. O objetivo da RI é fornecer ao usuário informações precisas e relevantes que atendam às suas necessidades de informação, mesmo que ele não possua conhecimento sobre a localização exata dos documentos que contêm tais informações [33]. A aplicabilidade da RI é vasta, podendo ser utilizada em motores de busca, bibliotecas digitais, sistemas de gerenciamento de documentos, sistemas conversacionais (*e.g. chatbots*), entre outras aplicações.

No contexto de *chatbots*, quando um usuário faz uma pergunta, o sistema identifica a intenção do usuário e extrai as informações relevantes do texto. Em seguida, o *chatbot* usa a recuperação de informação para procurar na base de conhecimento a informação mais relevante para responder à pergunta do usuário.

2.1.5 Inteligência Artificial Generativa

A inteligência artificial generativa é uma subárea da IA que se concentra na criação de sistemas capazes de gerar conteúdo semelhante ao que um ser humano pode criar, tornando o processo de criação de conteúdo mais eficiente e acessível [34]. Esses sistemas são geralmente baseados em redes neurais artificiais, treinadas com grandes conjuntos de dados, e podem ser usados para gerar conteúdos textuais e de multimídia (*e.g.* imagem, áudio e vídeo), com base nos padrões e estruturas que aprenderam com os dados de treinamento.

De acordo com Cao *et al.* (2023) [34], dentre as redes neurais mais utilizadas recentemente, destaca-se a *Transformer*, comumente usada em tarefas de processamento de linguagem natural, como tradução de idiomas e geração de texto. Por exemplo, é neste

tipo de rede em que se baseia o ChatGPT [35], um *chatbot* generativo e de domínio aberto que ganhou destaque mundial no final do ano de 2022.

2.1.6 Motores de conversação baseados PLN e RI

O motor de conversação é responsável por gerenciar o funcionamento de um *chatbot*, de modo que ele possa entender as mensagens dos usuários, interpretar suas necessidades e fornecer respostas adequadas. Para isso, utilizam-se algoritmos e modelos de aprendizado de máquina para processar as entradas do usuário, categorizar suas intenções e gerar as respostas com base em informações pré-definidas [36].

No contexto dos motores de conversação baseados em PLN e RI, esse processamento é possibilitado graças a utilização de elementos que compõem o sistema, como intenções, exemplos de treinamento e nós de diálogo.

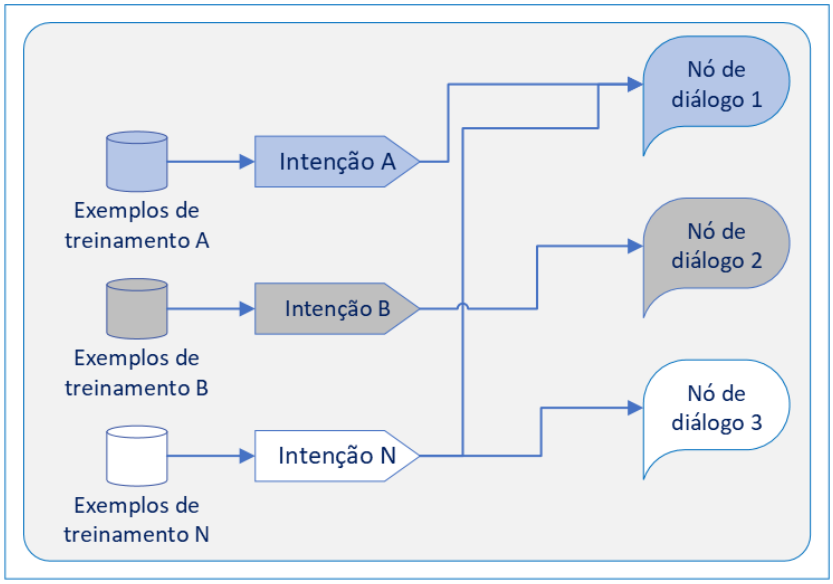


Figura 2.1: Elementos do motor de conversação de *chatbot* PLN e RI.

Fonte: autoria própria.

Uma intenção é o propósito ou objetivo do usuário ao interagir com o *chatbot*, como solicitar informações sobre um produto ou realizar uma transação. Já os exemplos de treinamento são frases fornecidas pelos desenvolvedores que refletem as diversas maneiras pelas quais os usuários podem expressar a mesma intenção. Assim, cada intenção é composta por um conjunto de exemplos de treinamento. Esses textos são usados no processamento das mensagens dos usuários, com foco em ensinar o assistente virtual a reconhecer aquela intenção. Já os nós de diálogo definem a lógica do fluxo da conversa, especificando como o *chatbot* deve responder a cada intenção detectada. Para isso, os nós de diálogo possuem condições de ativação, nas quais podem ser referenciadas as intenções.

Com base nessas informações, a Figura 2.1 representa a arquitetura dos elementos de um motor de conversação e suas inter-relações. A partir da imagem, é possível perceber que cada intenção possui seu conjunto de exemplos de treinamento e pode estar associada a um ou mais nós de diálogo. Por exemplo, para o "Nó de diálogo 1" ser ativado depende de alguma forma das intenções "A" e "N". Enquanto o "Nó de diálogo 2" seria ativado apenas com a detecção da "Intenção B", e o "Nó de diálogo 3" dependeria exclusivamente da "Intenção N".

Para melhor ilustrar estes conceitos, considere que o *chatbot* representado pela Figura 2.1 foi treinado para responder a questionamentos sobre serviços de trânsito. Assim, exemplificando, a "Intenção B", e seus respectivos componentes poderiam ser representados por:

- **Intenção:** "Obter CNH".
- **Exemplos de treinamento:**
  - "Como faço para tirar minha carteira de motorista?"
  - "Qual é o processo para obter a CNH?"
  - "Quais são os requisitos para tirar a carteira de habilitação?"
  - "Preciso de informações sobre a obtenção da CNH."
- **Nó de diálogo:**
  - **Condição de ativação:** "Se a intenção *'Obter CNH'* for detectada."
  - **Texto de resposta:** "Para obter a CNH, siga estes passos: 1. Inscreva-se em uma autoescola autorizada. 2. Complete aulas teóricas sobre legislação e direção defensiva. 3. Faça aulas práticas de direção. 4. Passe no exame prático de direção e obtenha a Permissão Para Dirigir (PPD). 5. Cumpra o período de PPD sem infrações. 6. Agende e seja aprovado no exame final para receber a CNH definitiva."

Neste contexto, atualmente existe uma diversidade de ferramentas para construção de *chatbot* que se fundamentam nos conceitos de intenções, exemplos de treinamento e nós de diálogo para criar estes sistemas. Cada uma delas utiliza uma nomenclatura diferente para esses elementos, mas sua essência permanece a mesma. Por exemplo, o Watson Assistant, da IBM, emprega *"intents"*, *"user expressions"* e *"dialog nodes"* [14]; o Dialogflow, da Google, utiliza *"intents"*, *"training phrases"* e *"fulfillments"* [15]; o Amazon Lex, da Amazon, também se referencia às intenções como *"intents"*, mas nomeia os exemplos de treinamento e os nós de diálogo, respectivamente, como *"sample utterances"* e *"slots"* [16];

e o Rasa NLU utiliza *"intents"*, *"training data"* e *"stories"* para indicar os três elementos [17].

2.2 Métricas para avaliar qualidade de *chatbots*

A avaliação de um *chatbot* visa medir sua efetividade em atender às necessidades dos usuários e funcionar corretamente. Embora seja um tema de grande interesse para pesquisadores, ainda não há um padrão amplamente aceito sobre como avaliar assistentes virtuais, conforme será apresentado no Capítulo 3 deste documento. De forma prática, isso resulta em uma falta de categorização para as métricas de avaliação de qualidade.

Autores como Casas *et al.* (2020) [37] confirmam a falta de padronização e sugerem ser comum cada projeto definir uma categorização quanto às métricas utilizadas. Na revisão da literatura que fizeram, eles identificaram diversas categorias de métodos de avaliação, como desempenho, satisfação do usuário, recuperação de informações, funcionalidades, coerência e perspectivas linguísticas, entre outras.

Do modo similar, Maroengsit *et al.* (2019) [22] também afirmam não haver um padrão de avaliação amplamente aceito, mas sugerem um conjunto menor de categorias de avaliação, incluindo conteúdo, satisfação do usuário e aspectos funcionais.

No entanto, essas categorias ainda não são suficientes para representar e agrupar adequadamente os tipos de avaliação e métricas usadas na avaliação de *chatbots*. Portanto, neste documento, será utilizada uma categorização própria de métricas, derivada da análise de vários artigos sobre o tema, incluindo os estudos de Casas *et al.* [37] e Maroengsit *et al.* [22]. Essa categorização inclui métricas de desempenho, qualidade das respostas, qualidade do diálogo e qualidade do *design*. Espera-se que essa categorização, detalhada nas subseções a seguir, represente melhor os objetivos de cada métrica na avaliação de *chatbot*.

2.2.1 Métricas de desempenho

No contexto de *chatbots*, métricas de desempenho são medidas utilizadas para avaliar a qualidade do *chatbot* durante as interações com os usuários. Essas métricas permitem que os desenvolvedores avaliem a eficácia e a assertividade do sistema, possibilitando identificar pontos que precisam de melhorias. Dentre as métricas incluídas nesta categoria, destacam-se as que são extraídas de uma matriz de confusão: acurácia, precisão, *recall* e *F1-score*.

Matriz de confusão

A matriz de confusão é uma tabela que mostra a frequência de classificação correta e incorreta de um modelo de aprendizado de máquina em relação a um conjunto de dados de teste. A matriz é construída a partir de uma comparação entre as previsões do modelo e as classes verdadeiras dos exemplos de teste [38].

Conforme apresentado na Figura 2.2, a matriz é organizada em quatro quadrantes, que representam as previsões corretas (verdadeiros positivos e verdadeiros negativos) e as previsões incorretas (falsos positivos e falsos negativos) do modelo em relação aos valores reais. Esses quadrantes são rotulados como "Verdadeiros positivos" (VP), "Falsos positivos" (FP), "Falsos negativos" (FN) e "Verdadeiros negativos" (VN).

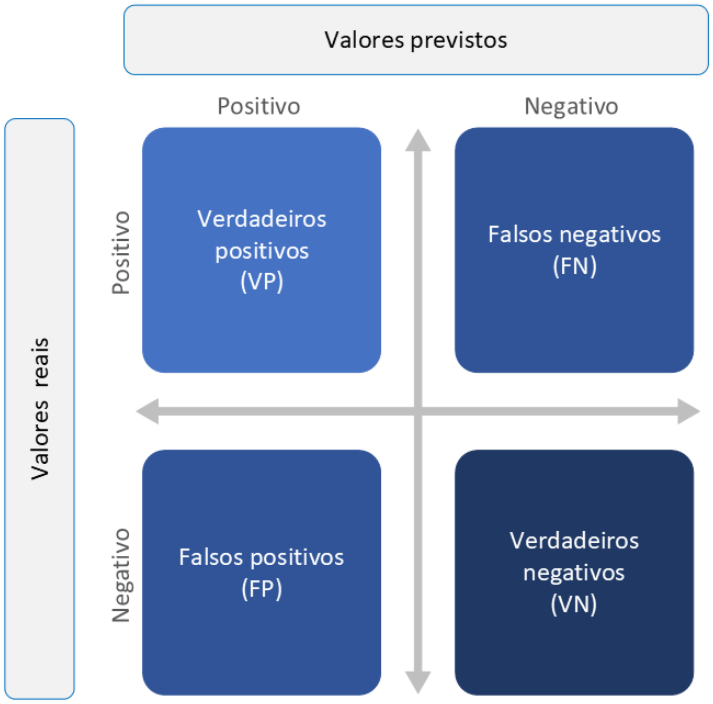


Figura 2.2: Matriz de confusão.

Fonte: autoria própria.

No domínio dos *chatbots*, a terminologia de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos diz respeito à capacidade do modelo do *chatbot* discernir com precisão o objetivo do usuário e fornecer uma reação adequada. Esses conceitos são amplamente empregados na avaliação do desempenho de modelos de processamento de linguagem natural. Neste contexto, Mohammad *et al.* (2020) [9] sugerem:

- **Verdadeiros positivos:** ocorrem quando o modelo do *chatbot* identifica corretamente a intenção do usuário como parte de seu escopo de treinamento e fornece uma

resposta adequada. Em outras palavras, o *chatbot* acerta ao responder corretamente a uma pergunta ou solicitação do usuário que está dentro de sua capacidade;

- **Verdadeiros negativos:** são os casos em que o *chatbot* identifica corretamente que a entrada do usuário não corresponde ao seu escopo de treinamento e, portanto, responde com a mensagem de *fallback*. Isso ocorre quando o *chatbot* reconhece que não pode responder adequadamente ao que foi perguntado;
- **Falsos positivos:** acontece quando o *chatbot* erroneamente classifica uma pergunta como pertencente a uma intenção específica quando, na verdade, ela pertence a outra intenção ou está fora de seu escopo. Ou seja, o *chatbot* responde como se tivesse reconhecido corretamente a intenção, mas na realidade, ele fornece uma resposta errada; e
- **Falsos negativos:** ocorrem quando o *chatbot* não consegue reconhecer uma intenção que deveria ter sido identificada. Ou seja, o *chatbot* falha em identificar corretamente uma pergunta ou solicitação do usuário que está dentro de sua capacidade, e em vez disso responde com a mensagem de *fallback*.

De acordo com estas informações, as métricas comumente utilizadas para avaliar o desempenho do modelo do *chatbot* são:

- **Acurácia:** utilizada para indicar a taxa de acerto geral, esta métrica é calculada através da equação,

$$Acurácia = \frac{VP + VN}{VP + FP + FN + VN}, \tag{2.1}$$

  a qual corresponde à proporção de respostas corretas em relação ao total de respostas dadas.

- **Precisão:** útil para medir a capacidade do modelo de evitar falsos positivos, a precisão é dada por

$$Precisão = \frac{VP}{VP + FP}, \tag{2.2}$$

  ou seja, expressa a proporção de respostas corretas (verdadeiros positivos) em relação ao número total de respostas classificadas como positivas (verdadeiros positivos e falsos positivos).

- ***Recall*:** é a taxa de verdadeiros positivos em relação ao total de respostas classificadas como positivas. Sendo calculada pela equação,

$$Recall = \frac{VP}{VP + FN}, \tag{2.3}$$

o *recall* mede a capacidade do modelo de identificar todas as instâncias positivas.

- ***F1-score*:** é a média harmônica entre a precisão e o *recall*, representando uma medida geral de desempenho de um modelo. Calculado por:

$$F1 = \frac{2 * (Precisão * Recall)}{Precisão + Recall}. \tag{2.4}$$

Assim, a utilização do *F1-score* é apropriada quando se deseja equilibrar a importância das métricas precisão e *recall*.

2.2.2 Métricas de satisfação do usuário

As métricas de satisfação do usuário são amplamente utilizadas para avaliar *chatbots* devido à complexidade inerente desses sistemas, que muitas vezes não possuem uma única resposta correta para as perguntas dos usuários [22]. Essas métricas visam mensurar a qualidade da experiência dos usuários ao interagirem com um *chatbot*, permitindo avaliar tanto respostas individuais quanto toda a conversa.

Nessa avaliação, é comum o uso da escala *Likert* [39], na qual os usuários podem expressar sua percepção utilizando uma variedade de notas que refletem seu nível de concordância ou discordância em relação às respostas recebidas. A escala pode ser ajustada conforme a necessidade de uso, podendo ser expressa por intervalos de números (*e.g.* de 1 a 5) ou de textos (*e.g.* discordo totalmente, discordo, indiferente, concordo, concordo totalmente).

Neste processo de avaliação, baseado nas opiniões dos usuários, é comum que eles também classifiquem as respostas recebidas considerando aspectos específicos e de interesse da equipe responsável pelo *chatbot*, como a adequação ao tema e a naturalidade da escrita [22]. Além disso, autores, como mencionado em [21], também podem utilizar a estratégia de avaliar a satisfação do usuário para qualificar o *chatbot* em relação à especificidade e coerência das respostas.

2.2.3 Métricas de qualidade das respostas

A avaliação de aspectos linguísticos dos textos fornecidos pelos *chatbots* compõe esta classe de métricas. De acordo com Casas *et al.* (2020) [37], existem quatro aspectos a serem observados em um texto: qualidade, quantidade, relação e modo. Qualidade refere-se à precisão, relevância e coerência geral das respostas do *chatbot*. Quantidade diz respeito ao montante de informações fornecidas na resposta a um usuário. Relação trata da relevância das respostas. Por fim, modo refere-se ao tom, estilo ou cortesia das respostas do *chatbot*.

Como pode ser observado, estes aspectos linguísticos tratam de conceitos relativamente subjetivos, o que se apresenta como desafio à automatização da sua avaliação. Por este motivo, diversos autores utilizam um conjunto de métricas na tentativa de obter a automação de alguns destes aspectos linguísticos. É o que apontam Finch e Choi (2020) [40], no trabalho que discutiu as limitações do uso de métricas automatizadas para avaliar sistemas conversacionais. Estes autores listaram um conjunto de métricas e as ordenaram segundo a quantidade de trabalhos que as utilizavam. Destas, destacam-se:

- **BLEU:** acrônimo do termo em inglês *Bilingual Evaluation Understudy*, a métrica BLEU foi originalmente desenvolvida para avaliar a qualidade de traduções automáticas, mas também vem sendo utilizada na avaliação de *chatbots* generativos. Ela compara a resposta gerada pelo assistente virtual com um conjunto de respostas de referência consideradas corretas ou adequadas. Esta comparação se dá através do cálculo de sobreposição de palavras, e gera um valor de precisão que varia em 0 (zero) e 1 (um). Quanto maior o valor obtido, maior é a sobreposição de palavras e, potencialmente, a qualidade da resposta [22, 8].
- ***Distinct-n*:** esta métrica avalia a diversidade lexical das respostas geradas por um *chatbot*. Ela mede quantos n-gramas únicos (isto é, palavras ou sequências de palavras adjacentes) são usados nas respostas. O "n" em *"distinct-n"* representa o tamanho do n-grama considerado. Por exemplo, o *distinct-1* mede a diversidade de palavras únicas, enquanto o *distinct-2* mede a diversidade de bigramas únicos. O resultado da métrica é expresso por um valor entre 0% e 100%, sendo que uma pontuação alta indica uma maior variedade lexical da resposta, o que pode ser um indicador de melhor qualidade e criatividade nas respostas [13].
- **Perplexidade:** é uma métrica usada para avaliar o desempenho de um modelo de linguagem. Ela mede o quão bom é o modelo para predizer uma sequência de palavras. Quanto menor a perplexidade, melhor é o modelo de linguagem em prever a próxima palavra em uma sequência, o que o torna mais coerente [41]. No contexto de *chatbots*, esta métrica pode ser utilizada para medir a fluência e a naturalidade das repostas, ou seja, o quão próximas estão de textos criados por humanos [13].
- **Legibilidade:** é uma medida que visa avaliar a facilidade com que um texto é compreendido por seus leitores. Para isso, essa métrica considera diversos fatores relacionados à estrutura, à complexidade e ao estilo do texto, para fornecer uma indicação sobre a sua acessibilidade e compreensibilidade a um público-alvo específico. Existem diversos índices de legibilidade utilizados para avaliar textos, dentre eles o *Flesch–Kincaid grade level* (FK) [42] e o *Gulpease Index* ($G - index$) [43]. Esses índices são calculados com base em fórmulas específicas que consideram diferentes

aspectos linguísticos do texto. Por exemplo, enquanto o *Flesch–Kincaid grade level* considera as relações entre quantidades de palavras (P) por frases (F) e sílabas (S) por palavras, o *Gulpease Index* observa as proporções de frases por palavras e letras (L) por palavras. Neste contexto, o trabalho de Moreno *et al.* (2022) [44] propõe uma adaptação das equações matemáticas destes índices para o idioma português, resultando respectivamente em:

$$FK = 227 - 1,04 \times \frac{\sum_{i=1}^{a} P_i}{\sum_{j=1}^{b} F_j} - 72 \times \frac{\sum_{k=1}^{c} S_j}{\sum_{i=1}^{a} P_i} \tag{2.5}$$

e

$$G\text{-}index = 89 + \frac{300 \times (\sum_{j=1}^{b} F_j) - 10 \times (\sum_{m=1}^{d} L_m)}{\sum_{i=1}^{a} P_i}. \tag{2.6}$$

Nestas equações, $\sum_{i=1}^{a} P_i$ representa a quantidade de palavras do texto, $\sum_{j=1}^{b} F_j$ indica o número de frases, $\sum_{k=1}^{c} S_j$ quantifica o total de sílabas daquele texto e $\sum_{m=1}^{d} L_m$ é a quantidade de letras.

2.2.4 Métricas de qualidade do diálogo

As métricas desta categoria se referem à avaliação da eficácia do diálogo estabelecido entre um usuário e um *chatbot*. Diferente das métricas de qualidade das respostas, apresentadas na Seção 2.2.3, que focam nas características linguísticas das respostas individuais, essa categoria visa medir a qualidade geral do diálogo. Para isso, são observados aspectos relacionados a adequabilidade das respostas no fluxo da conversa, observando-se, por exemplo, métricas como o tempo de resposta, a consistência e a relevância das respostas, bem como o quanto o *chatbot* consegue lidar com perguntas fora do seu escopo.

Essa abordagem é evidenciada em trabalhos recentes, como os de Mohammad *et al.* (2020) [9] e Yang *et al.* (2022) [13]. Mohammad *et al.* utilizam o tempo de resposta e as taxas de *fallback* e compreensão para analisar a qualidade do diálogo, enquanto Yang *et al.* [13] recorrem à consistência e à relevância das respostas.

Assim sendo, podem ser destacados os seguintes conceitos:

- **Tempo médio de resposta (tm_R):** esta métrica indica a velocidade com a qual o *chatbot* processa e gera uma resposta após receber a pergunta do usuário, sendo definida por

$$tm_R = \frac{\sum_{i=1}^{n} t_{R_i}}{n}, \tag{2.7}$$

onde t_R representa o tempo de resposta da interação i, podendo o i variar de 1 até n, o qual é o número total de interações realizadas. Assim, o tempo médio de resposta

(tm_R) é definido como a razão do somatório dos tempos de todas as respostas pela quantidade de interações realizadas.

- **Taxa de *fallback* (T_f):** quando um *chatbot* recebe uma consulta para a qual não possui resposta, ele pode recorrer a uma estratégia de *fallback*, que consiste em fornecer uma resposta genérica, solicitar mais informações ou redirecionar o usuário para outro canal de atendimento. Assim sendo, esta métrica é utilizada para medir a capacidade de um *chatbot* em lidar com as solicitações que estão além do seu escopo pré-definido, sendo calculada como

$$T_f = \frac{\sum_{k=1}^{n} i_{f_k}}{n}, \tag{2.8}$$

ou seja, a proporção das interações onde o *chatbot* precisou utilizar uma estratégia de *fallback* (i_f) em relação ao número total de interações (n). Uma alta taxa de *fallback* indica que o *chatbot* não consegue compreender efetivamente uma parte significativa das solicitações dos usuários, apontando assim para a necessidade de retreinamento [9].

- **Taxa de compreensão (T_c):** empregada para avaliar a capacidade do *chatbot* de entender imprecisões presentes nas mensagens do usuário, a taxa de compreensão (T_c) indica o quão bom é o assistente virtual em entender a necessidade do usuário quando erros ortográficos e/ou gramaticais estão presentes no texto [9]. Esta métrica é expressa de acordo com

$$T_c = \frac{R_c}{M_e}, \tag{2.9}$$

onde M_e significa a quantidade de mensagens com erros ortográficos e/ou gramaticais recebidas, e R_c é a quantidade de respostas corretas dadas a tais solicitações. Quanto mais alta essa taxa, maior é a capacidade do *chatbot* em entender a necessidade do usuário, mesmo com erros de escrita; em contrapartida, valores baixos indicam a necessidade de preparar melhor o *chatbot* para tais situações.

- **Taxa de consistência das respostas (T_{CR}):** esta métrica visa identificar se as respostas do *chatbot* permanecem coerentes e alinhadas com o contexto do diálogo estabelecido com o usuário. Ou seja, uma conversa consistente é aquela que fornece as mesmas respostas a perguntas repetidas ou com textos similares, mas que representam a mesma intenção do usuário [13]. Neste cenário, a taxa de consistência das respostas de uma conversa (T_{CR}) é dada pela equação

$$T_{CR} = \frac{R_c}{M_r}, \tag{2.10}$$

na qual M_r representa a quantidade de mensagens repetidas ou com textos similares, e R_c é a quantidade de respostas corretas fornecidas a estas mensagens.

- **Relevância das respostas:** a relevância se refere à pertinência das respostas fornecidas pelo *chatbot* em relação ao contexto estabelecido pelo usuário [40]. Ou seja, está relacionada à capacidade de compreender adequadamente a intenção do usuário e fornecer uma resposta que seja útil, precisa e satisfatória. Uma característica inerente a esta métrica é a subjetividade no processo de avaliação, sendo, por isso, comumente avaliada mediante julgamentos humanos, como nos trabalhos [45, 46]. Já em [13], a relevância de uma resposta é avaliada através da identificação e contabilização das respostas que apresentam conceitos previamente mencionados na conversa.

2.2.5 Métricas de qualidade do *design*

As métricas de qualidade do *design* de um *chatbot* são indicadores que avaliam aspectos relacionados à modelagem desse tipo de sistema. Ao analisar tais métricas, são considerados os elementos estruturais do assistente virtual, como os fluxos conversacionais, as intenções e os textos utilizados como exemplos de treinamento ou respostas [11, 12].

Essas métricas fornecem uma visão objetiva sobre a eficiência da modelagem do *chatbot*, permitindo a identificação antecipada de potenciais problemas no uso do assistente e, consequentemente, contribuindo para aprimorar a experiência do usuário final [12].

Dada a complexidade destes sistemas, vários aspectos podem ser observados em um projeto de modelagem de *chatbot*, proporcionando assim uma abundância de métricas passíveis de serem utilizadas para avaliar a qualidade do seu *design*. Os trabalhos [11] e [12] apresentam diversas dessas métricas, agrupando-as segundo os elementos da estrutura a serem avaliados, como intenções, fluxos conversacionais e respostas. Dentre essas, algumas métricas são detalhadas na presente seção, tendo em vista que serão utilizadas neste trabalho, conforme será apresentado no Capítulo 4.

Métricas relacionadas às intenções de um *chatbot*

- **Quantidade de exemplos de treinamento (QET):** esta métrica é responsável por contabilizar o número de exemplos de treinamento de cada intenção. Eles são utilizados pelo modelo de aprendizado de máquina do *chatbot* para reconhecer frases semelhantes dos usuários e relacioná-las à intenção correta. Quanto maior for esta quantidade, mais preciso é o reconhecimento da intenção [12]. Entretanto, valores excessivamente altos ou baixos não são recomendados. Sendo comum o estabelecimento de limites, inferior e superior, pelas plataformas de assistentes virtuais

disponíveis no mercado [14, 15]. Assim, a QET de uma intenção é obtida através do somatório de todos os seus exemplos de treinamento, conforme:

$$QET_I = \sum_{i=1}^{n} E_{T_i}, \tag{2.11}$$

onde E_T se refere a um exemplo de treinamento da intenção I, e n o número total de exemplos de treinamento da intenção.

- **Similaridade entre intenções distintas (SID):** visando identificar similaridade entre diferentes intenções, o que pode gerar confusão ao modelo do *chatbot*, esta métrica calcula o percentual de intenções que possuem exemplos de treinamento semanticamente semelhantes entre si [12]. Neste contexto, a similaridade entre intenções é calculada a partir dos textos dos seus exemplos de treinamento. Para tal, pode-se utilizar a abordagem de vetorização e distância de cosseno. Nesta abordagem, primeiro se transforma os textos em vetores numéricos e, em seguida, é calculada a distância de cosseno para medir o ângulo entre esses vetores. Quanto menor o ângulo, maior a similaridade, indicando que os textos têm palavras semelhantes e uma estrutura de linguagem parecida.

- **Tamanho dos exemplos de treinamento (TET):** calcula a quantidade de palavras presentes em cada sentença dos exemplos de treinamento. O exemplo de treinamento é utilizado para representar uma possível necessidade do usuário do *chatbot*, sendo necessário que estas frases não sejam muito longas, o que normalmente não é adequado em contextos de *chatbots* [12], tampouco curtas demais a ponto de inviabilizar a detecção daquela intenção [11]. Desta forma, para cada exemplo de treinamento de uma intenção, o TET é calculado pela equação

$$TET = \sum_{i=1}^{n} P_i, \tag{2.12}$$

na qual, n é o número total de palavras do exemplo de treinamento, e P representa uma palavra daquele texto.

- **Legibilidade dos exemplos de treinamento (LET):** similar ao que foi apresentado na Seção 2.2.3, a legibilidade dos exemplos de treinamento avalia a facilidade de entendimento de um texto por seus leitores. Como o exemplo de treinamento é o que o desenvolvedor espera que os usuários usem para invocar uma das intenções do *chatbot*, faz-se necessário que estes textos estejam alinhados à expectativa do público-alvo [11].

- **Representatividade dos exemplos de treinamento (RET):** estudos como [11] sugerem que a utilização de vocabulário comum e representativo, em relação ao idioma utilizado pelos usuários, tendem a melhorar a assertividade do *chatbot* na tarefa de identificar intenções. Com isso, esta métrica calcula o percentual de representatividade dos exemplos de treinamento em relação ao idioma utilizado. Ou seja, evidencia a taxa de termos empregados nas amostras de treinamento representativos no respectivo idioma, sendo calculado por

$$RET = \frac{\sum_{i=1}^{m} P_{R_i}}{\sum_{j=1}^{n} P_j}. \tag{2.13}$$

  Nesta equação, m representa o total de palavras representativas do idioma usadas no texto do exemplo de treinamento, P_R é uma palavra representativa daquele exemplo de treinamento, n é o número total de palavras do texto e P qualquer palavra utilizada pelo mesmo exemplo de treinamento.

Métricas relacionadas às respostas de um *chatbot*

Em oposição às métricas apresentadas na Seção 2.2.3, que visam a avaliação dos aspectos linguísticos dos textos das respostas, as métricas aqui apresentadas focam na avaliação dos textos sob a óptica da modelagem do fluxo de conversa.

- **Tamanho dos textos de resposta (TTR):** esta métrica calcula o tamanho dos textos de cada resposta do *chatbot*, ou seja, a quantidade de palavras utilizadas. Isto é importante porque textos longos são mais difíceis de entender e nem sempre adequados à interface de uso do *chatbots* [12]. Além disso, por se tratar de uma interface conversacional, os desenvolvedores tendem a fornecer mais detalhes do que necessário, quando, na verdade, deveriam focar em serem concisos [47]. Assim, o TTR é dado pela equação

$$TTR = \sum_{i=1}^{n} P_i, \tag{2.14}$$

  onde n é o total de palavras do texto de resposta e P_i representa uma palavra utilizada nesse texto.

- **Representatividade dos textos de resposta (RTR):** similar à métrica RET, entretanto direcionada aos textos de resposta do *chatbot* [11]. Desta forma, a RTR é calculado por:

$$RTR = \frac{\sum_{i=1}^{m} P_{R_i}}{\sum_{j=1}^{n} P_j}, \tag{2.15}$$

na qual, m representa o total de palavras representativas do idioma usadas no texto da resposta, P_{R_i} é a i-ésima palavra representativa daquela resposta, n é o número total de palavras do texto e P_j qualquer palavra utilizada pela mesma resposta.

- **Sentimento negativo dos textos das respostas (STR):** esta métrica afere o percentual das respostas que apresentam textos com sentimento negativo. Essa avaliação está intimamente ligada à satisfação do usuário, pois um *chatbot* que emite predominantemente frases negativas pode ter um efeito prejudicial na experiência do usuário [12]. Neste contexto, o STR de um *chatbot* é obtido via:

$$STR = \frac{\sum_{i=1}^{m} R_{N_i}}{\sum_{j=1}^{n} R_j}, \tag{2.16}$$

onde m indica o número de respostas com sentimento negativo, R_N é uma resposta nesta característica, n é o total de respostas cadastradas no *chatbot* e R é uma resposta qualquer do assistente virtual.

2.3 *Chatbots* na plataforma Serprobots

A proposta deste trabalho, apresentada no Capítulo 4, fundamenta-se na estrutura e nos princípios de funcionamento dos *chatbots*. Neste sentido, para compreender plenamente o *framework* DUBI, é essencial entender os componentes e o funcionamento desses assistentes virtuais na plataforma Serprobots, uma vez que o DUBI está sendo desenvolvido como uma aplicação e evolução direta dessa plataforma. Entretanto, é importante ressaltar que o funcionamento aqui discutido não se restringe apenas à Serprobots, pois outras soluções de mercado podem adotar conceitos semelhantes.

2.3.1 Componentes dos *chatbots*

Na plataforma Serprobots, os *chatbots* são compostos internamente por um conjunto de componentes organizados em uma ordem específica, formando o que é conhecido como fluxo de conversação. Esse fluxo é responsável por direcionar, sequencialmente, as mensagens dos usuários para cada componente, conforme detalhado na Seção 2.3.2.

Neste cenário, os componentes são elementos que desempenham funções específicas no processo de interação com o usuário, cada um possuindo uma tarefa bem definida e podendo ser combinado com outros para criar o referido fluxo de conversação. São duas as categorias de componentes: motores de conversação e componentes comportamentais.

Os componentes comportamentais são facilitadores que tratam de situações comuns em *chatbots*. Eles são oferecidos pela Serprobots para melhorar a produtividade das equipes,

evitando a necessidade de implementar o tratamento desses comportamentos em cada projeto. Atualmente, a plataforma oferece os seguintes:

- ***Onboarding***: permite definir a mensagem de boas-vindas ou apresentação que será exibida para o usuário no início de cada conversa. É um componente obrigatório, que sempre será o primeiro do fluxo de conversação e executado apenas ao iniciar uma nova conversa.
- **Inatividade**: esse componente permite definir um tempo de espera que o *chatbot* aguardará sem interação do usuário. Após o decorrer deste tempo, caso o usuário não interaja, a conversa é encerrada pelo assistente virtual. Diferente dos demais componentes, este não possui uma ordem de execução, sendo transversal ao fluxo da conversa.
- **Repetição**: de utilização opcional, o componente de repetição é responsável por identificar se o usuário está insistindo em uma mesma pergunta, ou perguntas similares, sequencialmente. Quando detectada esta situação, este componente permite que o *chatbot* envie uma mensagem específica ao usuário, buscando o direcionar ao contexto do fluxo de conversa. A mensagem enviada por este componente é configurada pela equipe desenvolvedora do assistente virtual.
- **Impropriedades**: este componente opcional permite definir uma lista de palavras ou expressões impróprias ao contexto do *chatbot*. Quando a mensagem do usuário passa por este componente, ele verifica a presença dessas palavras ou expressões. Se identificadas, o *chatbot* pode, por exemplo, responder informando que a linguagem é imprópria e incentivar o usuário a reformular a pergunta. A equipe de desenvolvimento do assistente virtual tem liberdade para definir a lista de impropriedades e o texto de resposta enviado pelo componente.
- **Avaliação**: visando obter a percepção do usuário quanto ao atendimento ofertado, o componente de avaliação permite definir o comportamento do *chatbot* ao ser avaliado pelo usuário, configurando as respostas predefinidas para as avaliações positiva e negativa. Além disso, no caso de uma avaliação negativa, este componente solicita ao usuário que informe o motivo da má avaliação. Esta informação fica disponível à equipe do *chatbot* e pode ser uma das fontes de melhoria do assistente.
- **Assunto desconhecido**: responsável por fornecer ao usuário uma mensagem de *fallback*, quando o assistente virtual não compreender a mensagem recebida, este componente é de uso obrigatório e localizado sempre na última posição do fluxo de conversa.

Já os motores de conversação são responsáveis por proporcionar o diálogo efetivo com o usuário, e podem ser implementados conforme as classificações apresentadas na Seção 2.1.1 deste capítulo. Atualmente, a plataforma oferece diferentes opções de motores, incluindo abordagens baseadas em regras, intenções, recuperação de informação e generativas. Neste contexto, são alternativas disponíveis na plataforma Serprobots:

- **RiveScript** [48]: linguagem de programação de regras de diálogo que permite criar interações conversacionais com *chatbots* de maneira textual e estruturada, utilizando padrões de entrada e saída predefinidos. É representante da categorização de motor de conversação baseado em recuperação de informação e regras.
- **IBM Watson *Assistant*** [14]: pertencente à classificação de motores de conversação baseados em recuperação de informação e em intenções, o IBM Watson *Assistant* é um serviço de inteligência artificial que utiliza recursos de processamento de linguagem natural para compreender e responder às perguntas e comandos dos usuários, facilitando a interação e o suporte automatizado.
- **Serprobots Perguntas & Respostas**: também exemplar dos baseados em Recuperação de informação (RI) e em intenções, o Serprobots Perguntas & Respostas é um motor de conversação desenvolvido pelo Serviço Federal de Processamento de Dados, que recorre a técnicas de indexação de texto e busca semântica para detectar a intenção do usuário, dentre os assuntos existentes em uma base de conhecimento, e recuperar uma resposta adequada àquela solicitação.
- ***Generative Pre-Training* (GPT)** [49]: é uma arquitetura de modelo de linguagem desenvolvida pela OpenAI, capaz de gerar texto com base em seu conteúdo pré-treinado ou utilizando uma base de conhecimento enriquecida. Ele representa a categoria dos motores de conversação generativos, podendo ser utilizado como base para a criação de *chatbots*, onde é treinado em contextos específicos para aprender a gerar respostas relevantes e contextuais às interações dos usuários.

Assim, a integração dos componentes comportamentais juntamente com os motores de conversação estabelece a base para o funcionamento do *chatbot*, conforme abordado na próxima seção.

2.3.2 Funcionamento dos *chatbots*

Conforme mencionado anteriormente, o fluxo de conversação de um *chatbot* é criado a partir do agrupamento de componentes, constituindo a base para o funcionamento deste assistente virtual. Esse fluxo representa uma sequência lógica de interações entre o *chatbot*

e o usuário durante uma conversa, direcionando o processamento das mensagens, a geração ou recuperação de respostas e o avanço do diálogo para atender às necessidades do usuário.

Na plataforma Serprobots, a equipe de desenvolvimento do assistente virtual é responsável por definir o fluxo de conversação, tendo autonomia para escolher quais componentes utilizar e em que ordem. Essa flexibilidade permite adaptar o *chatbot* conforme as necessidades específicas do projeto.

Desta forma, quando uma mensagem é recebida pelo assistente virtual, ela passa por uma cadeia de componentes que tentam processá-la. O primeiro componente recebe a mensagem e, se não puder fornecer uma resposta, encaminha-a para o próximo componente da sequência. Esse processo continua até que um componente possa responder, interrompendo assim o processamento do fluxo de conversação e retornando uma resposta ao usuário, conforme exemplificado na Figura 2.3. Nessa figura, os retângulos cinza representam os componentes ativados e que avaliaram se poderiam responder à mensagem; enquanto os retângulos azuis indicam componentes não ativados porque a mensagem foi respondida por algum anterior a eles.

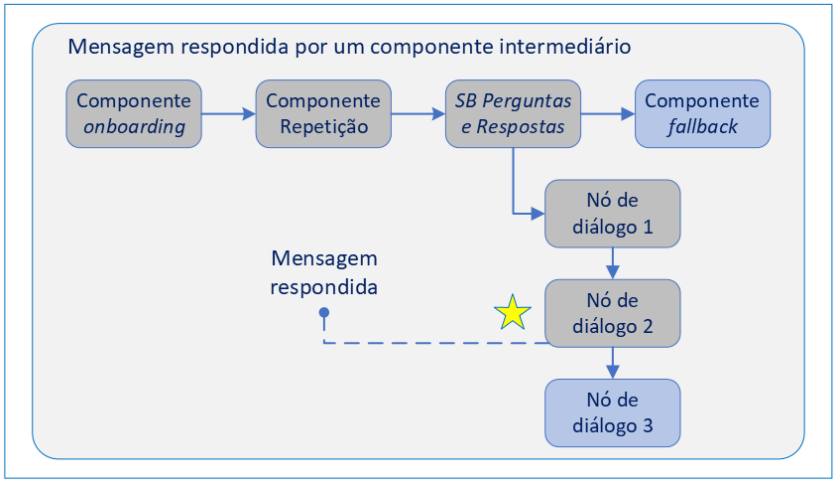


Figura 2.3: Exemplo do processamento de uma mensagem para a qual o *chatbot* possui resposta.

Fonte: autoria própria.

Para melhor ilustrar este funcionamento, imagine que o fluxo representado na Figura 2.3 é de um *chatbot* de suporte a vendas de uma empresa fictícia. Durante uma conversa, o usuário envia a mensagem *"Qual o telefone de contato da equipe de vendas?"*. Segundo o fluxo de conversação configurado, o processamento desta mensagem seria:

1. O *chatbot* recebe a mensagem do usuário e a entrega ao primeiro componente do fluxo, que neste caso é o "*onboarding*";

2. Por não ser a primeira mensagem da conversa, pois ela já havia sido iniciada antes, a mensagem é imediatamente entregue ao próximo da sequência, o componente "repetição";

3. O componente "repetição" analisa o texto da mensagem, comparando-o com o histórico das mensagens anteriores. Neste exemplo, não detecta repetição e, por isso, envia a mensagem ao próximo da lista;

4. O próximo componente do fluxo de conversação é o motor de conversação "Serprobots Perguntas & Respostas", representado na Figura 2.3 pelo item "SB Perguntas e Respostas". Este motor, como dito anteriormente, é do tipo RI e baseado em intenções, sendo um exemplo dos motores de conversação abordados na Seção 2.1.6. Assim, ao receber a mensagem, ele realiza o seguinte processamento:

   (a) **Detecção de Intenção:** o componente utiliza técnicas de PLN para detectar a intenção da mensagem do usuário. Para isso, ele analisa palavras-chave, contextos e padrões para identificar a intenção subjacente, a qual é o propósito ou ação desejada pelo usuário. No exemplo, a intenção "informar telefone de contato" é detectada.

   (b) **Busca pelo nó de diálogo**: com a intenção identificada, o motor de conversação consulta sua base de conhecimento, a qual é uma estrutura que contém intenções e nós de diálogo relacionados, buscando resposta à mensagem recebida. Nesse caso, ele avalia inicialmente o "Nó de diálogo 1" e percebe que a intenção não está associada a este nó. Passa ao "Nó de diálogo 2", identificando que a intenção "informar telefone de contato" é a condição de entrada deste nó.

   (c) **Recuperação da resposta:** com base na intenção detectada e no nó de diálogo encontrado, o motor de conversação recupera a resposta para a pergunta do usuário. Neste caso, por ser um motor de conversação baseado em recuperação de informação, o texto da resposta já estava previamente cadastrado. No exemplo, a resposta poderia ser algo como: *"O telefone de contato da equipe de vendas é +55 61 12345-6789"*.

   (d) **Interrupção do processamento:** tendo em vista que o componente conseguiu responder à mensagem do usuário, ele interrompe seu processamento, não avaliando os nós de diálogo seguintes, e retorna a resposta ao fluxo de conversação.

5. O fluxo de conversação é notificado do sucesso no processamento, o que faz ele enviar a resposta ao usuário e também interromper a sua execução. Desta forma, o componente *"fallback"* não é ativado.

Em situação oposta ao apresentado anteriormente, se o *chatbot* recebe uma mensagem que nenhum componente é capaz responder, ela percorre todo o fluxo de conversação e chega ao final da sequência, ativando o componente de mensagem de *fallback* do *chatbot*. Esta situação é ilustrada pela Figura 2.4.

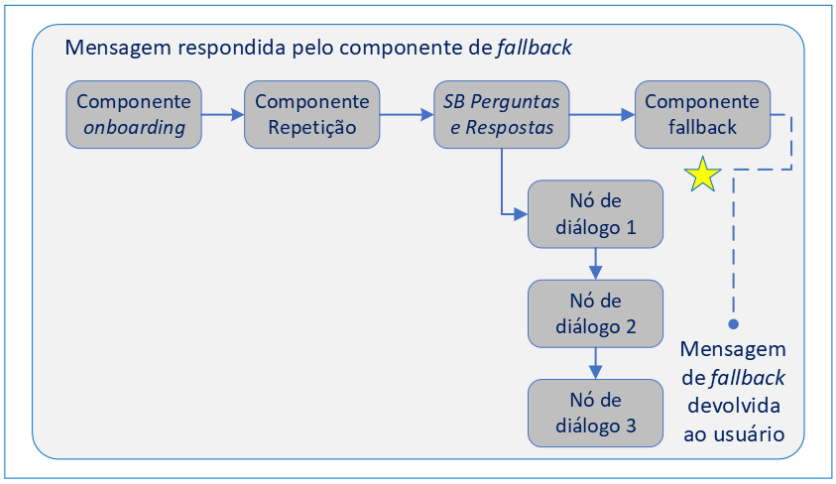


Figura 2.4: Exemplo do processamento de uma mensagem sem resposta pelo *chatbot*.

Fonte: autoria própria.

Em suma, a estruturação em fluxo e componentes permite que os *chatbots* gerenciem e encaminhem eficientemente as mensagens dos usuários, buscando uma interação fluente e respostas adequadas. Essa arquitetura flexível possibilita a personalização dos assistentes virtuais conforme as necessidades específicas de cada projeto.

Adicionalmente, essa estruturação proporciona à plataforma Serprobots acesso completo à organização interna de cada assistente virtual, permitindo a automação de análises sobre este conteúdo, como será discutido no Capítulo 4 deste documento.

2.4 Resumo do capítulo

Neste capítulo, foram apresentados alguns dos fundamentos de Inteligência Artificial (IA) para *chatbots*, visando estabelecer uma base de conhecimento para o entendimento abrangente deste trabalho. Dentre os conceitos apresentados, foram abordadas noções de Processamento de Linguagem Natural (PLN) e *Natural Language Understanding* (NLU), essenciais para a compreensão e interação efetiva dos *chatbots* com os usuários. Além disso, foram examinados os conceitos de recuperação de informação e da IA generativa,

destacando-se sua relevância para o aprimoramento das capacidades dos assistentes virtuais. Na Seção 2.2, foi apresentada uma categorização das métricas utilizadas para avaliar a qualidade de *chatbots*, que será adotada como referência ao longo deste estudo. Diversas métricas foram exploradas em cada categoria, conceituando-as e fornecendo detalhes a respeito de como são calculadas. Por fim, o funcionamento de um *chatbot* na plataforma Serprobots foi explorado, informando o fluxo de execução e quais os componentes podem ser utilizado por eles.

O próximo capítulo apresentará a revisão do estado da arte realizada para este trabalho, a qual recorre aos conceitos e fundamentos discutidos até então.

Capítulo 3

Análise sobre *chatbots*: definição, taxonomia e avaliação

Este capítulo apresenta uma visão geral sobre os *chatbots*, abordando diferentes aspectos da tecnologia, como: sua evolução histórica, taxonomias utilizadas e, principalmente, no que tange ao seu desempenho, assertividade e outros aspectos que tornam o *chatbot* uma opção atraente para atendimento ao usuário, dentre outras aplicações. Para alcançar esse objetivo, foi realizada uma pesquisa bibliográfica que considerou os trabalhos científicos disponíveis nas principais bases de conhecimento, são elas: Portal de Periódicos da CAPES [50], *Web of Science* [51], *IEEE Xplore* [52] e *ACM Digital Library* [53]. Essa pesquisa incluiu trabalhos publicados nos últimos cinco anos, oferecendo uma revisão crítica e atualizada do estado da arte em relação ao tema.

Assim sendo, o capítulo está estruturado da seguinte forma: a Seção 3.1 descreve o conceito de *chatbots* e como podem ser utilizados; na Seção 3.2, o histórico da evolução da tecnologia é apresentado, incluindo marcos importantes no desenvolvimento dos assistentes virtuais; a taxonomia dos *chatbots* é detalhada na Seção 3.3, identificando as principais categorias de *chatbots* e suas características distintas; já a conceituação sobre como pode ser avaliado um *chatbot* é detalhada na Seção 3.4; por fim, na Seção 3.5, as principais metodologias e métricas usadas para avaliar a qualidade dos *chatbots* são discutidas, incluindo as vantagens e limitações de cada uma delas, bem como as lacunas identificadas neste ramo de pesquisa sobre assistentes virtuais.

3.1 Definição de *chatbot*

Os *chatbots* são sistemas de computador projetados para interagir com os usuários por meio de conversas escritas ou faladas. Eles conseguem receber e interpretar mensagens enviadas pelos usuários, processá-las, identificar sua intenção e fornecer uma resposta

adequada e relevante, simulando uma conversa humana [1]. Neste cenário, um *chatbot* de sucesso é aquele onde os usuários considerem a conversa comparável com um diálogo entre humanos [24]. Essa capacidade de interagir com humanos, por meio de linguagem natural, é o que torna os *chatbots* tão úteis em uma variedade de aplicações e contextos de negócio.

Eles foram criados para facilitar a comunicação entre pessoas e instituições, oferecendo um canal de atendimento eficiente e acessível a qualquer hora do dia. Desde então, essa tecnologia tem impactado positivamente a experiência dos usuários, oferecendo a estes tempestividade na obtenção de respostas às suas necessidades. Do ponto de vista das instituições, os *chatbots* também oferecem benefícios, como a redução de custos com atendimento e o aumento da eficiência operacional [26]. Além disso, ao automatizar tarefas repetitivas e padronizadas, os assistentes virtuais permitem que as pessoas se dediquem a atividades mais estratégicas e de maior valor agregado.

De forma geral, os sistemas conversacionais utilizam algoritmos para entender e interpretar as mensagens dos usuários e, em seguida, fornecem respostas personalizadas a partir de uma base de conhecimento [24, 26]. Neste cenário, conforme ilustra a Figura 3.1, a arquitetura básica de um *chatbot* é composta por quatro partes distintas. A primeira é o módulo de entendimento da necessidade do usuário, responsável por processar as mensagens recebidas e identificar a intenção do usuário. O gerenciador de diálogo, segunda parte da arquitetura, orquestra as mensagens recebidas e as respostas geradas, mantendo o contexto da conversa. A base de conhecimento é outro importante componente do *chatbot*, responsável por fornecer os dados utilizados no treinamento do robô e na geração de respostas personalizadas. Por último, o módulo de geração de resposta utiliza o contexto da conversa e a base de conhecimento para gerar respostas que atendam à necessidade do usuário. Assim, conforme os *chatbots* são usados com mais frequência, mais dados são gerados e podem ser utilizados para melhorar o aprendizado dos robôs, permitindo que se adaptem melhor às necessidades do usuário, melhorando suas respostas e interações.

Graças a sua versatilidade, os sistemas conversacionais têm inúmeras aplicações em diferentes setores da economia, tais como varejo, finanças, saúde, governo, educação, entre outros. A depender da aplicação, o objetivo do *chatbot* muda. Por exemplo, no varejo, os *chatbots* podem ser usados para recomendar produtos ou informar sobre estes. Em finanças, eles podem auxiliar investidores com informações detalhadas sobre tipos de investimento. Na saúde, podem ser utilizados no processo de triagem de pacientes em plataformas de tele consulta. Em contexto governamental, podem auxiliar aos cidadãos a encontrar e realizar serviços públicos. Na educação, podem ser utilizados como tutores de aprendizado. Assim sendo, na maior parte das vezes, e independentemente do domínio de negócio, os assistentes virtuais se concentram no auxílio do atendimento ao cliente ou

usuário de um serviço [3].

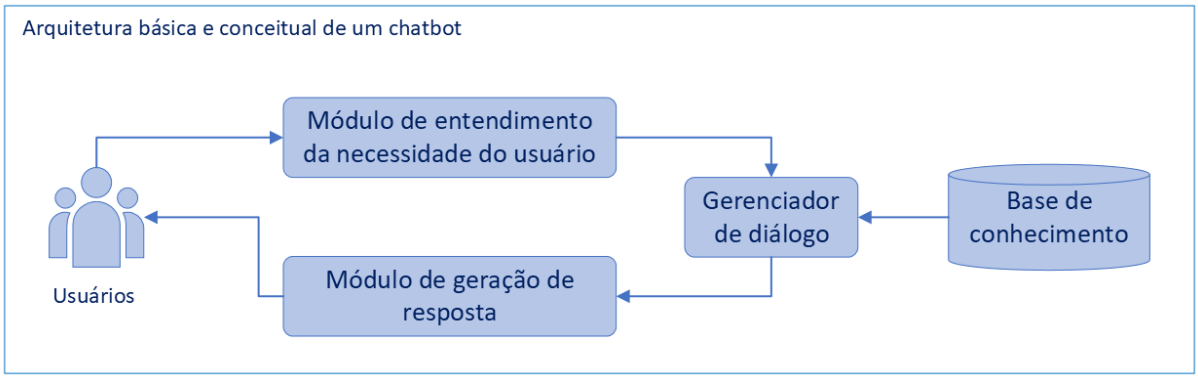


Figura 3.1: Arquitetura básica e conceitual de um *chatbot*.

Fonte: adaptado de [54].

Embora tenham ficado mais conhecidos recentemente, a evolução dos *chatbots* tem sido marcada por uma série de desenvolvimentos tecnológicos ao longo das últimas décadas. Um breve histórico desta evolução é apresentado na Seção 3.2, com destaque para aqueles avanços que foram marcos importantes da tecnologia.

3.2 Histórico da evolução da tecnologia

Apesar dos recentes destaques na impressa, no mercado e na academia, sobre o potencial dos assistentes virtuais, a história dos *chatbots* remonta ao início da inteligência artificial, na década de 1960, quando os primeiros sistemas de diálogo foram criados para simular a conversa humana [55]. Desde então, houve uma série de avanços tecnológicos que permitiram a criação de *chatbots* cada vez mais sofisticados e capazes de lidar com uma variedade de tarefas e situações.

O primeiro sistema de computador historicamente considerado um agente conversacional é o ELIZA [56], criado em 1966 pelo pesquisador Joseph Weizenbaum, do *Massachusetts Institute of Technology* (MIT). O ELIZA [57] foi projetado para simular uma psicoterapeuta e se comunicava com os usuários usando técnicas de processamento de linguagem natural. O sistema utilizava regras simples de substituição de palavras para dar respostas aos usuários, o que fazia parecer que estava entendendo a conversa. Embora limitado em termos de funcionalidade, o ELIZA representou o ponto de partida para as pesquisas sobre sistemas de diálogo automatizados.

Nos anos seguintes, diversos outros *chatbots* foram desenvolvidos e contribuíram com a evolução deste campo de pesquisa. Um deles foi o A.L.I.C.E. (*Artificial Intelligent Internet Computer Entity)*) [58], lançado em 1995. O A.L.I.C.E. representou um significativo

avanço tecnológico no ramo dos assistentes virtuais ao utilizar uma abordagem mais sofisticada de processamento de linguagem natural, com a capacidade de expandir sua base de conhecimento, interpretar as intenções do usuário e responder de forma mais precisa e contextualizada [1]. Isso foi possível devido a sua arquitetura ser baseada em *Artificial Intelligence Mark-up Language* (AIML) [59], uma extensão do *Extensible Markup Language* (XML), dando início à abordagem de *chatbots* estruturados em regras.

Já na década de 2010, com a popularização e viabilidade comercial do uso de redes neurais profundas, houve uma série de avanços significativos na tecnologia dos assistentes virtuais. Um deles foi o desenvolvimento de assistentes pessoais inteligentes por voz — como a Siri da Apple [60], o Google Assistant da Alphabet [61] e a Alexa da AWS [62] — que permitem aos usuários interagirem em linguagem natural e por voz [1]. Posteriormente, destacou-se também o Watson Assistant da IBM [14], que facilitou o desenvolvimento de *chatbots* ao oferecer aos desenvolvedores uma interface simples e que não exigia conhecimento em inteligência artificial para construir os assistentes virtuais. Em comum, estas tecnologias utilizam algoritmos de aprendizado de máquina para analisar semanticamente a mensagem do usuário e detectar as intenções dele, possibilitando prover respostas mais adequadas às necessidades do usuário. Isto sem a rigidez da estrutura de regras utilizadas em anos anteriores.

Mais recentemente, a partir de 2018, os sistemas baseados em perguntas e respostas têm se beneficiado dos significativos avanços nas pesquisas sobre modelos de linguagem, que permitem que eles processem e gerem texto de forma cada vez mais precisa e natural [63]. Um exemplo notável disto é o *Generative Pre-Training* (GPT), um modelo de linguagem desenvolvido pela OpenAI [49], que utiliza a técnica de aprendizado de máquina conhecida como *transformers* para, dentre outras funcionalidades, gerar texto em resposta a perguntas e estímulos [64]. Este tipo de abordagem, apesar de não resolver todos os problemas e ainda apresentar algumas limitações, candidata-se a ser uma das tecnologias mais avançadas em termos de sistemas de diálogo aberto. Não por menos, a indústria e a academia têm investido nesta linha, apresentando diversas outras iniciativas similares ao GPT, como os modelos de linguagem GLaM [65], LaMDA [66] e LLaMA [67].

Em geral, a evolução dos *chatbots* ao longo dos anos tem sido impulsionada por avanços em várias áreas, em especial o processamento de linguagem natural e o aprendizado de máquina. Esta evolução busca melhorias na tecnologia, para preencher lacunas deixadas por iniciativas anteriores. E uma forma de entender melhor esta evolução é por meio de uma taxonomia de *chatbots*, de modo que seja possível classificar estes sistemas de acordo com suas características, conforme é apresentado na Seção 3.3.

3.3 Taxonomias para *chatbots*

A taxonomia de *chatbots* é uma classificação dos diferentes tipos existentes, e pode ser baseada em suas características, tecnologias e funções. Neste contexto, a literatura é abundante em apresentar diferentes taxonomias para os assistentes virtuais [22] [68], no entanto, nem sempre são convergentes. Isto pode ser atribuído aos diferentes critérios utilizados por cada pesquisador no processo de classificação.

Entre as taxonomias propostas na literatura, a classificação de acordo com domínio de conhecimento, distinguindo entre *chatbots* de domínio aberto (também chamados de domínio geral ou *non-task-oriented*) e de domínio fechado (conhecidos também por domínio específico ou *task-oriented*), parece ser uma das mais bem aceitas. Essa taxonomia está presente nos trabalhos de Maroengsit *et al.* (2019) [22], Hussain *et al.* (2019) [68], Tai *et al.* (2019) [69], Almansor e Hussain (2020) [70], Silva e Rodrigues (2021) [23], Brabra *et al.* (2022) [36], Ni *et al.* (2022) [71] e Silva e Barbosa (2022) [72].

Outros autores preferem classificar os assistentes virtuais conforme a sua abordagem de geração de respostas, diferenciando entre *chatbots* baseados em regras (*rule-based*), baseados em recuperação de informação (*retrieval-based*) e baseados em algoritmos generativos (*generative-based*). Ou uma combinação destes três.

Tai *et al.* (2019) [69] e Silva e Rodrigues (2021) [23], referenciam os três termos. Já os trabalhos de Mohamad *et al.* (2021) [24] e Ma *et al.* (2022) [27], utilizam apenas os termos baseados em recuperação de informação e baseados em algoritmos generativos. Por fim, há ainda os que agrupam as abordagens *retrieval-based* e *generative-based* no conceito de "baseados em inteligência artificial" (*ML-based*), é o caso de Caldarini *et al.* (2022) [1] e Motger *et al.* (2023) [2].

Outras classificações também foram encontradas na literatura, considerando o domínio de aplicação do *chatbot*, o canal de interação com o usuário, as funcionalidades ofertadas, dentre outros aspectos. Mas, para o escopo deste trabalho, entende-se serem subdivisões das taxonomias apresentadas anteriormente.

Como pode ser visto, a classificação de *chatbots* é um tópico relevante de pesquisa e amplamente abordado na literatura. Entretanto, para este trabalho, entende-se que a classificação mais adequada deva considerar os três componentes principais da arquitetura de um *chatbot*, apresentados na Figura 3.1, refletindo assim os aspectos destacados pelos autores de [1, 24, 26, 22, 23, 72, 25]:

- o domínio de conhecimento, sendo as fontes para as respostas;
- as abordagens de geração das respostas; e
- as técnicas utilizadas para entender da necessidade do usuário.

Estes três aspectos refletem dimensões pelas quais se pode classificar os *chatbots*, conforme destacado na Figura 3.2. A primeira dimensão se refere à quantidade e à qualidade das informações que o *chatbot* tem à disposição para entender o usuário e gerar uma resposta satisfatória, trata-se do domínio de conhecimento, que pode ser categorizado em aberto ou fechado. As estratégias utilizadas pelo assistente virtual para gerar respostas às solicitações do usuário define a segunda dimensão, a qual diferencia os *chatbots* entre generativos ou baseados em recuperação de informação. Por fim, a terceira dimensão direciona quanto as técnicas empregadas pelo sistema para entender o que o usuário solicita, classificando-os em baseados em regras ou em intenções.

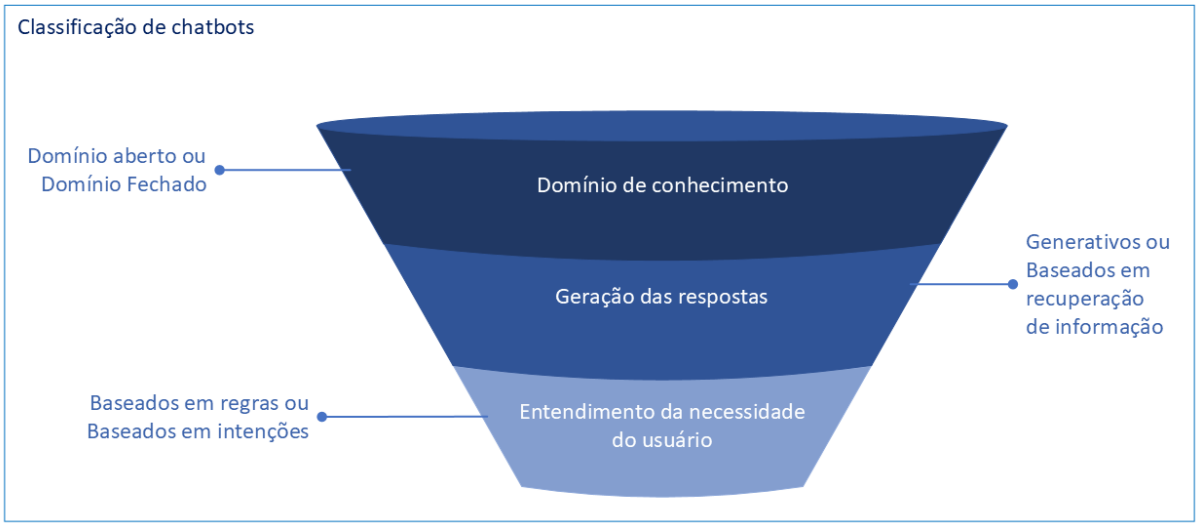


Figura 3.2: Taxonomia de *chatbots*.

Fonte: autoria própria.

3.3.1 Domínio de conhecimento

As bases utilizadas refletem o domínio de conhecimento do *chatbot*, que pode ser de domínio aberto ou domínio fechado [23][72].

Os *chatbots* de domínio aberto são aqueles desenvolvidos para responder a uma ampla variedade de perguntas e consultas, independentemente do tema [22]. Eles são projetados para fornecer informações gerais sobre diversos tópicos e atender a necessidades básicas dos usuários. Conseguem conversar com os usuários de uma forma mais natural e espontânea, sem limitações de tarefas específicas ou fluxos de conversa.

Por outro lado, os *chatbots* de domínio fechado são mais especializados e focados em fornecer respostas em um determinado contexto [22]. Esses assistentes virtuais são treinados para entender intenções específicas e fornecer respostas que levem à conclusão

de uma tarefa. Eles têm um conjunto limitado de comandos e respostas pré-definidos, o que significa que só podem responder a perguntas dentro do seu domínio específico.

Em resumo, os *chatbots* de domínio fechado tendem a ser mais eficientes, porque são desenvolvidos para lidar com um assunto específico. Já os de domínio aberto, tendem a ser mais complexos e difíceis de avaliar, uma vez que tratam de um número indeterminado de assuntos [23].

3.3.2 Geração das respostas

A abordagem sobre técnicas de geração de respostas se refere aos métodos pelos quais um *chatbot* gera ou recupera respostas para os usuários. Esta dimensão classifica os assistentes virtuais em baseados em recuperação de informação e generativos.

Os *chatbots* baseados em recuperação de informação respondem às perguntas dos usuários ao recuperar informações pré-armazenadas em uma base de dados. Eles conseguem fornecer respostas precisas e confiáveis, mas podem ter limitações quando se trata de perguntas que não estão relacionadas às informações armazenadas na base de conhecimento utilizada. Neste tipo, é possível ter controle total sobre a qualidade das respostas, uma vez que elas não são geradas automaticamente [1].

Já os assistentes virtuais generativos são treinados para gerar respostas de forma autônoma, com base em um grande conjunto de dados de treinamento. Eles usam técnicas de aprendizado profundo para aprender com as perguntas e respostas dos usuários e, em seguida, gerar respostas apropriadas. *Chatbots* generativos são úteis em ambientes onde as perguntas podem ser imprevisíveis e requerem respostas criativas. Entretanto, não é possível controlar completamente os textos gerados, existindo, inclusive, o risco de cometer erros na formação das frases com informações que não condizem com a realidade [24].

3.3.3 Entendimento da necessidade do usuário

As estratégias pelas quais um assistente virtual reconhece e entende a necessidade do usuário são determinantes para serem fornecidas respostas relevantes. Neste contexto, pode-se classificar os *chatbots* em baseados em regras e baseados em intenções.

Os *chatbots* baseados em regras utilizam um conjunto de regras pré-definidas para identificar a necessidade do usuário e fornecer respostas apropriadas. Essas regras buscam refletir as possíveis perguntas que os usuários farão ao robô. Para tal, podem ser utilizados linguagens (*e.g.* AIML) ou *frameworks* (*e.g.* RiveScript [48]) específicos. Embora os *chatbots* deste tipo sejam relativamente simples de implementar e possam ser eficazes para tarefas específicas, eles possuem grandes limitações na compreensão de linguagem

natural. Desta forma, à medida que a entrada se torna mais natural ou o domínio se move para o aberto, a eficiência das abordagens baseadas em regras se deteriora [25].

No sentido contrário, os assistentes virtuais baseados em intenção utilizam técnicas de processamento de linguagem natural para analisar semanticamente as consultas do usuário e, em seguida, recorrem a abordagens de aprendizado de máquina para classificar estas consultas em intenções [26], visando fornecer respostas relevantes. Nessa abordagem, o *chatbot* é treinado em um conjunto de dados que contém exemplos de perguntas e respostas para um determinado domínio ou tarefa. Embora possam ser mais complexos de implementar do que os *chatbots* de regras, eles têm a vantagem de serem mais adaptáveis e flexíveis, podendo ser treinados para lidar com uma ampla variedade de situações.

3.3.4 Considerações sobre a taxonomia

Vale destacar que essa classificação não é exaustiva e que diferentes abordagens podem ser combinadas para se obter melhores resultados. Por exemplo, é possível construir um *chatbot* de domínio aberto, generativo e baseado em intenções. Assim como também é viável desenvolver um assistente virtual de domínio fechado, baseado em recuperação de informação e em regras.

Esta flexibilidade, apesar de útil, apresenta desafios extras no processo de desenvolvimento de *chatbots*, em especial quanto à avaliação destes. Assunto abordado na seção seguinte.

3.4 Categorias de avaliação de *chatbots*

Assim como em qualquer outro sistema de computador, o objetivo de se avaliar um *chatbot* é garantir que ele funcione corretamente e atenda de maneira efetiva as necessidades de seus usuários. Entretanto, a avaliação deste tipo de sistema apresenta algumas particularidades, como a necessidade de avaliar a capacidade de entendimento da linguagem natural (complexa por natureza) e de responder de maneira apropriada (considerando o contexto e a intenção do usuário), o que torna esta tarefa desafiadora [1].

Aliado a isto, é importante destacar que a ampla adoção dos *chatbots* no atendimento direto aos usuários implica em uma preocupação maior de se garantir a qualidades destes sistemas, tendo em vista que falhas pós-implantação podem ocasionar perdas financeiras, de reputação e de audiência, dificilmente recuperadas [11].

Neste cenário, apesar de existir uma série de trabalhos relacionados à avaliação de assistentes conversacionais, alguns autores afirmam que ainda se trata de um problema de pesquisa em aberto [1, 21], devido à falta de um padrão estabelecido e amplamente aceito, dificultando a comparação entre diferentes *chatbots*. Isso se deve, em parte, à

variedade de tipos existentes, desde os baseados em recuperação de informação até os generativos, passando pelos de domínio aberto e fechado, conforme explanado na Seção 3.3.

De todo modo, é possível categorizar a avaliação destes sistemas em duas dimensões principais: método de avaliação e tipo de interação.

- Método de avaliação: cobre as avaliações manuais e automatizadas [1, 22, 40]; e
- Tipo de interação: trata das avaliações interativas e estáticas [40, 12].

Cada uma destas abordagens apresenta vantagens e desvantagens, dependendo do objetivo da avaliação e do *chatbot* avaliado, conforme apresentado a seguir.

3.4.1 Método de avaliação

Essa dimensão se refere à forma como a avaliação do *chatbot* é realizada, se por meio de uma avaliação conduzida por humanos (manual) ou por meio de um processo automatizado, sem a intervenção humana.

A avaliação manual envolve a análise humana do desempenho do *chatbot*, geralmente por meio da observação de interações entre o usuário e o robô. É comum que avaliadores humanos avaliem a qualidade das respostas do *chatbot*, bem como sua capacidade de entender às perguntas realizadas. Em outras palavras, a avaliação humana captura a interpretação subjetiva do ponto de vista do usuário [40].

Embora esta abordagem permita medir o desempenho do assistente conversacional em situações complexas e avaliar a naturalidade e personalidade das respostas, há desvantagens a serem consideradas. A avaliação manual pode ser demorada, imprecisa e dispendiosa [9], dificultando sua escalabilidade. Além disso, os resultados podem ser influenciados pelos avaliadores, levando a variações e vieses [1].

Já a avaliação automática utiliza métricas e algoritmos para avaliar o desempenho do *chatbot*. Por isso, tende a ser mais eficiente em termos de tempo e recursos necessários, quando comparada à avaliação manual. Entretanto, além de não haver ainda um padrão amplamente aceito na indústria e na academia, as métricas de avaliação automatizada parecem não ter a capacidade para avaliar corretamente a qualidade, eficiência e eficácia da conversa [1].

De toda forma, a rapidez, a escalabilidade e a objetividade dos resultados, além da capacidade de avaliar grandes volumes de interações rapidamente, torna esta abordagem mais atrativa no processo de desenvolvimento de *chatbots*.

3.4.2 Tipo de interação

Essa dimensão trata do tipo de interação que ocorre durante a avaliação do *chatbot*, se é um processo interativo, em que o avaliador pode interagir com o robô, ou se é um processo estático, no qual o *chatbot* é avaliado apenas com base em sua estrutura e conteúdo.

A avaliação estática se concentra em avaliar a estrutura do projeto de modelagem dos fluxos conversacionais de um *chatbot*, sem que haja uma interação de conversa. O objetivo principal desta abordagem é identificar antecipadamente problemas de modelagem que podem impactar negativamente o uso do *chatbot*. Neste tipo de avaliação, a estrutura de conversação, os fluxos de diálogos e o mapeamento de intenções e entidades, entre outros aspectos relacionados ao *design* do *chatbot*, são analisados visando identificar problemas que podem levar a uma experiência ruim de usuário, como informações imprecisas, respostas inapropriadas ou fluxos de conversação confusos [12].

Por sua vez, a avaliação interativa é realizada através da interação entre o *chatbot* e os usuários, reais ou simulados, em uma conversa. Esta abordagem permite avaliar a qualidade das respostas fornecidas para cada pergunta realizada e é, geralmente, conduzida por meio de testes de usuário. Nestes testes, os usuários interagem com o *chatbot* em diferentes cenários e a qualidade das respostas é avaliada por especialistas ou métricas objetivas. Ela possibilita avaliar a capacidade do assistente conversacional em fornecer respostas corretas, para atender às necessidades do usuário [73].

De forma geral, ambas as abordagens são importantes para avaliar a qualidade de um *chatbot* e se complementam entre si. Se a avaliação estática permite detectar problemas estruturais na modelagem dos fluxos conversacionais, a avaliação interativa possibilita avaliar o desempenho do *chatbot* durante a interação com o usuário. Por fim, é importante frisar que estas avaliações podem ser realizadas tanto manual quanto automatizadamente.

3.5 Revisão da literatura sobre avaliação de *chatbots*

Como discutido anteriormente, a avaliação de *chatbots* é uma área de estudo que gera interesse na comunidade acadêmica, existindo uma série de publicações relacionadas ao tema. Alguns destes trabalhos foram apresentados e discutidos em *surveys* publicados recentemente, como nos estudos de Maroengsit *et al.* (2019) [22], Deriu *et al.* (2021) [8], Caldarini *et al.* (2022) [1] e Motger *et al.* (2023) [2].

Neste cenário, para melhor compreender as principais técnicas e abordagens de avaliação disponíveis, foi realizada uma busca sistemática nas bases de conhecimento especificadas na introdução deste capítulo, que considerou os trabalhos científicos publicados nos últimos cinco anos, de 2019 até 2024. As palavras-chave utilizadas foram: *chatbot performance evaluation, conversation quality, chatbot testing, chatbot design evaluation, chatbot*

metrics, chatbot quality assurance, e suas combinações. O termo "*chatbot*" também foi intercambiado com os sinônimos apresentados na Seção 3.1.

Conforme apresentado no Capítulo 1, o objetivo deste trabalho é propor uma abordagem automatizada para avaliar *chatbots*. Portanto, nesta revisão da literatura, foram excluídos os estudos que se basearam exclusivamente na abordagem de avaliação manual. Assim, a pesquisa se concentrou em publicações que tratavam da avaliação automática de *chatbots*, com foco na dimensão "tipo de interação" — avaliações estáticas ou interativas [40, 12]. Surpreendentemente, foi encontrada uma quantidade limitada de trabalhos que atendiam a esses requisitos no período e nas fontes pesquisadas. Neste cenário, as seções seguintes detalham os trabalhos encontrados nesta pesquisa, organizando-os por categoria — avaliações estáticas (Seção 3.5.1) e interativas (Seção 3.5.2) — e ordem cronológica.

3.5.1 Avaliação estática

Embora a avaliação estática auxilie na detecção de problemas estruturais de um *chatbot*, que podem impactar negativamente no seu desempenho, foram identificados poucos trabalhos que priorizam essa abordagem. Esta situação é surpreendente, considerando que a avaliação estática é passível de automatização e capaz de identificar pontos de melhorias mesmo sem a interação entre o usuário e o assistente virtual.

Neste cenário, Gao *et al.* (2021) [11] propuseram uma abordagem computacional para avaliar a qualidade dos *chatbots* antes de implantá-los, visando reduzir custos e prevenir possíveis perdas econômicas e de audiência. A abordagem extrai e analisa 48 recursos do *design* bruto do assistente conversacional, cobrindo suas intenções, fluxos de conversação e textos das respostas, sem depender da interação com o usuário. O estudo analisou 1.050 *chatbots* de domínio fechado, baseados em recuperação de informação e intenções. Os autores classificaram estes projetos em "populares" e "impopulares" e, na sequência, seus dados foram utilizados como base de treinamento para um modelo de predição que indica, conforme a estrutura e o conteúdo do *chatbot*, qual a probabilidade de sucesso daquele projeto. As métricas avaliadas incluíram características das intenções, respostas e fluxos de conversa. Em relação às intenções e respostas, os autores avaliaram (i) a legibilidade dos textos, (ii) a utilização de sinais linguísticos e (iii) o uso de palavras simples e representativas do idioma. Além disso, estabeleceram uma relação entre a quantidade de intenções e complexidade, sendo mais complexos aqueles projetos com maior quantidade de intenções mapeadas. No que tange os fluxos de conversa, foi avaliada a densidade destes, através da observação das quantidades de vértices (*i.e.*, nós de diálogo) e as conexões entre eles. Como resultado, o trabalho sugere que fluxos de conversa com baixa densidade indicam maior concisão, o que é mais bem recebido pelos usuários. Além disso, verificou-se que *chatbots* mais complexos obtêm avaliações mais positivas. Por fim,

os autores afirmam ser indicado usar vocabulários comuns e representativos ao projetar intenções e respostas. No entanto, uma das limitações do trabalho é a necessidade do envolvimento humano para rotular as conversas concluídas com sucesso, utilizadas como gabarito para o modelo de predição.

Já Cañizares *et al.* (2022) [12] abordaram em seu trabalho o problema da falta de suporte para medir estaticamente as propriedades dos *chatbots* projetados, como indicadores de tamanho, complexidade, qualidade e usabilidade, em diversas plataformas de desenvolvimento de mercado. Para mitigar esse problema, os autores propõem um conjunto de 20 métricas estáticas, divididas em três conjuntos: (i) estatísticas de conceitos do *chatbot*, (ii) legibilidade das respostas ou complexidade dos enunciados das intenções e (iii) complexidade dos diálogos, que são independentes da plataforma de implementação. A proposta inclui uma ferramenta chamada Asymob, que suporta a tradução de *chatbots* definidos em diversas plataformas para o formato neutro das métricas, viabilizando assim realizar a medição. Além disso, as métricas são classificadas quanto ao impacto potencial nas dimensões de eficácia, eficiência e satisfação do usuário. Os autores avaliaram as métricas em uma coleção de *chatbots* de domínio fechado, baseados em recuperação de informação e intenções, implementados nas ferramentas *Dialogflow* [15] e Rasa NLU [17]. No estudo, mostraram que essas métricas ajudaram a detectar problemas de qualidade e serviram de base para comparar *chatbots* de diferentes origens e tecnologias. No entanto, a proposta não define um limiar mínimo ou um valor-alvo para as métricas, o que pode limitar sua utilidade em resolver o problema da subjetividade da avaliação. Cañizares *et al.* concluem que a utilização deste conjunto de métricas, para avaliar a qualidade do projeto de um *chatbot*, pode ser uma ferramenta para orientar e controlar a qualidade do sistema durante todo o seu desenvolvimento, tornando-se um complemento para o teste interativo. Estas métricas podem descobrir problemas que não são o alvo do teste interativo e podem ser usadas para acionar recomendações de melhoria do *design* do *chatbot*.

3.5.2 Avaliação interativa

Na categoria de artigos específicos sobre avaliação interativa, foram encontrados mais trabalhos que discutem como avaliar a qualidade de um *chatbot* a partir da interação, real ou simulada, entre o usuário e o assistente conversacional. O resultado desta revisão da literatura indica que esta é a abordagem mais utilizada nos trabalhos de pesquisa publicados. Conforme destacado nos cinco trabalhos descritos a seguir.

Neste contexto, Deriu e Cieliebak (2019) [74] estudaram como avaliar de forma automatizada sistemas de diálogo conversacional, visando correlacionar com as avaliações humanas. Segundo os autores, as técnicas mais comuns de avaliação são baseadas em *crowdsourcing*, ou em métricas automatizadas, têm uma correlação pobre com as avalia-

ções humanas. Além disso, as métricas treinadas existentes ainda têm limitações, como depender de um contexto estático, que não refletem o comportamento do sistema de diálogo em uso. Assim, o trabalho propõe uma abordagem baseada em diálogos gerados automaticamente, avaliados por juízes humanos; estes dados são utilizados como base de treinamento de um modelo de regressão para prever a qualidade do sistema. O método, chamado AutoJudge, apresentou alta correlação com as avaliações humanas e pode ser aplicado para avaliar sistemas de diálogo sem depender dos contextos estáticos (*e.g.*casos de teste especificados por especialistas). O processo seguiu três fases: (i) geração de dados (utilizaram um conjunto variado de modelos para gerar diálogos de forma automática), (ii) anotação dos dados (utilizaram humanos para rotular a qualidade dos pares pergunta-resposta) e (iii) implementação de melhorias (com base nos dados anotados, foi treinado um modelo de regressão para prever a qualidade das respostas do *chatbot*). Como o foco principal do trabalho foi avaliar a correlação da proposta com o julgamento humano, e não a qualidade das respostas propriamente ditas, as métricas avaliadas foram as correlações de *Pearson* [75] e de *Spearman* [76] e o *Mean Absolut Error* (MAE). Apesar de ser um trabalho que recorre à geração automática de diálogo, tem como limitação a necessidade do julgamento humano no processo de avaliação.

Já Forkan *et al.* (2020) [9] propõem a ferramenta ECHO para avaliar automaticamente *chatbots* baseados em nuvem em diferentes domínios de conversação. A ferramenta utiliza uma arquitetura sistemática para integrar e avaliar *chatbots* e uma estrutura de avaliação comum que permite comparar os resultados produzidos por *chatbots* de múltiplas nuvens. Os testes foram realizados em dois domínios de conversação e as métricas avaliadas foram: tempo médio de resposta, taxa de *fallback* (que refere-se ao número de vezes que o *chatbot* não conseguiu entender a pergunta do usuário dentro do escopo em uma conversa), *comprehensive rate* (usada para medir a capacidade de corrigir erros nas entradas do usuário), acurácia, precisão, *recall* e *F1-score*. O trabalho utiliza conceitos de verdadeiros positivos e negativos e falsos positivos e negativos para obter uma metodologia mais objetiva e assertiva nas avaliações. Como limitações do estudo, destaca-se a necessidade de conhecer previamente a estrutura do *chatbot* e de criar manualmente uma base rotulada de testes, além do fato de os autores terem focado os testes apenas na dimensão assertividade.

Na proposta de Deriu *et al.* (2020) [21], uma avaliação capaz de pontuar os *chatbots* quanto a sua capacidade de imitar um humano em uma conversa foi analisada. Os autores destacaram que atualmente os principais problemas no contexto de avaliação de assistentes conversacionais são o custo e a ineficiência das avaliações humanas, além da falta de correlação entre as avaliações automáticas e humanas. Para resolver isso, eles propõem uma estrutura de avaliação que substitui conversas entre *bots* e humanos por conversas

entre *bots*. Nessa proposta, as mensagens de teste foram geradas com base no *dataset* de diálogos DailyDialog [77] e usando os modelos de linguagem GPT-2 [78] e BERT-Rank [79]. As conversas foram avaliadas por humanos, que opinaram se determinadas mensagens eram produzidas por robôs ou humanos. Os autores sugeriram uma métrica chamada análise de sobrevivência, que mede por quanto tempo o *chatbot* consegue manter o comportamento humano. Para validar a proposta, foi realizado um experimento com três *chatbots* de domínios distintos. Embora os autores tenham mencionado que se trata de uma abordagem econômica e repetível, a proposta não é uma automação da avaliação, mas sim uma metodologia que requer o envolvimento humano no trabalho de avaliação. Além disso, outra limitação da proposta é que ela não avalia a assertividade dos *chatbots*, mas sim a capacidade deles imitarem o comportamento humano.

Seguindo um caminho diferente, Bravo-Santos *et al.* (2020) [10] propuseram uma metodologia para testes automatizados de *chatbots* usando a ferramenta CHARM. Eles apontam que as principais ferramentas de mercado, como Watson Assistant [14], Dialog-Flow [15] e Amazon Lex [16], não oferecem suporte para testes em *chatbots*, o que pode prejudicar a qualidade final do projeto. A metodologia consiste em criar casos de teste para avaliar a robustez do mecanismo de Processamento de Linguagem Natural (PLN) e a precisão do assistente virtual na identificação das intenções do usuário. Para isso, os autores propõem testes que incluem variações das frases de treinamento do *chatbot*, construídas por meio de funções de mutação. A ferramenta gera mensagens com variações em relação aos exemplos de treinamento originais, aplicando operadores de caracteres (que emulam erros de digitação), linguagem (para criar expressões diferentes, mas com mesmo significado), palavras (substituindo palavras originais por sinônimos) e números (substituindo números por escrita por extenso e vice-versa). Os casos de teste são executados e interpretados em três dimensões: coerência, robustez e precisão. O teste de coerência é o mais simples, executado sem nenhuma mutação, com objetivo de detectar defeitos de baixa granularidade (*e.g.* intenções muito semelhantes). O teste de robustez avalia o quão bom é o *chatbot* em lidar com erros de digitação, sendo executado com as mutações de caracteres e de números. Por último, o teste de precisão avalia a capacidade do *chatbot* de prever a intenção correta, quando as mensagens têm formulação diferente das frases de treinamento, assim recorre às mutações de linguagem e de palavras. Os autores concluem que a solução contribui para a melhoria da qualidade final dos projetos de *chatbot*. No entanto, é importante ressaltar que a metodologia foi testada apenas com assistentes virtuais simples, o que não garante sua eficácia em contextos mais complexos. Além disso, é necessário haver intervenção humana para filtrar declarações sem sentido, geradas nas mutações de linguagem e de palavras. O escopo deste trabalho foi delimitado a *chatbots* de domínio fechado, baseados em intenções e recuperação da informação.

Por sua vez, Yang *et al.* (2022) [13] também abordam o problema da avaliação automática de sistemas de diálogo. Mas argumentam que o uso de roteiros de teste estáticos como referência dificulta a obtenção de dados precisos e demanda acesso aos modelos dos *bots* para um teste mais completo. Além disso, as avaliações interativas conduzidas por humanos são caras e demoradas. Para solucionar essas dificuldades, propuseram uma estrutura interativa de avaliação na qual os *chatbots* competem entre si como em um torneio, usando métricas flexíveis de pontuação. Esse método, chamado de ChatMatch, permite comparar *chatbots* quanto a sua eficiência, independentemente de sua arquitetura de modelo e domínios de treinamento. O método usa três algoritmos para pontuar a diversidade, consistência e relevância das respostas dos *bots*. Os resultados dos testes apontam que a proposta produz avaliações que se correlacionam bem com avaliações realizadas por humanos. No entanto, é importante ressaltar que o julgamento humano é necessário para rotular as respostas como verdadeiras.

3.5.3 Discussão sobre avaliação de *chatbots*: avanços, limitações e direções futuras

Esta seção aborda os avanços e limitações encontrados nos trabalhos anteriores, visando fornecer uma análise comparativa das abordagens e métricas utilizadas. A partir dessas informações, são identificadas as lacunas e propostas direções futuras para preenchê-las. Essa discussão oferece uma visão crítica do estado atual da pesquisa em avaliação de *chatbots*, permitindo uma reflexão sobre possíveis melhorias nas abordagens existentes. Essas considerações também servem como orientação para futuros estudos, incluindo esta dissertação de mestrado.

Com base no que foi apresentado nas Seções 3.5.1 e 3.5.2, a Tabela 3.1 fornece um resumo dos trabalhos discutidos anteriormente, possibilitando a comparação entre eles quanto às abordagens, métodos e métricas utilizados para avaliar os *chatbots*. A tabela possui nove colunas, sendo a primeira destinada à identificação dos atributos observados na comparação. As colunas de 2 a 8 apresentam informações sobre os trabalhos estudados, enquanto a última coluna representa o que seria uma proposta mais abrangente para a avaliação desses sistemas. Esta proposta incluiria uma abordagem mais ampla quanto aos tipos de *chatbots* e à forma de avaliação, bem como o uso de métricas que abrangessem todas as categorias mencionadas na Seção 2.2.

Para esta comparação, as linhas da tabela representam:

- **Método de avaliação:** conforme Seção 3.4.1, pode ser A (automático) ou M (quando houver uso, em qualquer grau, do método manual);

- **Tipo de interação:** pode assumir os valores E (estática), I (interativa) ou "Amb." (para ambos os tipos), conforme a Seção 3.4.2;
- **Métricas de desempenho:** representada por "✓" quando utilizada, pelo menos, uma das métricas apresentadas na Seção 2.2.1, ou por "-" caso contrário;
- **Métricas de satisfação do usuário:** simbolizada pelo "✓" quando uma das métricas da Seção 2.2.2 é utilizada, ou por "-" caso contrário;
- **Métricas de qualidade das respostas:** similar ao item anterior, mas em relação às métricas da Seção 2.2.3;
- **Métricas de qualidade do diálogo:** também utiliza o "✓" quando uma das métricas explicadas na Seção 2.2.4 é utilizada, ou por "-" caso contrário;
- **Métricas de qualidade de *design*:** assim como os anteriores, pode ser "✓" (quando presente as métricas apresentadas na Seção 2.2.5) ou "-", caso contrário;
- **Domínio de conhecimento:** conforme Seção 3.3.1, pode ser DF (domínio fechado), DA (domínio aberto), "Amb." (para ambos os domínios) ou N/I (quando não identificado);
- **Geração de resposta:** representada por RI (recuperação de informação), GE (generativos), "Amb." (para ambos os tipos de geração de resposta) ou N/I (quando não identificado), conforme a Seção 3.3.2;
- **Entendimento da necessidade:** pode assumir IT (baseado em intenções), RE (baseado em regras) ou N/I (quando não identificado), como especificado na Seção 3.3.3.

A comparação realizada possibilitou identificar algumas lacunas ainda em aberto nessa área. A primeira delas é a constatação de que a avaliação manual ainda é amplamente utilizada, apesar de suas desvantagens em relação ao custo, imprecisão e tempo de execução [8]. Mesmo os trabalhos que se apresentam como abordagens automatizadas dependem, em maior ou menor grau, da intervenção humana no processo, como [9, 21, 74]. Isso indica uma necessidade de desenvolvimento de novas abordagens de avaliação de *chatbots* que priorizem processos automatizados, eficientes e repetíveis.

Também foi identificada uma lacuna em relação à análise da qualidade da estrutura e do conteúdo dos assistentes virtuais, uma vez que poucos estudos se concentram nesse aspecto. Essa abordagem pode servir para detectar erros com antecedência, ajudando a diminuir o custo do processo de desenvolvimento desses sistemas e a reduzir o impacto negativo do *chatbot* em produção. Com exceção dos trabalhos [11] e [12], que propõem o

Tabela 3.1: Trabalhos sobre avaliação de *chatbots*.

Atributo observado	Propostas							
	[11]	**[12]**	**[74]**	**[9]**	**[21]**	**[10]**	**[13]**	*
Método de avaliação	A	A	M	M	M	A	A	A
Tipo de interação	E	E	I	I	I	I	I	Amb.
Métricas de desempenho	-	-	-	✓	-	✓	-	✓
Métricas de satisfação do usuário	-	-	-	-	✓	-	-	✓
Métricas de qualidade das respostas	✓	✓	✓	-	-	-	✓	✓
Métricas de qualidade do diálogo	-	-	-	✓	-	-	✓	✓
Métricas de qualidade de *design*	✓	✓	-	-	-	-	-	✓
Domínio de conhecimento	DF	DF	N/I	DF	DA	DF	DA	Amb.
Geração de resposta	RI	RI	N/I	RI	GE	RI	GE	Amb.
Entendimento da necessidade	IT	IT	N/I	IT	IT	IT	IT	IT

(*) Representa o que seria uma proposta mais abrangente para avaliação de *chatbots*.

uso da avaliação estática, a maioria dos estudos prioriza a avaliação por meio de interação com o *chatbot*.

Além disso, os trabalhos avaliados neste estudo, e apresentados na Tabela 3.1, frequentemente se limitam ao uso de métricas de apenas duas categorias. Por exemplo, alguns focam no desempenho e qualidade de *design*, enquanto outros consideram a qualidade das respostas e dos diálogos. No entanto, essa abordagem negligencia outras métricas relevantes, resultando em uma avaliação limitada e superficial da qualidade do *chatbot*. Para uma avaliação mais abrangente e precisa, faz-se necessário considerar uma maior quantidade de categorias de métricas.

Por fim, verificou-se também que os estudos analisados se limitaram a utilizar apenas uma das abordagens de avaliação, seja estática ou interativa. No entanto, a adoção conjunta destas abordagens pode gerar benefícios significativos no processo de avaliação dos *chatbots*, uma vez que os resultados obtidos podem ser complementares. Apesar da avaliação estática possibilitar a identificação de pontos de melhoria no *design* ou no conteúdo de treinamento do *chatbot*, sem a interação com ele é impossível avaliar sua assertividade. Por outro lado, a avaliação interativa permite identificar erros nas respostas dadas aos usuários, que podem ser consequências de problemas no treinamento do assistente virtual, sendo de difícil detecção sem analisar sua estrutura.

Essas lacunas apontam para a necessidade de aprofundamento em novas abordagens de avaliação de *chatbots* e indicam possíveis direções para futuros trabalhos de pesquisa. É importante destacar que a avaliação de *chatbots* é fundamental para garantir a qualidade e a eficácia desses sistemas, especialmente em um contexto em que a interação humano-máquina se torna cada vez mais frequente e relevante. Portanto, o desenvolvimento de

novas abordagens de avaliação de *chatbots* é essencial para aprimorar a experiência do usuário e garantir a efetividade desses sistemas.

Sendo assim, e segundo a coluna (*) da Tabela 3.1, uma abordagem desejável para avaliar *chatbots* é aquela que permita uma avaliação abrangente e integrada, considerando tanto aspectos estáticos quanto interativos de forma automatizada, eliminando a necessidade de conhecimento prévio do conteúdo do assistente virtual ou a criação manual de roteiros de testes. Essa proposta possibilitaria a avaliação dos *chatbots* independente das suas classificações em termos de domínio de conhecimento ou geração de respostas. Para isso, é importante observar e medir métricas que abranjam todas as categorias apresentadas na Seção 2.2.

Diante desse contexto, propor soluções que busquem melhorias no processo de avaliação de *chatbots*, conforme indicado anteriormente, representa um avanço em relação ao estado da arte atual. É nesse sentido que este trabalho se propõe a avançar, conforme será detalhado no Capítulo 4 deste documento, buscando oferecer uma solução que se aproxime da abordagem mais abrangente mencionada na Tabela 3.1

3.6 Resumo do capítulo

Este capítulo abordou diversos tópicos relacionados aos *chatbots*, a partir de uma pesquisa do estado da arte realizada para este trabalho. Inicialmente, a tecnologia foi conceituada e sua evolução histórica foi brevemente apresentada. Em seguida, foram discutidas as taxonomias para classificar os assistentes virtuais, considerando o domínio de conhecimento e as abordagens para gerar respostas e entender as necessidades dos usuários. A avaliação dos *chatbots* foi outro ponto abordado, categorizando-a por método (automático ou manual) e por tipo de interação (estática ou interativa). Tendo sido este último tópico detalhado com a apresentação dos estudos dedicados ao tema.

Por fim, foi realizada uma discussão sobre a avaliação de *chatbots*, pontuando os avanços alcançados, as limitações existentes e as direções futuras a serem exploradas. Também foi apresentada uma proposta que visa realizar uma análise automática e abrangente, abordando tanto métricas de avaliação estática quanto iterativa, de modo a contemplar elementos além do que é usualmente considerado nos estudos atuais. Caminho pelo qual este trabalho almeja seguir.

Capítulo 4

DUBI: um *framework* para avaliação automática de *chatbots*

O objetivo deste capítulo consiste em apresentar o *framework* DUBI, que se caracteriza por ser uma metodologia de avaliação automática de *chatbots* que engloba tanto aspectos estáticos quanto interativos do sistema. Serão detalhados os aspectos relevantes relacionados à sua arquitetura, módulos e componentes, com o intuito de proporcionar uma compreensão do funcionamento da solução. Além disso, também serão apresentadas as métricas empregadas durante o processo de avaliação, bem como discutidos os benefícios e diferenciais do *framework* DUBI em relação aos estudos anteriores.

Para isso, a estrutura deste capítulo é assim organizada: a Seção 4.1 fornece uma visão geral da proposta de solução, enquanto a sua arquitetura é apresentada na Seção 4.2, os módulos do *framework* DUBI são detalhados nas seções 4.3 e 4.4, respectivamente, já os benefícios e diferenciais da proposta são discutidos na Seção 4.5 e, por último, o capítulo é finalizado com um resumo do que foi apresentado, disponível na Seção 4.6.

4.1 Visão geral da proposta

Embora os estudos atuais tenham contribuído relevantemente e apresentado avanços consideráveis no cenário de avaliação de *chatbots*, a análise do estado da arte, apresentada no capítulo anterior, revelou lacunas nesse campo de pesquisa.

Conforme discutido na Seção 3.5.3, as soluções disponíveis ainda não são abrangentes o suficiente para abordar uma avaliação ampla dos assistentes virtuais. Essa limitação se deve, principalmente, ao foco desses estudos em apenas uma abordagem de avaliação, seja ela estática ou interativa. E, por isso, acabam por priorizar uma pequena parcela de métricas passíveis de observação em um projeto de *chatbot*. Além disso, a automação completa do ciclo de avaliação também tem se mostrado um desafio aos pesquisadores. Estas

características afastam tais propostas do objetivo de viabilizar às equipes desenvolvedoras de assistentes virtuais um diagnóstico amplo sobre a qualidade de seus projetos.

Esta situação é agravada pela falta de recursos adequados para automatizar os testes em projetos de *chatbots* por parte dos principais fornecedores de serviços de conversação baseados em inteligência artificial, tais como IBM *Watson Assistant* [14], Google *Dialog-Flow* [15], Amazon *Lex* [16] e RASA NLU [17], comprometendo a qualidade destes projetos [10, 12]. Diante disso, o SERPRO decidiu desenvolver sua própria plataforma, chamada Serprobots, para construção e gerenciamento de assistentes virtuais. A plataforma Serprobots utiliza os motores de conversação disponíveis no mercado, além de oferecer recursos adicionais não encontrados nas soluções mencionadas anteriormente, conforme detalhado na Seção 2.3. A partir dela, o SERPRO desenvolve todos os seus assistentes virtuais, tanto para uso interno quanto para seus clientes. Portanto, a disponibilização de mecanismos automáticos de avaliação é essencial à plataforma, visando proporcionar maior qualidade a estes projetos.

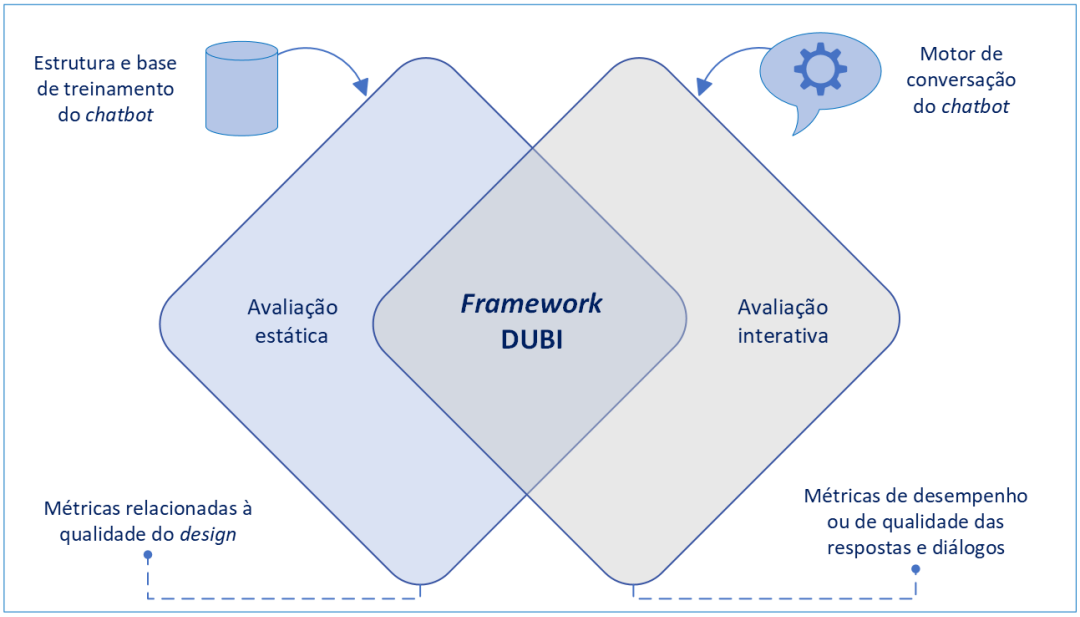


Figura 4.1: *Framework* DUBI: combina os benefícios das abordagens estática e interativa.

Fonte: autoria própria.

Para se aproximar deste objetivo, é necessária uma solução que avalie os *chatbots* tanto em termos estáticos quanto interativos, de forma totalmente automatizada, sem intervenção humana na geração de casos de teste, visando uma vasta aferição das métricas de qualidade. É neste contexto que o presente trabalho se destaca, buscando oferecer uma avaliação ampla e automática, com foco nos aspectos estáticos relacionados à estrutura e ao conteúdo de treinamento, assim como nos aspectos interativos, por meio da interação com o *chatbot*, simulando comportamentos de usuários reais, a fim de viabilizar uma

análise quantitativa da assertividade do assistente virtual, conforme ilustrado na Figura 4.1. Assim, entende-se que essa abordagem preencherá as lacunas discutidas na Seção 3.5.3 deste documento, referente à falta de uma solução que avalie automaticamente assistentes virtuais, observando aspectos estáticos e interativos. Além disso, também propiciará ao SERPRO ofertar serviços de *chatbots* de maior qualidade a seus clientes.

Neste cenário, este trabalho apresenta o *framework* DUBI, uma solução para a avaliação automática de *chatbots* que será integrada à plataforma Serprobots. As seções seguintes deste capítulo detalharão a proposta de solução, abordando a arquitetura e a estruturação do *framework*, destacando os benefícios da abordagem adotada e explicando as técnicas e métricas que serão utilizadas ao longo do processo de avaliação.

4.2 Arquitetura do *framework* DUBI

A presente seção tem por objetivo apresentar em detalhes a solução proposta deste trabalho, o *framework* DUBI, que, conforme informado no Capítulo 1, tem seu escopo delimitado pelos *chatbots* de domínio fechado, baseados em intenções e que utilizam a abordagem de recuperação de informação. Assim sendo, aqui serão descritos e discutidos a arquitetura e o funcionamento da solução, fornecendo uma visão detalhada da análise viabilizada.

Nesse sentido, o objetivo do *framework* DUBI é avaliar a qualidade e o desempenho de *chatbots*, identificando possíveis problemas e fornecendo métricas para uma análise detalhada. Para tal, ele é composto por dois módulos principais, o *Design Understanding* (DU) e o *chatBot Intelligence* (BI), responsáveis por realizar avaliações estáticas e interativas, respectivamente. O módulo DU realiza uma análise detalhada da estrutura e conteúdo de treinamento do assistente virtual, observando aspectos como características textuais e relacionamento entre elementos. Por outro lado, o módulo BI se concentra na avaliação interativa, envolvendo a geração automática de conjuntos de teste e a simulação de interações entre usuários e o *chatbot*, a partir dos dados contidos em sua base de treinamento.

Conforme representado na Figura 4.2, que apresenta a arquitetura do *framework* DUBI, o fluxo de funcionamento da solução é dividido em sete passos distintos, cada um desempenhando uma função específica para alcançar os objetivos propostos. Estes passos são:

- **Avaliação estática:**
  - **Passo 1:** a partir do conteúdo estático do *chatbot*, como intenções, exemplos de treinamento, textos de respostas e fluxos de conversas, o módulo *Design*

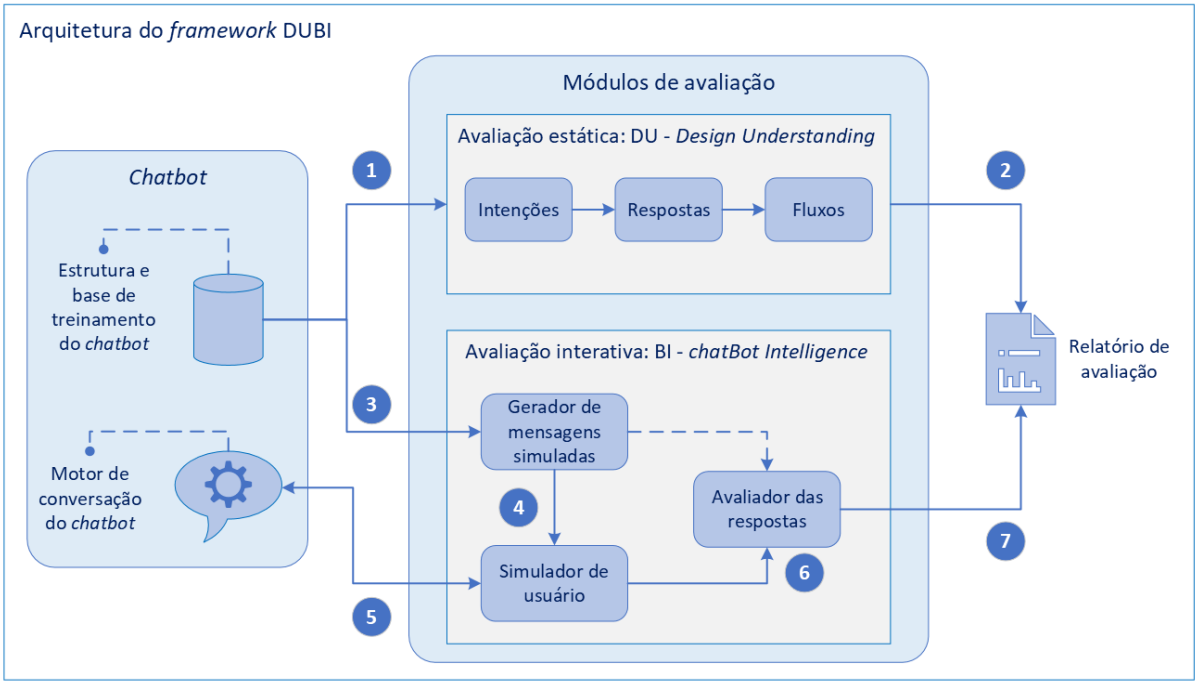


Figura 4.2: Arquitetura do *framework* DUBI.

Fonte: autoria própria.

Understanding (DU) realiza uma análise e avaliação detalhada dessas estruturas. Esse módulo utiliza diversas métricas relacionadas a essas estruturas para identificar possíveis pontos de melhoria e fornecer uma visão abrangente da qualidade do assistente virtual. As métricas utilizadas nesta etapa são apresentadas na Seção 4.3.

  - **Passo 2:** com base na avaliação realizada no passo anterior, são identificados pontos de melhoria no conteúdo estático do *chatbot*. Essa identificação assertiva de problemas estruturais, ou de conteúdo, possibilita uma abordagem proativa na resolução de questões que possam impactar negativamente o uso do assistente virtual. Os pontos de melhoria são organizados e incluídos como parte do relatório de avaliação da solução, fornecendo um panorama das áreas que requerem atenção.

- **Avaliação interativa:**

  - **Passo 3:** utilizando o conteúdo estático do *chatbot*, o *framework* gera automaticamente contextos de conversas, que serão interpretados como casos de teste. Esse é o início da avaliação interativa realizada pelo módulo *chatBot Intelligence* (BI). Os casos de teste consistem em sequências de mensagens criadas automaticamente, com base nos textos dos exemplos de treinamento das inten-

ções do *chatbot*. Para gerar essas mensagens, é utilizada uma abordagem de IA generativa, na qual modelos de linguagem são empregados para produzir conteúdo textual, conforme detalhado na Seção 4.4.1. Essa é a responsabilidade do componente "gerador de mensagens simuladas".

- **Passo 4:** após a geração das mensagens simuladas, elas são organizadas em casos de testes positivos e negativos, a serem utilizados pelo "simulador de usuário". Os casos de testes positivos são compostos por mensagens que se enquadram no escopo do *chatbot*, portanto se espera uma resposta com o conteúdo relacionado à pergunta. Já os casos de teste negativos são aqueles com mensagens fora do escopo do assistente virtual, nesse caso, é aguardado uma resposta de *fallback*. Todos os cenários de casos de teste gerados pelo "gerador de mensagens simuladas" são apresentados em detalhes na Seção 4.4.
- **Passo 5:** o "simulador de usuário" é o componente responsável por executar cada um dos casos de teste criados no passo anterior. Nesta execução, cada mensagem simulada é enviada ao *chatbot* e a resposta obtida é armazenada para posterior avaliação quanto a assertividade. Esse componente é encarregado de organizar e executar os cenários de teste, garantindo uma avaliação abrangente do desempenho do assistente virtual.
- **Passo 6:** após a execução da simulação, o componente "avaliador das respostas" compara as respostas obtidas pelo "simulador de usuário" com as respostas esperadas, fornecidas pelo "gerador de mensagens simuladas", com base nas informações da estrutura do *chatbot*. Essa comparação permite avaliar a assertividade do sistema em cada caso de teste executado, identificando possíveis falhas ou desvios em relação ao comportamento esperado.
- **Passo 7:** com base nas informações extraídas da interação com o assistente virtual, o resultado é inserido no relatório de avaliação, complementando as informações já produzidas no Passo 2. Nesse momento, são detalhadas as métricas obtidas a partir da interação com o *chatbot*, como acurácia, taxa de *fallback*, taxa de compreensão, entre outras, conforme detalhado na Seção 4.4.3. Essas métricas fornecem uma visão aprofundada do desempenho do *chatbot* e auxiliam na identificação de áreas que precisam ser aprimoradas.

Em resumo, o *framework* DUBI utiliza os módulos *Design Understanding* (DU) e *chatBot Intelligence* (BI) para realizar uma análise abrangente do conteúdo estático e do desempenho interativo do *chatbot*. Os sete passos descritos acima permitem identificar pontos de melhoria, criar casos de teste automatizados, simular interações com usuários e avaliar a assertividade do assistente virtual. Desta forma, a abordagem DUBI permite

avaliar o *chatbot* a qualquer momento, inclusive antes da sua implantação, com custos e esforços significantemente baixos. Essa avaliação é compilada em um relatório detalhado, que apresenta métricas relevantes e pode fornecer percepções valiosas para otimização contínua do *chatbot*.

Entende-se que esta estratégia atende às necessidades apontadas na Seção 3.5.3, quando foram discutidas as lacunas atualmente existentes, proporcionando assim uma contribuição à evolução do estado da arte desta linha de pesquisa. Assim, nas seções seguintes estes módulos serão descritos, explicando-se suas respectivas funções e abordagens utilizadas, bem como as métricas observadas por cada um deles.

4.3 O módulo DU - *Design Understanding*

A concepção e organização lógica dos componentes que estruturam um *chatbot*, tais como intenções, exemplos de treinamento, textos de respostas e fluxos de conversa, é o que define o *design* de um assistente virtual.

Estudos recentes, como os [11] e [12], indicam que um *design* mal estruturado pode resultar em respostas inadequadas, uma experiência frustrante para o usuário e, como consequência, perdas de audiência e financeira. Assim, um *design* bem estruturado é essencial para garantir uma interação eficaz e satisfatória. Portanto, a busca por essa garantia passa pela identificação de defeitos antes da implantação do *chatbot*. É neste contexto que o módulo *Design Understanding* está inserido, ou seja, realizar a avaliação estática de um assistente virtual a partir da observação de características dos componentes de seu *design*.

Para isso, essas características são agrupadas em aspectos relacionados ao *design*, permitindo a análise da qualidade e identificação de potenciais problemas. Os aspectos observados são: estrutura, relação e linguístico. O aspecto de estrutura visa observar características relacionadas à formação de cada componente, como tamanhos de texto, quantidade de exemplos e equilíbrio de classes. O aspecto de relação se concentra na identificação de características referentes ao relacionamento entre os componentes, como intenções órfãs, fluxos inalcançáveis ou ciclos. Já o aspecto linguístico tem como foco avaliar características que impactam na receptividade do *chatbot* pelos usuários, como legibilidade e simplicidade dos textos. A Tabela 4.1 apresenta as métricas observadas pelo módulo DU, categorizando-as por componente do *chatbot* e aspecto de *design*.

Parte destas métricas também foram utilizadas nos estudos [11] e [12], especializados na avaliação estática e discutidos na Seção 3.5.1. Essas métricas, apresentadas na Seção 2.2.5 deste documento, são respectivamente: LET, LTR, QCF, QET, RET, RTR, SID, STR, TET e TTR. Por outro lado, este trabalho inclui a observação de outras métricas

Tabela 4.1: Métricas observadas pelo módulo DU - *Design Understanding*.

Métrica	Descrição	Aspecto
Intenções		
QET	Quantidade de exemplos de treinamento	Estrutura
BET	Balanceamento dos exemplos de treinamento	Estrutura
SID	Similaridade entre intenções distintas	Relação
TET	Tamanho dos exemplos de treinamento	Estrutura
LET	Legibilidade dos exemplos de treinamento	Linguístico
RET	Representatividade dos exemplos de treinamento	Linguístico
QIO	Quantidade de intenções órfãs	Relação
Respostas		
TTR	Tamanho dos textos de resposta	Linguístico
LTR	Legibilidade dos textos de resposta	Linguístico
RTR	Representatividade dos textos de resposta	Linguístico
STR	Sentimento negativo dos textos das respostas	Linguístico
Fluxos de conversa		
QID	Qtd. de nós de diálogo com condições de entrada iguais	Estrutura
QDV	Qtd. de nós de diálogo com condição de entrada = $true$	Estrutura
QDF	Qtd. de nós de diálogo com condição de entrada = $false$	Estrutura
QDR	Qtd. de nós de diálogo sem texto de resposta	Estrutura
QCF	Qtd. ciclos nos fluxos de conversa	Relação

que podem influenciar diretamente na qualidade do *design* do *chatbot*, explicadas com mais detalhes na Seção 4.3.1, sendo elas: BET, QID, QDF, QIO, QDR e QDV. Desta forma, entende-se que haverá uma avaliação mais abrangente das características relacionadas à estrutura de um assistente virtual.

4.3.1 Métricas observadas no módulo DU

Conforme informado na seção anterior, o conjunto de métricas observadas e aferidas no módulo *Design Understanding* do *framework* DUBI inclui métricas já utilizadas em outros trabalhos disponíveis na literatura, e especificadas na Seção 2.2.5 deste documento. Mas este trabalho também propõe novas métricas a serem observadas, pois se entende que com essa extensão de métricas haverá uma melhor cobertura da avaliação estática, no que diz respeito aos aspectos relevantes de *design* de um *chatbot*.

Neste contexto, esta seção descreve as métricas extras propostas por esse trabalho, sendo elas: balanceamento da quantidade de exemplos de treinamento (BET), quantidade de intenções órfãs (QIO), quantidade de nós de diálogo com condições de entrada iguais (QID), quantidade de nós de diálogo com condição de entrada sempre verdadeira (QDV),

quantidade de nós de diálogo com condição de entrada sempre falsa (QDF) e a quantidade de nós de diálogo sem texto de resposta (QDR).

Este conjunto de métricas é resultado da observação empírica do SERPRO, adquirida por meio de sua experiência em desenvolvimento e sustentação da plataforma Serprobots, bem como da criação de diversos *chatbots* nessa mesma plataforma. Embora essas métricas possam não ter sido mencionadas em estudos anteriores, entende-se a importância delas para avaliar a qualidade do *design* dos assistentes virtuais, em complemento àquelas métricas discutidas na Seção 2.2.5. A seguir, tais métricas são detalhadas:

- **Balanceamento dos exemplos de treinamento (BET):** essa métrica é responsável por avaliar o balanceamento entre as quantidades de exemplos de treinamento das intenções do *chatbot*. Entende-se que essa avaliação é necessária, porque um desequilíbrio significativo pode prejudicar o desempenho e a precisão do assistente virtual em sua tarefa primordial de identificar a necessidade do usuário. Quando as intenções têm quantidades muito discrepantes de exemplos de treinamento, o *chatbot* pode se tornar enviesado, priorizando as intenções com mais exemplos e resultando em respostas inadequadas ou uma compreensão limitada das intenções menos representadas. Um balanceamento adequado dos exemplos de treinamento ajuda a garantir que o assistente virtual tenha um desempenho mais preciso ao lidar com diferentes intenções dos usuários. Neste sentido, a métrica BET de cada intenção é calculada pela equação

$$BET_I = \frac{QET_I - \mu_{ET}}{\mu_{ET}}, \tag{4.1}$$

  na qual, I representa a intenção analisada, QET_I a quantidade de exemplos de treinamento desta intenção e μ_{ET} a média de exemplos de treinamento por intenção do *chatbot*. Um valor positivo de BET indica que a intenção possui mais exemplos do que a média do assistente virtual, sugerindo a necessidade de reduzi-los para evitar o desbalanceamento. Por outro lado, quando negativa, a métrica aponta para uma quantidade inferior em relação à média, recomendando a inclusão de novos exemplos de treinamento para que o *chatbot* se alinhe em relação ao balanceamento.

- **Quantidade de intenções órfãs (QIO):** uma intenção órfã é aquela que não está relacionada a nenhuma resposta no fluxo conversacional do *chatbot*. Assim, a identificação de intenções nestas condições se apresenta como necessário, pois intenções órfãs podem resultar na falta de resposta por parte do assistente virtual, mesmo quando ele tem capacidade de identificar a necessidade do usuário e associá-la a alguma de suas intenções. Isso impacta negativamente na assertividade e na capacidade do *chatbot* de compreender e fornecer uma resposta apropriada aos usuários,

comprometendo a experiência do usuário e a eficácia geral do projeto. É neste contexto que esta métrica está inserida, sendo obtida através do somatório de todas as intenções órfãs existentes, conforme:

$$QIO = \sum_{i=1}^{n} I_{O_i}, \tag{4.2}$$

onde I_O se refere a uma intenção órfã, e n o número total de intenções nestas condições.

- **Quantidade de nós de diálogo com condições de entrada iguais (QID):** se diferentes nós de diálogo possuem as mesmas condições de entrada, o assistente virtual pode identificar corretamente a intenção do usuário, mas retorna uma resposta incorreta. Isso acontece porque, após identificar a intenção da mensagem, e conforme explicado na Seção 2.3.2, o motor de conversação segue sequencialmente a árvore de diálogos e seleciona o primeiro nó que satisfaz a condição de entrada, com base na intenção detectada. Esse cenário pode resultar em respostas erradas do *chatbot*, afetando negativamente a assertividade, a qualidade da interação e a satisfação do usuário. Assim, esta métrica é responsável por identificar e quantificar os nós de diálogo presentes nesta situação, através da equação

$$QID = \sum_{i=1}^{n} \sum_{j=1}^{n} \delta(CE_{D_i}, CE_{D_j}), com \quad i \neq j. \tag{4.3}$$

  Nesta equação, n representa o número total de nós de diálogo e CE_D significa a condição de entrada de um nó de diálogo; enquanto $\delta(CE_{D_i}, CE_{D_j})$ é uma função delta de Kronecker [80], que retorna 1 se os elementos CE_{D_i} e CE_{D_j} forem iguais, e 0 caso contrário.

- **Quantidade de nós de diálogo com condição de entrada sempre verdadeira (QDV):** esta métrica avalia a presença de nós de diálogo em um *chatbot* que possuem uma condição de entrada constantemente verdadeira. Essa situação ocorre geralmente devido a um erro de modelagem, na qual a condição de entrada do nó é fixada com o valor *"true"*, em vez de estar adequadamente associada a alguma intenção, como ilustrado na Figura 2.1. É importante evitar que existam nós de diálogo nessas condições, a fim de garantir respostas assertivas. Pois, quando um nó de diálogo possui essa condição permanente, ele será ativado sempre que o motor de conversação o examinar, independentemente da intenção do usuário detectada ou dos nós subsequentes na árvore de diálogos. Isso pode levar a respostas inadequadas e confusão para o usuário, prejudicando a eficácia do assistente virtual. Desta

forma, a QDV de um *chatbot* é dada por:

$$QDV = \sum_{i=1}^{n} D_{V_i}, \tag{4.4}$$

em que n é o número de nós de diálogo existentes e D_V é um nó de diálogo que possui condição de entrada sempre verdadeira.

- **Quantidade de nós de diálogo com condição de entrada sempre falsa (QDF):** esta métrica avalia a quantidade de nós de diálogo em um *chatbot* que nunca serão ativados durante uma conversa. Em outras palavras, ela visa identificar condições de entrada que sempre serão falsas. Isso pode acontecer quando a condição de entrada é fixada com o valor *"false"* ou quando é usada uma expressão lógica com o operador *"AND"* entre duas intenções. Para ilustrar isso, dada a Figura 2.1, imagine que a condição de entrada do "Nó de diálogo 1" fosse *"'Intenção A' AND 'Intenção N"'*. O resultado desta expressão sempre será falso, visto que uma mensagem de usuário não pode ser associada a duas intenções ao mesmo tempo. Além de impactar no desempenho do assistente virtual, esta situação pode causar confusão e dificultar a implementação de melhorias ou a adição de novos recursos. Portanto, evitar nós de diálogo com condição sempre falsa promove um *design* mais limpo e sustentável do *chatbot*. Assim, a QDF é obtida a partir da equação

$$QDF = \sum_{i=1}^{n} D_{F_i}, \tag{4.5}$$

onde n é o número de nós de diálogo existentes e D_F é um nó de diálogo que possui condição de entrada sempre falsa.

- **Quantidade de nós de diálogo sem texto de resposta (QDR):** trata-se de uma métrica que avalia a quantidade de nós de diálogo que não possuem respostas efetivas, ou seja, respostas com textos vazios. Essa medida desempenha um papel importante na avaliação da qualidade do *chatbot*, uma vez que nós de diálogo sem respostas efetivas comprometem a assertividade e a experiência do usuário. Ao analisar a quantidade de nós com respostas vazias, é possível identificar e corrigir lacunas no fluxo de conversa. Isso evita que o assistente virtual falhe em fornecer uma resposta ao usuário, mesmo tendo compreendido corretamente sua necessidade, o que poderia ser interpretado como um erro do assistente virtual. Consequentemente, essa métrica contribui para aprimorar a eficácia do *chatbot*, assegurando que todos os nós proporcionem respostas aos usuários e promovam uma interação mais

satisfatória. Neste contexto, a QDR é expressa por:

$$QDR = \sum_{i=1}^{n} D_{SR_i}, \tag{4.6}$$

na qual, o número de nós de diálogos é representado por n e o D_{SR} determina um nó de diálogo que não possui texto de resposta.

Ressalta-se ainda que, além da proposição destas métricas, o contexto no qual o *framework* DUBI está inserido implica em selecionar uma abordagem para viabilizar o cálculo da métrica SID. Conforme apresentado na Seção 2.2.5 deste documento, tal métrica pode ser calculada empregando a distância de cosseno entre vetores que representam os textos sob análise. Entretanto, existem diversas técnicas para transformar textos em representações vetoriais. Diante desta variedade de métodos de vetorização, foi realizado um experimento comparativo para identificar a técnica mais apropriada ao contexto específico deste trabalho, ou seja, vetorização de textos curtos que representam exemplos de treinamento de intenções do *chatbot*. Os resultados desse experimento, detalhados no Apêndice C, indicaram que a abordagem Word2Vec [81], empregando um modelo treinado com corpus em língua portuguesa e representando os textos em vetores de 600 dimensões [82], sobressaiu-se como a mais eficaz para o *framework* DUBI.

A partir da apresentação destas métricas, entende-se que o módulo DU abrange uma ampla gama de métricas observadas, satisfazendo duas das categorias mencionadas na Seção 2.2 deste documento. Essas categorias incluem métricas de qualidade de respostas, através da observação da legibilidade e da representatividade dos textos utilizados no treinamento do *chatbot*, e métricas relacionadas à qualidade do *design*, como as descritas na Tabela 4.1.

Além disso, é importante frisar que algumas dessas métricas podem ter diferentes valores recomendados de acordo com particularidades do projeto. Para isso, é necessário que uma parametrização seja possível de ser realizada durante a avaliação estática. É nesta linha que segue a próxima seção, informando quais dessas métricas são passíveis de configuração.

4.3.2 Parâmetros de configuração do módulo DU

Um aspecto importante na avaliação estática é a dependência do motor de conversação utilizado pelo assistente virtual, uma vez que a estrutura do assistente é um dos elementos avaliados. Cada motor de conversação (*e.g.* Serprobots Perguntas & Respostas, IBM *Watson Assistant* ou Google *DialogFlow*) possui sua própria forma de estruturar os *chatbots*, o que pode implicar em recomendar valores distintos para determinadas métricas.

Outras dependências que impactam na avaliação da estrutura são o perfil do público-alvo e o canal de comunicação pelo qual o *chatbot* será utilizado.

Para lidar com essa peculiaridade, o módulo *Design Understanding* (DU) disponibiliza a configuração de parâmetros específicos que podem ser ajustados conforme as características do assistente virtual em avaliação.

Neste cenário, dentre as métricas apresentadas na Tabela 4.1, destacam-se:

- QET e TET: podem ser configuradas segundo a recomendação do fabricante do motor de conversação;
- LET e LTR: podem ser configuradas conforme a habilidade de leitura esperada do público-alvo;
- RET e RTR: definidas conforme decisão de projeto, indicando se o desejado é ter um vocabulário em maior ou menor grau de representatividade, em relação aos termos mais comuns do idioma utilizado; e
- TTR: passível de configuração a depender do canal de comunicação utilizado para interagir com o *chatbot*.

Essa abordagem permite uma generalização eficaz da avaliação estática realizada neste módulo, independentemente das particularidades de cada projeto. Um exemplo prático deste processo de configuração de parâmetros é apresentado no Capítulo 5 do presente trabalho. Já a próxima seção descreverá o funcionamento geral do módulo DU.

4.3.3 Funcionamento do módulo DU

Em termos de fluxo de funcionamento, o módulo DU é inicialmente invocado recebendo como parâmetros de entrada o identificador único do *chatbot* a ser avaliado, bem como os valores de referência a serem considerados na avaliação, conforme detalhado na Seção 4.3.2. Este identificador é então utilizado para carregar a estrutura do assistente virtual correspondente por meio da plataforma Serprobots, que hospeda os *chatbots* e fornece uma *Application Programming Interface* (API) de comunicação para essa recuperação. Essa estrutura é representada por um arquivo *JavaScript Object Notation* (JSON), que contém todas as informações do assistente virtual, como intenções, nós de diálogo, fluxos de conversa e relacionamento entre os elementos.

Com base nas informações contidas no arquivo de estrutura do *chatbot*, o módulo DU realiza a avaliação estática daquele assistente virtual. Assim, as métricas especificadas na Tabela 4.1 são observadas e medidas, conforme detalhado na seção anterior deste capítulo.

Ao finalizar a avaliação, os resultados obtidos são formatados em outro arquivo JSON, representando o relatório de avaliação estática. Esse relatório inclui as métricas aferidas e

os indícios de melhoria identificados no *chatbot* avaliado, permitindo aos desenvolvedores melhor compreender as áreas que precisam ser aprimoradas. O Apêndice A detalha a estrutura do relatório resultante da avaliação realizada por este módulo.

Assim, o funcionamento do *Design Understanding* (DU) pode ser resumido pelos passos a seguir:

1. A aplicação cliente invoca o serviço de avaliação estática do módulo DU, especificando como parâmetros de configuração: o identificador do *chatbot* a ser avaliado, a escolaridade do público-alvo e o canal de comunicação utilizado para interagir com o assistente virtual.
2. O módulo DU processa a solicitação, carrega a estrutura do *chatbot* correspondente, através da plataforma Serprobots, e realiza a avaliação.
3. O retorno da avaliação é um arquivo JSON contendo as métricas identificadas e seus respectivos valores, permitindo que a aplicação cliente utilize essas informações para análise e aprimoramento do *chatbot*.

Dessa forma, compreende-se que uma parcela significativa do processo de avaliação de assistentes virtuais, conhecida como avaliação estática, é adequadamente abordada pelo módulo DU. No entanto, é importante ressaltar que a avaliação estática por si só não é suficiente para uma análise completa, sendo essencial complementar seus resultados com conclusões obtidas por meio da avaliação interativa dos *chatbots*. É nesse contexto que o módulo BI, detalhado na próxima seção, desempenha um papel fundamental na proposta do *framework* DUBI.

4.4 O módulo BI — *chatBot Intelligence*

Como discutido anteriormente, a avaliação da estrutura e do conteúdo de treinamento de um assistente virtual pode antecipar a identificação de potenciais problemas que ocorrerão durante o uso do *chatbot*. Apesar disto, a avaliação estática não consegue medir o desempenho do sistema em uso propriamente dito. Portanto, embora relevante e necessária, ela não é suficiente para aferir de forma abrangente a qualidade de um assistente virtual.

Para isso, é necessário também analisar métricas e características que só podem ser observadas durante a interação com o *chatbot*. É nesse sentido que o módulo *chatBot Intelligence* (BI) se direciona, ou seja, observar e medir indicadores que determinem a qualidade do desempenho em responder às perguntas recebidas.

Esta abordagem já foi utilizada em estudos anteriores, como mencionado na Seção 3.5.2 deste documento. No entanto, estes estudos encontraram dificuldades em automatizar o

processo por completo. Por sua vez, o módulo BI propõe um avanço neste aspecto, eliminando a necessidade de intervenção humana tanto na criação dos casos de teste quanto na avaliação dos resultados.

Para alcançar esse objetivo, a estratégia proposta é utilizar o próprio conteúdo de treinamento do *chatbot*, como intenções, exemplos de treinamento e textos de respostas, para gerar mensagens simuladas, enviá-las ao assistente virtual e avaliar suas respostas. Essa estratégia e o funcionamento do módulo são detalhados na próxima seção.

4.4.1 Funcionamento do módulo BI

O módulo *chatBot Intelligence* (BI), assim como o módulo DU, também recorre à API de comunicação da plataforma Serprobots para se comunicar com o *chatbot* a ser avaliado. Desta forma, consegue tanto acessar o conteúdo de treinamento dos *chatbots* quanto interagir com eles para avaliar sua eficácia em diferentes cenários. Conforme ilustrado na Figura 4.2, este módulo é composto por 3 principais componentes: gerador de mensagens simuladas, simulador de usuário e avaliador das respostas. Nesta seção, estes componentes e seus respectivos funcionamentos são detalhados.

Etapa 1: Gerador de mensagens simuladas

O processo fundamental para o módulo BI é a geração de mensagens simuladas e sua organização em casos de testes. O objetivo desta etapa é viabilizar que estas mensagens possam ser enviadas ao assistente virtual, imitando o comportamento de um usuário real.

A geração destas mensagens ocorre com base no conteúdo modelado do *chatbot*, especificamente os exemplos de treinamento das intenções cadastradas, e no emprego de um modelo de linguagem generativo, o qual recorre a técnicas de aprendizado de máquina e redes neurais para criar novos conteúdos a partir de contextos iniciais e instruções específicas [34]. Este processo, ilustrado na Figura 4.3, segue o seguinte fluxo:

1. **Definição dos dados de contexto:** com base no identificador do *chatbot* em avaliação, os textos dos exemplos de treinamento de suas intenções são carregados e utilizados como dados de contexto para gerar as mensagens. Conforme pode ser observado na coluna "Textos originais" da figura, cada intenção possui um conjunto distinto de textos representando seus exemplos. Assim, para cada intenção um contexto diferente é criado.

2. **Montagem da instrução para geração das mensagens:** conforme o tipo de teste que se deseja fazer, diferentes tipos de mensagens podem ser solicitados ao modelo de linguagem. Estes tipos são detalhados na Seção 4.4.2, mas, em resumo,

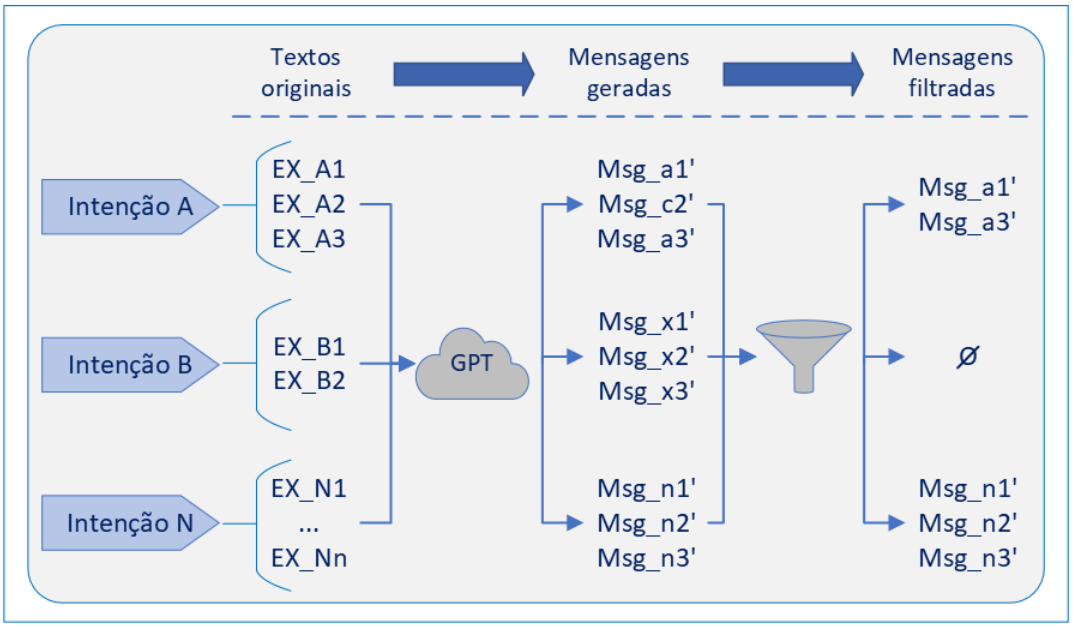


Figura 4.3: Fluxo de geração das mensagens de teste.

Fonte: autoria própria.

podem resultar duas possíveis instruções (gerar mensagens semanticamente similares ou diferentes), segundo os dados de contexto. São essas instruções que definem o que deve ser gerado para simular um usuário. Um exemplo prático dessas instruções é apresentado na Seção 5.4.

3. **Geração das mensagens simuladas:** uma vez definidos o contexto e a instrução, o modelo de linguagem é requisitado para realizar a tarefa de gerar as mensagens simuladas. Na Figura 4.3, o resultado desta requisição é representado pela coluna "Mensagens geradas". Na execução do módulo BI, o objetivo destas mensagens é simular interações que um usuário real poderia fazer com o *chatbot*. Destaca-se que, para o escopo deste trabalho, embora qualquer modelo de linguagem pudesse ser utilizado, decidiu-se por utilizar o *Generative Pre-Training* (GPT) [49], na versão 3.5 turbo, tendo em vista que o acesso a ele é fornecido pelo SERPRO.

4. **Validação das mensagens geradas:** no passo anterior, não há garantias de que o modelo de linguagem interpretará e executará sem erros a instrução fornecida. Assim, ao receber as mensagens geradas, o componente "gerador de mensagens simuladas" valida se elas estão conforme os requisitos da instrução utilizada. Ou seja, se a instrução foi "gerar mensagens semanticamente similares", o resultado deve atender a este critério, o mesmo se aplica para o caso oposto, quando se deseja criar mensagens semanticamente diferentes. Para isso, é calculada a similaridade entre os textos (das mensagens geradas e dos exemplos de treinamento) e, segundo

o objetivo da instrução, alguma mensagem pode ser descartada, conforme definido no próximo item.

5. **Filtragem das mensagens geradas:** com base em um limiar de similaridade, que pode ser configurado como parâmetro do projeto, e na validação realizada no item anterior, as mensagens geradas serão filtradas. Assim, como resultado, apenas aquelas que atenderem as requisitos seguem em frente no processo do módulo BI. Na Figura 4.3 esta tarefa é exemplificada. Perceba que as mensagens geradas para a "Intenção A" foram "Msg_a1", "Msg_c2" e "Msg_a3", para a "Intenção B" se obteve "Msg_x1", "Msg_x2" e "Msg_x3", enquanto para a "Intenção N" se gerou "Msg_n1", "Msg_n2" e "Msg_n3". Agora, considere que a instrução utilizada foi a de gerar mensagens similares. Assim, poder-se-ia chegar a conclusão que a "Msg_c2" não é semanticamente similar aos textos originais da "Intenção A"; já para a "Intenção B" todas as mensagens geradas estão em desacordo com o requisito; enquanto a "Intenção N" apresentou um desempenho superior, recebendo todas as mensagens geradas semanticamente similares aos seus exemplos de treinamento.

6. **Definição das mensagens a serem utilizadas:** após realizar a filtragem das mensagens geradas, cada intenção possuirá uma lista com as mensagens que de fato são utilizadas no caso de teste. O tamanho desta lista poderá variar de zero até m, sendo esta variável a quantidade máxima de mensagens geradas pelo modelo de linguagem para aquela intenção. Essa finalização do processo é representada pela coluna "Mensagens filtradas", na Figura 4.3. Neste exemplo, enquanto as intenções "A" e "N" seriam alvo do teste realizado, a "Intenção B" não seria coberta pelo caso de teste, visto que não haveria mensagens associadas a ela que seriam enviadas ao *chatbot*.

Assim, após gerar as mensagens simuladas para todas as intenções, e realizar a filtragem daquelas adequadas ao cenário de teste, o componente organiza as mensagens resultantes no artefato "caso de teste". Este artefato, detalhado na Seção 4.4.2 deste documento, é utilizado pelo componente "simulador de usuário" como base para simulação de interações, conforme explicado na próxima seção.

Etapa 2: Simulador de usuário

O componente simulador de usuário tem como função simular a interação entre usuários e o *chatbot*, sendo responsável pela execução dos casos de teste e pela coleta das respostas obtidas durante essas interações simuladas.

Através da API de conversação disponibilizada pela plataforma Serprobots, o simulador se conecta ao assistente virtual em avaliação para iniciar uma conversa e executar um

caso de teste. Este artefato, conforme detalhado na Seção 4.4.2, consiste em uma lista de unidades de teste, que representam as interações a serem simuladas.

Assim, após a conversa ser iniciada, o simulador de usuário executa sequencialmente todas as unidades de teste presentes no caso de teste em questão. A cada requisição realizada, o texto da resposta obtida é armazenado na unidade de teste correspondente, juntamente com o tempo decorrido entre o envio da mensagem e o recebimento da resposta.

Uma vez finalizado o processo, a conversa simulada é encerrada e o artefato de caso de teste é atualizado com as informações recebidas na execução de cada unidade de teste para posterior análise.

Esta abordagem sistemática permite o cálculo posterior das métricas de desempenho, qualidade de diálogo e cobertura de testes, especificadas na Seção 4.4.3. A avaliação destas métricas é escopo do terceiro componente do módulo BI, o "avaliador de respostas", conforme detalhado na próxima seção.

Etapa 3: Avaliador das respostas

O componente avaliador das respostas, embora seja o último a ser executado pelo módulo *chatBot Intelligence* (BI), tem importância equivalente aos componentes anteriores. Sua função principal é comparar as respostas obtidas do *chatbot*, durante interações simuladas, com as respostas esperadas e calcular as métricas observadas por esta parte do *framework* DUBI. Essa comparação permite uma avaliação precisa do desempenho do assistente virtual.

O objetivo desta análise é diagnosticar, de maneira objetiva e automática, o desempenho e a qualidade do diálogo do assistente virtual. Isso é alcançado através da observação e aferição de métricas como acurácia, tempo médio de resposta, taxas de compreensão e taxas de consistência das respostas, dentre outras, conforme detalhado na Seção 4.4.3 deste capítulo.

Além de fornecer uma avaliação quantitativa da assertividade do *chatbot*, este componente também identifica declarativamente as mensagens que foram malsucedidas, bem como o percentual de cobertura alcançado pelo teste. Essas informações são então consolidadas e estruturadas no formato JSON, compondo assim o relatório com o resultado da avaliação interativa do *framework* DUBI. A estrutura deste relatório de avaliação do módulo BI é detalhada no Apêndice B.

Após finalizada a avaliação interativa, os resultados obtidos complementam o relatório final de avaliação do *chatbot*, que, nesse ponto, já contém os resultados da análise estática, conforme ilustrado na Figura 4.2 deste capítulo.

Como pôde ser observado, o funcionamento do módulo *chatBot Intelligence* (BI) se baseia na utilização de casos de testes. Estes artefatos são criados, alterados e consultados automaticamente no processo de avaliação interativa, servindo como base para tal. Desta forma, a próxima seção descreve como estes artefatos são estruturados.

4.4.2 Casos de testes gerados pelo módulo BI

Casos de teste são artefatos utilizados no desenvolvimento de *software* para verificar e validar o comportamento de um sistema. Eles descrevem cenários específicos nos quais o *software* é testado, permitindo identificar possíveis falhas, validar se os requisitos estão sendo atendidos e avaliar a qualidade do sistema [83].

No contexto de *chatbots*, essa perspectiva não difere, pois estes artefatos podem ser utilizados para organizar cenários que se deseja validar por meio da interação com o assistente virtual. Em outras palavras, pelo uso de casos de testes, é possível avaliar se o *chatbot* está compreendendo corretamente as perguntas e fornecendo respostas adequadas.

Neste cenário, o *framework* DUBI utiliza o componente "gerador de mensagens simuladas" para gerar automaticamente os casos de teste com base no conteúdo de treinamento do *chatbot*, conforme explicado na Seção 4.4.1. Estes artefatos são compostos por um conjunto de mensagens simuladas, cada uma associada a uma intenção específica do *chatbot* e a uma resposta esperada. De modo que, ao ser executado, o módulo *chatBot Intelligence* (BI) consegue identificar as interações onde o assistente virtual se comportou como o esperado e as que não.

Para testar uma variedade de cenários possíveis, e visando se aproximar do que seriam interações entre usuários reais e o assistente virtual, este trabalho propõe a categorização das mensagens utilizadas nos casos de testes em três tipos, conforme a seguir:

- **Mensagem — tipo 1:** mensagens que fazem parte do escopo de treinamento do *chatbot* e sem erros de grafia. Essas mensagens estão no domínio e contexto de aprendizado do assistente virtual, representando perguntas ou comandos corretos que os usuários podem enviar. O objetivo deste tipo de mensagem é avaliar o desempenho do *chatbot* em sua capacidade de identificar corretamente as intenções dos usuários e responder adequadamente.
- **Mensagem — tipo 2:** são mensagens que também estão no escopo do assistente virtual, entretanto, geradas propositalmente com erros de grafia. Essas mensagens simulam entradas com erros ortográficos ou gramaticais, desafiando o *chatbot* a lidar com essas situações e fornecer respostas corretas.
- **Mensagem — tipo 3:** geradas com textos que estão fora do escopo de treinamento do assistente virtual, podendo apresentar ou não erros de grafia. Essas mensagens

têm por objetivo avaliar a capacidade do *chatbot* de reconhecer quando não pode fornecer uma resposta ao usuário, por não conhecer tal assunto, mas consegue retornar com uma mensagem de *fallback*.

Esta categorização permite criar cenários diversificados de testes e, consequentemente, avaliar o desempenho do assistente virtual em situações mais próximas daquelas que ele se deparará em produção, como mensagens corretas no escopo, com erros de grafia ou fora do escopo. Dessa forma, entende-se que a classificação dos tipos de mensagens nos casos de teste contribui na aferição da qualidade e confiabilidade do *chatbot*, viabilizando uma cobertura mais abrangente dos cenários de teste.

Sendo assim, os casos de testes gerados pelo módulo BI são representados em formato JSON, visando uma estrutura consistente para facilitar a sua interpretação e execução. A Tabela 4.2 apresenta a proposta de estrutura de um caso de teste, o qual é composto por um identificador único, um título, uma descrição e uma lista de unidades de teste. Cada unidade de teste é especificada por um identificador único, o tipo de mensagem testada e o texto gerado para simular a interação do usuário, com as respectivas características definidas pelo tipo discriminado. Além disso, para viabilizar a avaliação da interação, também faz parte da unidade de teste a intenção do *chatbot* a qual ela está vinculada, as respostas esperada e obtidas, bem como o tempo gasto na requisição de teste.

Tabela 4.2: Descrição das seções e atributos do artefato caso de teste.

Seção	**Atributo**	**Tipo**	**Descrição**
—	id	Alfanumérico	Identificação do caso de teste.
	title	Alfanumérico	O título do caso de teste.
	desc	Alfanumérico	A descrição do caso de teste.
test_units	—	Lista	As unidades de teste do artefato, cada uma contendo:
	id	Alfanumérico	Identificação da unidade de teste.
	type	Inteiro	O tipo de mensagem simulada.
	binding_intent	Alfanumérico	A identificação da intenção do *chatbot* testada.
	simulated_message	Alfanumérico	A mensagem simulada utilizada no teste.
	expected_answer	Alfanumérico	A resposta esperada para esta interação.
	received_answer	Alfanumérico	A resposta recebida do *chatbot*.
	response_time	Real	O tempo de resposta, em segundos, desta interação.

Uma vez criados e executados, os casos de teste viabilizam a aferição de uma série de métricas relacionadas ao desempenho, à qualidade do diálogo do *chatbot* e à cobertura do teste realizado, conforme apresentado na Seção 4.4.3.

4.4.3 Métricas aferidas no módulo BI

Conforme explicado anteriormente, a avaliação interativa possibilita medir o desempenho de um assistente virtual com base nas respostas fornecidas por ele em um diálogo. Essa

abordagem é a mais utilizada pelos trabalhos disponíveis na literatura, como exemplificado pelos estudos discutidos na Seção 3.5.2. Esses estudos contribuíram de forma relevante com o tema ao proporem métricas objetivas, capazes de aferir o desempenho do *chatbot* e a qualidade do diálogo conduzido por ele. Estas métricas foram apresentadas nas Seções 2.2.1 e 2.2.4, respectivamente, sendo as seguintes: acurácia, precisão, *recall*, *F1-score*, tempo médio de resposta (tm_R), taxa de compreensão (T_c), taxa de consistência das respostas (T_{CR}) e relevância das respostas.

Para este trabalho, dentre estas métricas, entende-se que a "relevância das respostas" não é adequada devido a sua natureza subjetiva e por ser, frequentemente, dependente da apreciação individual dos usuários, dificultando assim o estabelecimento de critérios objetivos para uma avaliação automatizada. Mesmo em tentativas anteriores de automatização, como mencionado em [13], a subjetividade persiste, com os autores definindo um método de avaliação sem embasamento científico. Por estas razões, optou-se por não utilizar esta métrica durante a avaliação do módulo BI.

Por outro lado, verificou-se que as publicações atuais na literatura não oferecem métodos de analisar as situações nas quais o assistente virtual enfrenta incertezas ao interpretar as entradas dos usuários. Neste contexto, o *framework* DUBI introduz uma métrica denominada "taxa de ambiguidade" para abordar essa lacuna. Esta métrica é especialmente relevante para avaliar a habilidade do *chatbot* em distinguir entre intenções semelhantes. Ela quantifica as situações em que o assistente virtual tem dificuldade em selecionar a intenção mais apropriada para uma mensagem recebida, especialmente quando o conteúdo da mensagem está próximo de mais de uma intenção possível. Isso resulta na apresentação de várias intenções candidatas, todas com níveis de confiança relativamente baixos. Em face a tal ambiguidade, o *chatbot* adota uma abordagem de desambiguação, tipicamente exibindo ao usuário um conjunto das possíveis intenções detectadas e requisitando uma clarificação adicional, usualmente por meio de uma indagação como *"você quis dizer?"*. Portanto, a taxa de ambiguidade reflete a recorrência dessa situação, e a sua diminuição serve como um sinalizador de aprimoramento do sistema de classificação de intenções do assistente virtual. Ou seja, uma taxa de ambiguidade baixa sugere que o *chatbot* possui uma boa capacidade de discernir as necessidades dos usuários a partir dos dados contidos em sua base de conhecimento. Desta forma, esta métrica pode ser calculada através da equação:

$$\psi = \frac{N_d}{N_t}, \tag{4.7}$$

onde N_d é a quantidade de interações que ativaram o recurso de desambiguação e N_t é o total de mensagens de uma seção de conversa.

Além dessas métricas, decidiu-se por incluir também os percentuais de cobertura do teste, especificamente em relação às intenções e nós de diálogo testados. Esta decisão

se baseou na necessidade de indicar a extensão e abrangência dos testes realizados no sistema. Pois, ao medir essas coberturas, possibilita-se compreender quais intenções e nós de diálogo foram de fato testados, oferecendo uma visão clara dos pontos abordados durante o processo de avaliação. Assim sendo, tem-se:

- **Cobertura de intenções testadas:** mede a proporção de intenções testadas em relação ao total de intenções definidas para o *chatbot*. Esta métrica é calculada pela equação:

$$\textit{Cobertura de intenções testadas} = \frac{\sum_{i=1}^{m} I_{T_i}}{\sum_{j=1}^{n} I_j}, \tag{4.8}$$

  onde m é o número de intenções testadas, n é o total de intenções do *chatbot*, I_T é uma intenção testada e I_j é a j-ésima intenção do assistente virtual. Nesta equação, m sempre será menor ou igual a n.

- **Cobertura de nós de diálogo testados:** é a proporção de nós de diálogo testados em relação ao total de nós existentes no fluxo de conversação do *chatbot*. Sendo a fórmula para calcular esta métrica a seguinte:

$$\textit{Cobertura de nós de diálogos testados} = \frac{\sum_{i=1}^{m} D_{T_i}}{\sum_{j=1}^{n} D_j}, \tag{4.9}$$

  na qual, m é o número de nós de diálogo testados, n é o total de nós de diálogos do assistente virtual, D_T é um nó de diálogo testado e D representa um nó de diálogo qualquer do *chatbot*. Aqui, também se repete a condição de m sempre ser menor ou igual a n.

Isto posto, todas as métricas contempladas no módulo BI são sumarizadas na Tabela 4.3, de modo a permitir uma visão geral dos respectivos conceitos.

Com base no exposto até o momento, entende-se que a abordagem do *framework* DUBI demonstra vantagens significativas em comparação aos estudos anteriores, especialmente por possibilitar avaliações estáticas e interativas de maneira automatizada, superando assim limitações dos métodos anteriores. Neste sentido, os benefícios e diferenciais desta proposta de trabalho serão abordados na próxima seção deste documento, fornecendo uma visão clara do valor agregado que o DUBI aspira ofertar ao processo de avaliação de *chatbots*.

4.5 Benefícios e diferenciais do DUBI

No capítulo anterior deste documento, uma discussão sobre os avanços e as limitações dos trabalhos atuais relacionados à avaliação de *chatbots* foi apresentada na Seção 3.5.3.

Tabela 4.3: Métricas observadas pelo módulo BI — *chatBot Intelligence*.

Métrica	**Descrição**
Acurácia	Utilizada para indicar a taxa de acerto geral.
Precisão	Útil para medir a capacidade do modelo de evitar falsos positivos.
Recall	Mede a capacidade do modelo de identificar todas as instâncias positivas.
F1-Score	É a média harmônica entre a precisão e o *recall*, representando uma medida geral de desempenho de um modelo.
Tempo médio de resposta (tm_R)	Indica o tempo necessário para o *chatbot* processar uma mensagem e gerar uma resposta.
Tx. de *fallback* (T_f)	Indica o percentual de mensagens que o *chatbot* não soube responder.
Tx. de ambiguidade (ψ)	Calcula o percentual de interações que geraram dúvidas no chatbot ao tentar selecionar a melhor resposta.
Tx. de consistência das respostas (T_{CR})	Mede a capacidade do *chatbot* enviar a mesma resposta ao receber mensagens semanticamente parecidas.
Tx. de compreensão (T_c)	Avalia a capacidade do *chatbot* de entender mensagens com erros ortográficos e/ou gramaticais.
Cobertura de intenções testadas	Calcula a proporção de intenções impactadas em um teste.
Cobertura de nós de diálogo testados	Indica a proporção de nós de diálogo testados.

Nessa análise, foram identificadas duas lacunas principais: a avaliação ser restrita a uma das abordagens, estática ou interativa, e os processos não serem completamente automatizados.

O *framework* DUBI, por sua vez, destaca-se por adotar uma abordagem híbrida, implementando tanto a avaliação estática quanto a interativa, de modo que seus resultados se complementem e forneçam uma ampla avaliação de qualidade dos *chatbots*. Essa combinação permite a identificação de problemas que seriam difíceis de serem detectados durante uma interação comum.

A automação de todo o processo de avaliação é outro diferencial desta proposta. Tanto o módulo *Design Understanding* (DU) quanto o *chatBot Intelligence* (BI) automatizarão todas as tarefas, desde a obtenção dos dados até a execução e avaliação de resultados dos

casos de testes. Esta característica permitirá que a tarefa de testar e avaliar assistentes virtuais seja repetível, rápida e de baixo custo.

Desta forma, acredita-se que estes diferenciais possibilitam o benefício de avaliar os *chatbots* antes de suas implantações, colaborando assim com a detecção prévia de erros, a tempo de corrigi-los antes dos usuários interagirem com o assistente virtual. Esta particularidade tende a aumentar a satisfação do usuário com o sistema, visto que menos erros serão por ele percebidos.

Com base nas informações apresentadas, pode-se concluir que o *framework* DUBI é a proposta que melhor se aproxima àquela considerada a mais abrangente para avaliação de *chatbots*, indo além dos estudos abordados na Seção 3.5. Ainda neste sentido, é importante frisar que essa proposta referida como "mais abrangente" não representa um trabalho existente na literatura, mas sim algo que seria desejável, agregando aspectos de outros estudos relacionados no estado da arte, conforme discutido na Seção 3.5.3. A fim de ilustrar essa afirmação, a Tabela 4.4 realiza uma comparação entre as características da proposta dita como mais abrangente e as do *framework* DUBI.

Tabela 4.4: *Framework* DUBI *vs.* proposta mais abrangente.

Atributo observado	Propostas	
	*	**DUBI**
Método de avaliação	Automático	Automático
Tipo de interação	Estática e interativa	Estática e interativa
Métricas de desempenho	✓	✓
Métricas de satisfação do usuário	✓	-
Métricas de qualidade das respostas	✓	✓
Métricas de qualidade do diálogo	✓	✓
Métricas de qualidade de *design*	✓	✓
Domínio de conhecimento	DF ou DA	DF
Geração de resposta	GE ou RI	RI
Entendimento da necessidade	Baseado em intenções	Baseado em intenções

Onde: (*) = proposta mais abrangente, (✓) = usa a métrica, (-) = não utiliza a métrica, DF = domínio fechado, DA = domínio aberto, GE = generativo e RI = recuperação de informações.

Conforme apresentado nesta tabela, o *framework* DUBI difere da abordagem mais abrangente em apenas três atributos: métricas de satisfação do usuário, domínio de conhecimento e geração de resposta. A restrição do escopo deste trabalho, apresentada no Capítulo 1, explica a falta de suporte para *chatbots* de domínio aberto e generativos.

Já a ausência das métricas de satisfação do usuário é justificada pelo fato de que elas são obtidas a partir das opiniões dos usuários após o uso do assistente virtual. Isso está em desacordo com o objetivo do *framework* DUBI, o qual é avaliar prematuramente os *chatbots*, antes mesmo de serem disponibilizados para os usuários finais. Além disso,

a plataforma Serprobots já possui um componente para a avaliação da satisfação dos usuários, conforme apresentado na Seção 2.3.1, o que permite ao SERPRO obter essa métrica.

4.6 Resumo do capítulo

O presente capítulo apresentou detalhadamente o *framework* DUBI, desenvolvido para viabilizar a avaliação automatizada de *chatbots*. Inicialmente, foi fornecida uma visão geral da solução, destacando seu funcionamento e principais características. Em seguida, foram explorados em detalhes a arquitetura, os módulos e os componentes do *framework*, proporcionando um entendimento abrangente de sua estrutura. Além disso, foram apresentadas as métricas utilizadas para avaliar a qualidade dos assistentes virtuais, bem como os diferenciais do DUBI.

Em suma, a proposta se destaca por utilizar uma abordagem híbrida, envolvendo a avaliação estática e interativa de *chatbots*, bem como por automatizar todo o processo de avaliação. Vale ressaltar que, apesar de ter sido proposto para ser parte integrante da plataforma Serprobots do SERPRO, o *framework* DUBI possui capacidade de generalização, podendo ser aplicado em *chatbots* construídos em diferentes plataformas. No próximo capítulo, serão apresentados o detalhamento e as informações sobre a preparação e configuração experimental do procedimento realizado para validar a viabilidade técnica e conceitual da solução.

Capítulo 5

Ambiente experimental

Este capítulo apresenta o experimento realizado com objetivo de validar o conceito proposto nesta pesquisa, bem como a viabilidade técnica do *framework* DUBI. O experimento abrangeu o escopo completo deste trabalho, permitindo a avaliação de todos os aspectos funcionais do DUBI. Aqui, serão detalhados os procedimentos experimentais adotados, o ambiente utilizado e as características dos assistentes virtuais utilizados.

Desta forma, o texto foi assim estruturado: a Seção 5.1 apresenta uma visão geral do experimento realizado; na sequência, na Seção 5.2, a metodologia adotada é detalhada; já a Seção 5.3 caracteriza os *chatbots* utilizados no experimento, informando suas características de negócio e estruturais; o ambiente e os parâmetros de configuração utilizados no procedimento são apresentados na Seção 5.4; por fim, o capítulo é finalizado com um breve resumo, apresentado pela Seção 5.5.

5.1 Visão geral do experimento

Conforme detalhado no capítulo anterior, o *framework* DUBI se propõe a viabilizar uma avaliação abrangente e automática de *chatbots*, empregando para isso as avaliações estática e interativa. Estas, respectivamente, focam em avaliar a estrutura e o conteúdo de treinamento do assistente virtual, bem como as respostas obtidas durante a interação simulada com o assistente virtual.

Neste contexto, um experimento foi realizado com o intuito de validar a viabilidade técnica e conceitual do *framework* DUBI. Para isso, o procedimento foi planejado para abranger o escopo completo da solução proposta por este trabalho, assegurando que todos os aspectos funcionais do DUBI fossem testados e avaliados. A execução deste teste prático envolveu a aplicação dos módulos *Design Understanding* (DU) e *chatBot Intelligence* (BI) em três *chatbots* reais, selecionados a partir de critérios específicos que os qualificavam como representativos para o estudo, conforme detalhado na Seção 5.3. Es-

tes assistentes virtuais já estavam em funcionamento e não foram criados especificamente para este procedimento, o que possibilitou uma avaliação em um cenário realista e com dados concretos.

Entende-se que a relevância deste experimento reside na possibilidade de verificar como as orientações fornecidas pela avaliação estática, conduzida pelo módulo DU, podem influenciar positivamente o desempenho dos *chatbots* em interações reais, mensuradas pelo módulo BI. Esta abordagem ajuda a compreender a interdependência entre a modelagem dos assistentes virtuais e a sua eficiência durante as interações com seus os usuários.

Isto posto, a experimentação visou reproduzir os passos apresentados na Seção 4.2 do capítulo anterior, para evidenciar a aplicabilidade do *framework* DUBI em situações práticas, avaliando a sua capacidade em diagnosticar e sugerir melhorias, bem como aferir os impactos das mudanças. A metodologia adotada para planejar e executar este experimento é detalhada na próxima seção.

5.2 Metodologia adotada para o experimento

O experimento foi planejado para testar o pressuposto de que os aprimoramentos estruturais sugeridos pelo DUBI resultam em melhorias no desempenho dos *chatbots*. Para este contexto, e conforme detalhamento da próxima seção deste capítulo, um conjunto de assistentes virtuais foram selecionados para serem objetos de estudo do experimento. Sendo assim, esta seção tem como foco esclarecer o processo pelo qual o experimento foi conduzido, detalhando as etapas da metodologia adotada.

De forma geral, conforme ilustrado na Figura 5.1, o experimento consistiu em avaliar cada um dos *chatbots*, tanto em termos estáticos quanto interativos, por quatro rodadas de avaliações. Entre as rodadas de avaliação, uma nova versão do respectivo assistente virtual era gerada, com base nas sugestões do módulo DU do *framework*.

Destaca-se que, em cada uma das rodadas de avaliação, o módulo BI foi executado por 10 vezes. Ao final das repetições, foram coletadas as métricas de desempenho do *chatbot*, não somente os valores absolutos, mas também os valores médios, desvios padrão e intervalos de confiança. O objetivo destas repetições foi mitigar o efeito de variabilidade, na etapa de geração das mensagens de testes, detalhada na Seção 4.4.1, inerente à utilização dos grandes modelos de linguagem.

Inicialmente, cada *chatbot* foi submetido a uma avaliação na sua versão original, marcando a primeira das quatro rodadas. Durante esta fase, conhecida como "Rodada 1", o módulo *Design Understanding* foi aplicado para gerar o relatório de avaliação estática, que incluiu métricas e sugestões de melhorias na estrutura do assistente virtual. Em seguida, o módulo *chatBot Intelligence* foi empregado para realizar a avaliação interativa,

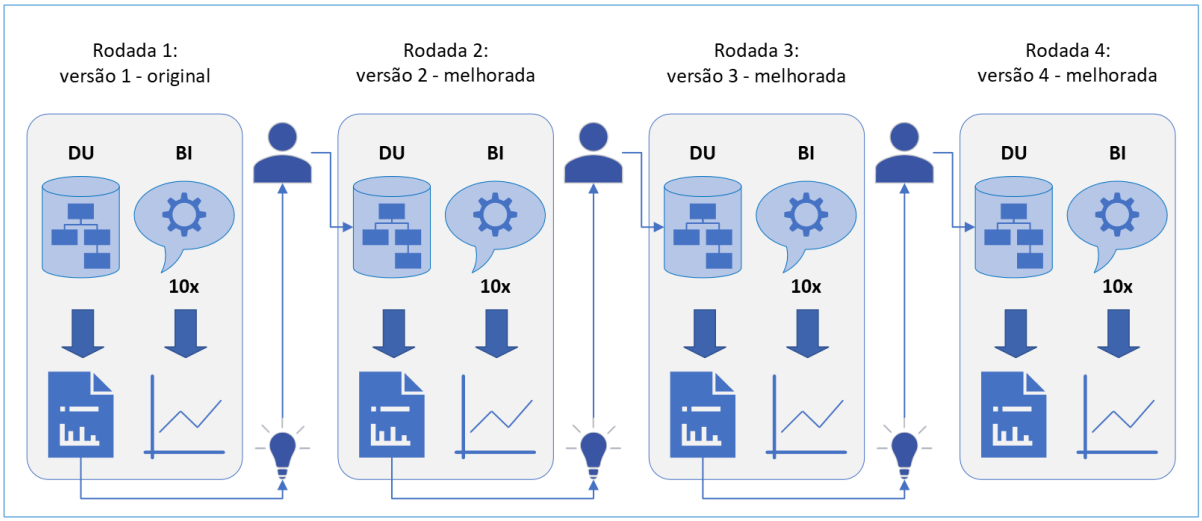


Figura 5.1: Fluxo do experimento de validação.

Fonte: autoria própria.

proporcionando uma linha base comparativa para o desempenho do *chatbot* no decorrer do experimento.

Após esta análise inicial, as recomendações de melhorias indicadas no relatório do módulo DU foram implementadas por um especialista humano, resultando em uma nova versão do *chatbot*. Esta versão atualizada foi então avaliada na "Rodada 2", seguindo o mesmo procedimento da primeira rodada, porém com o foco na versão aprimorada do assistente virtual. Este ciclo de avaliação e melhoria foi repetido mais duas vezes, gerando as versões subsequentes do assistente virtual e as respectivas "Rodada 3" e "Rodada 4", conforme apresentado na Figura 5.1.

O experimento foi limitado a quadro rodadas de avaliação para cada *chatbot* para assegurar condições de comparações semelhantes. Além disso, o intuito não era alcançar a versão ótima dos *chatbots*, mas sim avaliar a validade das melhorias propostas pelo *framework*.

Na próxima seção deste capítulo, os assistentes virtuais utilizados no experimento são apresentados, oferecendo um contexto mais aprofundado sobre as características deles.

5.3 Caracterização dos *chatbots* objetos de estudo

Para simular um cenário que se aproximasse da aplicação real do *framework* DUBI, este trabalho utilizou *chatbots* reais e pré-existentes, os quais não foram especificamente desenvolvidos para esta pesquisa. Destaca-se ainda que, pelo mesmo motivo, não foram introduzidas ou alteradas propositalmente qualquer característica dos assistentes virtuais

para testar algum aspecto funcional do *framework*. Visando facilitar sua identificação ao longo do texto, eles foram designados como "Bot-1", "Bot-2" e "Bot-3".

Os assistentes virtuais selecionados são baseados em intenções e em recuperação de informação, operando sob o formato de *single-turn* [84] para responder a perguntas. Como motor de conversação, utilizam o IBM watsonx Assistant [14] e suas características específicas, incluindo detalhes sobre o foco de negócio, estágio do projeto, perfil do público-alvo e da equipe de desenvolvimento, são apresentadas na Tabela 5.1.

O "Bot-1", com desenvolvimento finalizado e em produção, atende a consultas sobre serviços municipais, tendo sido desenvolvido por um time experiente em projetar assistentes virtuais. O "Bot-2", foi construído para ser utilizado em um evento de treinamento, apresentando uma estrutura mais simplificada e também tendo sido construído por uma equipe com experiência na tecnologia. Por fim, "Bot-3", ainda em fase de desenvolvimento, está sendo desenvolvido para atender a usuários de um serviço público federal, sua equipe não tem larga experiência neste tipo de projeto e o público-alvo tem escolaridade esperada de ensino médio completo.

Tabela 5.1: Características dos *chatbots*.

Característica	**Bot-1**				**Bot-2**				**Bot-3**			
Globais												
Negócio	Serviços municipais				Educacional				Atendimento			
Estágio atual	Finalizado				Finalizado				Desenvolvimento			
Equipe	Experiente				Experiente				Inexperiente			
Público-alvo	Ensino fundamental				Ensino superior				Ensino médio			
Por rodada do experimento												
	R1	**R2**	**R3**	**R4**	**R1**	**R2**	**R3**	**R4**	**R1**	**R2**	**R3**	**R4**
# Intenções	50	50	50	50	27	27	27	27	80	77	77	77
# Exemplos	570	672	658	651	238	277	277	276	1091	1224	1213	1222
# Diálogos	52	52	52	52	30	30	30	30	186	184	184	184

Em relação à estrutura, durante o experimento os "Bot-1" e "Bot-2" se mantiveram estáveis com 50 e 27 intenções respectivamente, sem necessidade de alteração nesses quantitativos durante o procedimento. Conforme será detalhado no Capítulo 6, o módulo DU não identificou deficiências no relacionamento entre as intenções e os demais elementos dos *chatbots*. Ambos também conservaram o número de nós de diálogo, embora tenham passado por ajustes nos conteúdos das respostas. Assim, as modificações nos "Bot-1" e "Bot-2", como ilustrado na Tabela 5.1, centraram-se na otimização dos exemplos de treinamento. Por outro lado, o "Bot-3" seguiu um percurso distinto, com redução nas quantidades de intenções e nós de diálogo devido a problemas como intenções órfãs e diálogos redundantes, identificados pela avaliação estática na primeira rodada de testes

(R1). Este *chatbot* também sofreu ajustes nos exemplos de treinamento, alinhando-se às mudanças estruturais necessárias.

De maneira geral, a variação na quantidade de elementos dos *chatbots* não influenciou as métricas avaliadas, visto que o módulo DU oferece recomendações objetivas sobre as áreas que requerem aprimoramento, independente do tamanho da estrutura do assistente virtual. Embora um *design* mais complexo do *chatbot* possa demandar um esforço adicional na execução das melhorias sugeridas, tal complexidade não se refletiu em um impacto mensurável nas métricas avaliadas.

Por fim, é importante ressaltar que nenhum dos assistentes virtuais mencionados utilizava dados pessoais ou sensíveis em seu conteúdo de treinamento. Tal característica foi fator determinante na seleção desses *chatbots* como objetos de estudo para o experimento. Essa abordagem contribuiu significativamente para mitigar preocupações éticas e legais, assegurando conformidade com os padrões de proteção à privacidade e segurança de dados.

5.4 Ambiente e parâmetros de configuração

Para garantir a liberdade necessária na condução do experimento, e proteger o funcionamento dos *chatbots* em seus ambientes reais, decidiu-se por criar cópias desses assistentes virtuais em um ambiente de teste isolado. Para isso, o *design* e todas as características de cada *chatbot* foram replicados, incluindo as integrações com a plataforma Serprobots e o motor de conversação, garantindo que o comportamento no experimento fosse o mesmo do real.

Por outro lado, destaca-se que uma das características relevantes do *framework* DUBI é sua capacidade de lidar com peculiaridades de cada projeto de *chatbot*, permitindo a configuração de parâmetros específicos para a avaliação estática. Tais parâmetros podem ser configurados de acordo com recomendações do próprio DUBI, do fornecedor do motor de conversação ou conforme os requisitos do projeto. Esta característica, discutida anteriormente na Seção 4.3.2, contribui para que a proposta deste trabalho possa ser generalizada e aplicada a outros contextos que não apenas ao da plataforma Serprobots.

Neste cenário, para avaliar os *chatbots* de maneira adequada às suas particularidades, foram utilizados valores e critérios que serviram de base para este experimento. No entanto, é importante ressaltar que esses parâmetros devem ser estabelecidos pela equipe de curadoria do assistente virtual e podem variar conforme as características estruturais do *chatbot*, conforme já abordado anteriormente. Assim, as métricas abaixo foram parametrizadas da seguinte forma:

- BET: considerando que o balanceamento dos exemplos de treinamento das intenções pode ter papel determinante na classificação correta da intenção do usuário, é recomendado que se aceite valores baixos de desbalanceamento. Por isso, para o experimento esta métrica foi ajustada para aceitar no máximo 25% de desbalanceamento.
- LET e LTR: os assistentes virtuais do experimento foram modelados em português. Por isso, a análise de legibilidade utilizou o índice *Flesch-Kincaid grade level* adaptado para a língua portuguesa [44]. Para cada *chatbot* este índice foi configurado de acordo com seu público-alvo, conforme denota a Tabela 5.1.
- QET e TET: estas métricas foram configuradas com limiares de $5 \leqslant QET \leqslant 20$ e $3 \leqslant TET \leqslant 20$, considerando as recomendações do motor de conversação utilizado pelos *chatbots* [14].
- RET e RTR: considerando que os limiares aceitos por tais métricas são determinados pelas características de cada projeto, para o experimento tais limiares foram configurados com os valores de 50%.
- SID: a similaridade de texto foi avaliada a partir da distância de cosseno entre os vetores de palavras [85]. Valores de similaridade a partir de 0,8 foram considerados inadequados.

Por fim, conforme detalhado nas Seções 4.4.1 e 4.4.2, é importante destacar que o *framework* DUBI utiliza grandes modelos de linguagem para gerar as mensagens de testes. Ao interagir com esses modelos, é necessário fornecer duas instruções específicas a cada solicitação. A primeira, conhecida como mensagem do sistema, orienta como o modelo deve se comportar e fornece o contexto necessário para que ele possa gerar uma resposta adequada. Já a segunda instrução é denominada "*prompt*", e instrui o modelo sobre a ação específica a ser realizada. Para este experimento, foram utilizadas as seguintes instruções, parametrizadas com o atributo "{textos}" que era substituído pelos exemplos de treinamento das intenções dos *chatbots*:

- **Mensagem do sistema**: instrução sobre como o modelo deve se comportar. Utilizada a cada requisição, independente do tipo de mensagem de teste desejado.

> *Dado um conjunto de textos no formato "TEXTOS = ['texto 1', 'texto 2', 'texto 3', ..., 'texto n']", analise esses textos para identificar as palavras-chave mais relevantes no contexto do conjunto. Em seguida, crie perguntas utilizando essas palavras-chave e seus sinônimos, mesmo que estes não estejam na lista original. Você deve seguir os seguintes passos:*
> *1. Extração de Palavras-Chave:*

1.1. Analise os textos fornecidos e identifique as palavras-chave mais frequentes e significativas. Considere substantivos, verbos e adjetivos essenciais para entender o contexto geral dos textos.
2.Geração de Perguntas:
2.1. Utilize as palavras-chave identificadas e seus sinônimos para criar perguntas relevantes. As perguntas podem ser sobre definições, detalhes, causas, efeitos ou qualquer outro aspecto relacionado às palavras-chave.
3. Formato do retorno:
3.1. Retorne as frases no seguinte formato: "FRASES=['Frase1', 'Frase2', 'Frase3', ..., 'Frase-n']".
3.2. Se não for possível gerar alguma frase, o texto 'SB-NAO-CONSEGUI' deve ser utilizado no respectivo lugar da frase.

- ***Prompt* tipo 1:** conforme informado na Seção 4.4.2, as mensagens do tipo 1 são aquelas que fazem parte do escopo de treinamento do *chatbot* e sem erros de grafia. A partir destas mensagens, é possível aferir as métricas de desempenho do assistente virtual, em relação à capacidade de identificar corretamente as intenções dos usuários.

  Gere 3 frases utilizando as palavras-chave, ou seus sinônimos, presentes nos textos da lista a seguir: <LISTA_TEXTOS>{textos}</LISTA_TEXTOS>.

- ***Prompt* tipo 2:** gera as mensagens que também estão no escopo do assistente virtual, entretanto, geradas propositalmente com erros de grafia. Estas mensagens desafiam o chatbot a lidar com essas situações, além de viabilizar a aferição da métrica taxa de compreensão (T_C), apresentada na Seção 2.2.4.

  Gere 3 frases utilizando as palavras-chave, ou seus sinônimos, presentes nos textos da lista a seguir: <LISTA_TEXTOS>{textos}</LISTA_TEXTOS>. Na geração de cada frase, insira propositalmente erros ortográficos ou gramaticais.

- ***Prompt* tipo 3:** gera mensagens com textos que estão fora do escopo de treinamento do assistente virtual, podendo apresentar ou não erros de grafia. O objetivo dessas mensagens é avaliar a capacidade do *chatbot* reconhecer que aquele assunto não faz parte do seu escopo de aprendizado.

  Gere 3 frases que questionem sobre assuntos completamente fora do escopo dos textos apresentados na lista a seguir. As frases geradas podem conter ou não erros de grafia.
  <LISTA_TEXTOS>{textos}</LISTA_TEXTOS>.

As duas partes, mensagem do sistema e *prompt*, são usadas em conjunto para guiar o modelo de linguagem não apenas na geração de uma resposta coerente e contextualmente apropriada, mas também que o faça nos parâmetros operacionais desejados. Enquanto o

prompt é a instrução que se deseja que o modelo execute, a mensagem do sistema adota um formato mais orientativo sobre como o modelo deve se comportar durante aquela execução.

5.5 Resumo do capítulo

Neste capítulo, foi apresentado o experimento de validação destinado a comprovar a viabilidade técnica e conceitual do *framework* DUBI. A metodologia adotada foi detalhada, enfatizando o fluxo da execução do experimento para responder a pergunta de pesquisa. Além disso, o capítulo ofereceu uma caracterização dos *chatbots* empregados no estudo, assim como descreveu o ambiente e os parâmetros estabelecidos para a condução do experimento.

No próximo capítulo, os resultados alcançados através deste experimento serão apresentados e discutidos em detalhes, proporcionando que reflexões sobre a eficácia do *framework* DUBI sejam trazidas à luz.

Capítulo 6

Análise de resultados do *framework* DUBI

O presente capítulo é dedicado à apresentação e ao exame detalhado dos resultados obtidos no experimento descrito no Capítulo 5. Aqui são apresentados os dados de cada rodada de avaliação para todos os *chatbots* objetos do estudo, permitindo avaliar a correlação entre as métricas de desempenho e as melhorias implementadas sugeridas pelo *framework* DUBI. As conclusões aqui expostas complementam e expandem as descrições metodológicas do capítulo anterior, onde o experimento foi delineado.

Para isso, esse texto apresenta uma breve introdução na Seção 6.1 com considerações prévias sobre os resultados, bem como exemplifica o processo de intervenções realizadas nos *chatbots* na Seção 6.2. Na sequência, o capítulo está assim dividido: a Seção 6.3 aborda os resultados do "Bot-1"; na Seção 6.4, os resultados do "Bot-2" são explorados; e a Seção 6.5 se concentra nos resultados do "Bot-3". Em seguida, a Seção 6.6 sintetiza os achados em uma discussão geral, e a Seção 6.7 encerra o capítulo com um resumo das principais conclusões.

6.1 Metodologia e considerações sobre os resultados

Conforme detalhado na Seção 5.2, o experimento seguiu uma metodologia que permitisse validar se os aprimoramentos estruturais sugeridos pelo DUBI resultam em melhorias no desempenho dos *chatbots*. Em resumo, o procedimento consistiu em avaliar cada um dos *chatbots* objetos do estudo tanto em termos estáticos quanto interativos, ao longo de quatro rodadas de avaliação. Esta delimitação foi estabelecida para assegurar um parâmetro controlado que permitisse uma análise comparativa entre os assistentes virtuais. Entretanto, é importante salientar que na aplicação real do *framework* DUBI, o número

de interações de aprimoramento não é fixo, mas sim adaptável às necessidades de cada projeto, sendo definido conforme a equipe de desenvolvimento julgar apropriado.

No experimento, a cada rodada de avaliação foram implementadas melhorias nos *chatbots* com base nas recomendações do módulo DU. Como consequência, novas versões dos assistentes virtuais eram geradas, as quais eram submetidas a uma nova rodada de avaliação. Os resultados obtidos foram então comparados com os das rodadas anteriores para verificar o progresso e a eficácia das intervenções. Para fornecer uma visão mais detalhada sobre essas modificações, exemplos das intervenções realizadas serão apresentados na Seção 6.2.

Destaca-se que, o experimento foi conduzido com a intenção de simular a aplicação do DUBI em condições reais de uso, utilizando para isso assistentes virtuais pré-existentes. Dessa forma, foi possível observar o comportamento do *framework* sem alterações artificiais nos *chatbots* que enviesassem os resultados.

Devido a essa abordagem, algumas métricas específicas do módulo *Design Understanding* (DU) não foram observadas, pois os assistentes virtuais já estavam consoantes aos padrões recomendados, possivelmente devido a testes e homologações anteriores realizadas pelas respectivas equipes de desenvolvimento. Entretanto, a falta de dados para essas métricas não compromete a validade geral do estudo, uma vez que o foco estava em testar a suposição de que melhorias estruturais nos *chatbots* resultam em um desempenho aprimorado, e não em alcançar a versão ideal de cada *chatbot*.

Assim, neste capítulo os resultados serão apresentados categorizados por *chatbot*. As seções seguintes apresentarão, para cada assistente virtual, os respectivos resultados obtidos tanto nas avaliações estáticas quanto nas interativas, englobando todas as rodadas executadas. Também serão analisados os impactos no desempenho do *chatbot*, identificados pelas métricas do módulo BI, gerados pelas alterações implementadas nos respectivos *chatbots* e apontadas pelas métricas DU. Todos os valores das métricas dos assistentes virtuais “Bot-1”, “Bot-2” e “Bot-3” são apresentados, respectivamente, nas Tabelas 6.1, 6.2 e 6.3, onde os melhores valores de cada métrica estão destacados em negrito.

Em relação à execução do módulo DU, este capítulo se concentrará na exposição das métricas e na análise dos impactos das melhorias implementadas, sendo o processo de implementação destas melhorias exemplificado na próxima seção. Quanto à avaliação interativa, os valores que serão expostos correspondem às médias obtidas após as 10 execuções de cada avaliação do módulo *chatBot Intelligence* (BI), conforme descrito na Seção 5.2. De modo que o detalhamento dos relatórios das avaliações estática e interativas serão apresentados no Apêndice D.

6.2 Intervenções realizadas nos *chatbots*

As intervenções realizadas nos *chatbots* do experimento foram implementadas com base nos pontos de melhoria identificados pelo módulo DU do *framework* DUBI. Como já abortado anteriormente no Capítulo 4, o relatório de avaliação estática evidencia se os componentes observados em uma avaliação, tais como intenções, exemplos de treinamento ou nós de diálogo, estão adequados em relação aos parâmetros pré-estabelecidos. Para os que não estão, o relatório detalha o componente específico com a potencial falha, conforme modelo apresentado no Apêndice A.

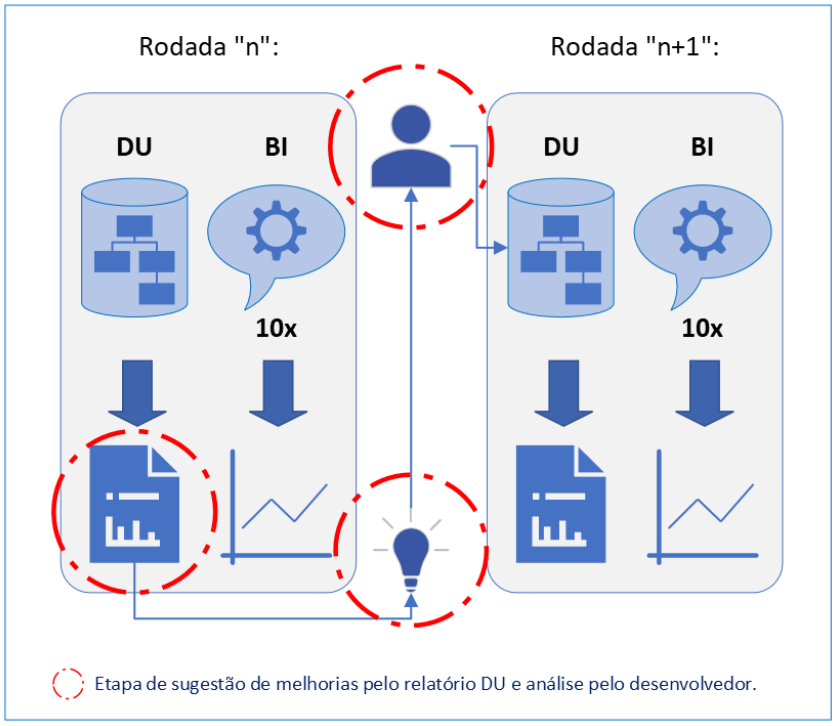


Figura 6.1: Detalhamento da etapa de melhorias baseadas no relatório DU.

Fonte: autoria própria.

Esta etapa de melhorias é ilustrada na Figura 6.1, através dos círculos pontilhados. Resumidamente, o relatório da avaliação estática da "Rodada n" detalha as métricas aferidas e, para as que não estão adequadas aos parâmetros pré-estabelecidos, aponta especificamente os aspectos que requerem atenção. Com isso, a equipe responsável pelo desenvolvimento do *chatbot* utiliza essas informações para determinar a necessidade de intervenções. Caso sejam realizadas, uma nova versão do assistente virtual é criada e submetida à "Rodada n+1" de avaliação.

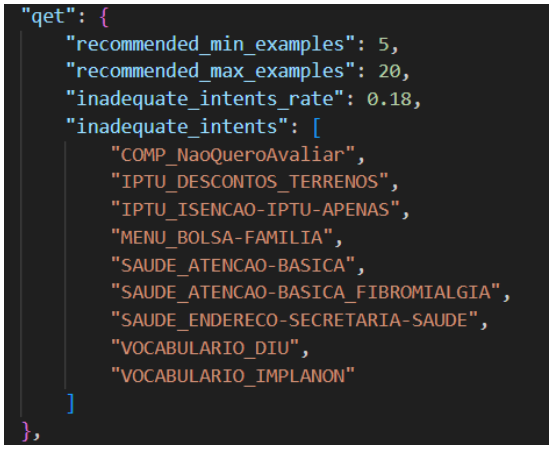
Para exemplificar as intervenções realizadas durante o experimento, esta seção descreve algumas implementações feitas nos *chatbots*. Embora estes exemplos não sejam exaustivos em relação a todas as métricas avaliadas, eles ajudam a entender a dinâmica

das modificações. Neste contexto, destaca-se que os detalhes completos dos relatórios avaliativos dos assistentes virtuais do experimento estão disponíveis no Apêndice D.

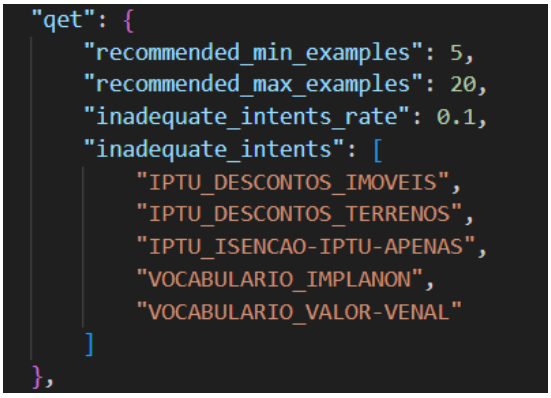
Assim, é importante ressaltar que o objetivo desta seção não é realizar o detalhamento e a análise dos resultados do experimento, visto que são os objetos de discussão das Seções 6.3 a 6.6. No entanto, a apresentação desses exemplos é relevante para auxiliar na compreensão das análises discutidas no decorrer deste capítulo.

6.2.1 Exemplos de intervenções realizadas no "Bot-1"

Na primeira rodada de avaliação (R1), o relatório DU indicou que o *chatbot* "Bot-1" possuía 9 intenções com quantidades inadequadas de exemplos de treinamento, representando 18% das intenções. Essas intenções tinham ou menos de 5, ou mais de 20 exemplos de treinamento, conforme apresentado na Figura 6.2a. Neste cenário, as intervenções realizadas, para que elas passassem a atender aos valores recomendados, consistiram em incluir novos exemplos nas intenções com QET inferior a 5, e em eliminar exemplos de treinamento das intenções com mais de 20. Como resultado, na segunda rodada de avaliação (R2), conforme mostrado na Figura 6.2b, a porcentagem de intenções inadequadas diminuiu para 10%. Entre R1 e R2, 4 intenções foram totalmente corrigidas, restando 5 intenções para melhorias futuras.



(a) Trecho do relatório DU para a métrica QET do "Bot-1" na R1.



(b) Trecho do relatório DU para a métrica QET do "Bot-1" na R2.

Figura 6.2: Comparação da métrica QET entre as R1 e R2 do "Bot-1"

Outro exemplo interessante para destacar nas intervenções realizadas no "Bot-1" se refere à métrica LET. Neste cenário, na segunda rodada de avaliação (R2), o relatório de avaliação estática identificou que 36% das intenções deste *chatbot* tinham legibilidade inadequada. Estas intenções são apontadas pelo relatório, detalhando a legibilidade apurada, conforme ilustra a Figura 6.3a. Neste caso, as intervenções focaram na melhoria dos

textos dos exemplos de treinamento das referidas intenções. Com isso, na terceira rodada de avaliação (R3), a inadequação caiu para 28%, como ilustrado na Figura 6.3b. Como exemplo, a intenção "IPTU_DESCONTOS-PAGAMENTO-PARCELADO" melhorou de um índice de 34,33 para 37,02, embora ainda abaixo do valor esperado de 50. Em contrapartida, a intenção "COMP_Avaliacao" piorou de 47,29 para 45,94. Já a intenção "IPTU_ISENCAO-IPTU-APENAS", que estava inadequada na R2, foi adequada na R3, não sendo mais listada nesta rodada, indicando que as alterações realizadas sobre ela surtiram o efeito esperado.


```
"let": {
    "idx_min_expected_readability": 50.0,
    "desc_min_expected_readability": "Ensino médio: 1o ano.",
    "inadequate_intents_rate": 0.36,
    "inadequate_intents": [
        {
            "intent": "COMP_Avaliacao",
            "idx_readability": 47.29,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_DESCONTOS-PAGAMENTO-PARCELADO",
            "idx_readability": 34.33,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_DOCUMENTOS-ARBORIZACAO",
            "idx_readability": 40.85,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_DOCUMENTOS-DOADORES-MEDULA",
            "idx_readability": 46.88,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_ISENCAO-IPTU-APENAS",
            "idx_readability": 0.0,
            "desc_readability": "Profissional."
        },
        {
            "intent": "IPTU_REQUERIMENTO-DESCONTO",
            "idx_readability": 44.07,
            "desc_readability": "Ensino superior: em curso."
        },
        {...
```


(a) Trecho do relatório DU para a métrica LET do "Bot-1" na R2.


```
"let": {
    "idx_min_expected_readability": 50.0,
    "desc_min_expected_readability": "Ensino médio: 1o ano.",
    "inadequate_intents_rate": 0.28,
    "inadequate_intents": [
        {
            "intent": "COMP_Avaliacao",
            "idx_readability": 45.94,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_DESCONTOS-PAGAMENTO-PARCELADO",
            "idx_readability": 37.02,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_DOCUMENTOS-ARBORIZACAO",
            "idx_readability": 40.85,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_DOCUMENTOS-DOADORES-MEDULA",
            "idx_readability": 46.88,
            "desc_readability": "Ensino superior: em curso."
        },
        {
            "intent": "IPTU_REQUERIMENTO-DESCONTO",
            "idx_readability": 44.07,
            "desc_readability": "Ensino superior: em curso."
        },
        {...
```

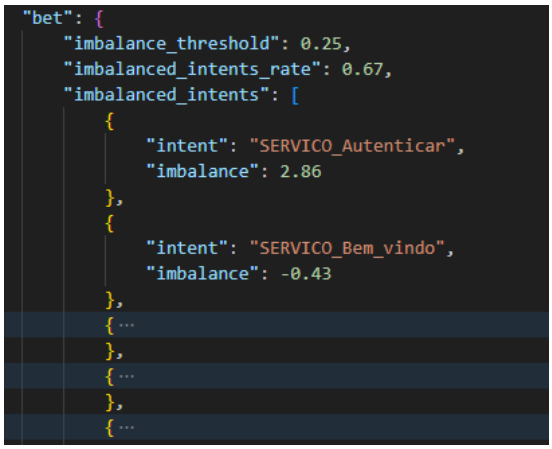

(b) Trecho do relatório DU para a métrica LET do "Bot-1" na R3.

Figura 6.3: Comparação da métrica LET entre as R2 e R3 do "Bot-1"

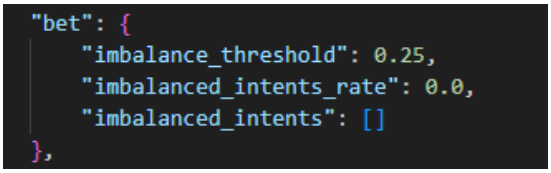
6.2.2 Exemplos de intervenções no "Bot-2"

No contexto do "Bot-2", na avaliação inicial (R1) a métrica *BET* indicou que este *chatbot* apresentava 67% de intenções não conformes com o critério de balanceamento estabelecido, como apresentado na Figura 6.4a. Dentre as intenções inadequadas, exemplifica-se as "SERVICO_Autenticar" e "SERVICO_Bem_vindo", que apresentaram desbalanceamento de 286% e -43%, respectivamente. A intenção "SERVICO_Autenticar", com um excesso de exemplos de treinamento, requereu a exclusão de um volume considerável de entradas para mitigar o superavit, enquanto a intenção "SERVICO_Bem_vindo" neces-

sitou da inclusão de exemplos adicionais para corrigir o déficit e alinhar-se à média de distribuição das demais intenções. Após essas ações corretivas, foi realizada uma segunda rodada de avaliação (R2), na qual o *chatbot* apresentou conformidade com a métrica BET, como pode ser verificado na Figura 6.4b.



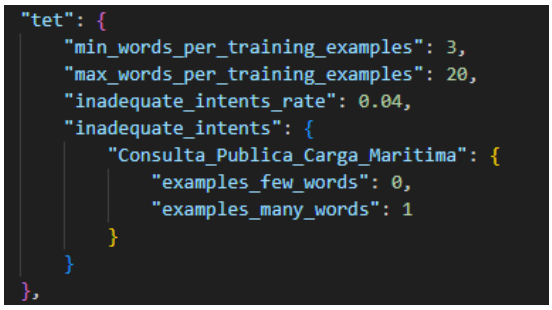
(a) Trecho do relatório DU para a métrica BET do "Bot-2" na R1.



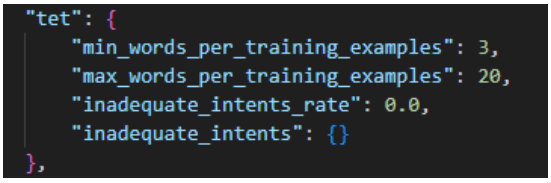
(b) Trecho do relatório DU para a métrica BET do "Bot-2" na R2.

Figura 6.4: Comparação da métrica BET entre as R1 e R2 do "Bot-2"

Outro exemplo interessante deste assistente virtual diz respeito à métrica TET. Após a avaliação da R2, o relatório DU indicou que 4% das intenções apresentavam exemplos de treinamento inadequados em relação ao tamanho dos seus textos. Como apresentado na Figura 6.5a, havia uma intenção específica que necessitava de ajuste. Para este caso, a avaliação estática apontou que a intenção possuía um exemplo de treinamento com mais palavras do que o máximo recomendado. Desta forma, foi necessário diminuir o tamanho do texto deste exemplo de treinamento. Após essa intervenção, na R3, o assistente virtual ficou adequado à métrica TET, como representado na Figura 6.5b.



(a) Trecho do relatório DU para a métrica TET do "Bot-2" na R2.

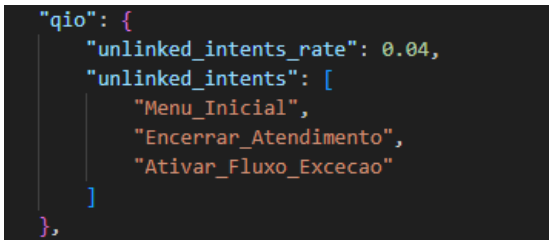


(b) Trecho do relatório DU para a métrica TET do "Bot-2" na R3.

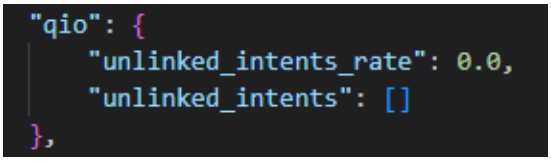
Figura 6.5: Comparação da métrica TET entre as R2 e R3 do "Bot-2"

6.2.3 Exemplos de intervenções no “Bot-3”

No âmbito do “Bot-3”, observou-se que na R1 este *chatbot* possuía 3 intenções órfãs, ou seja, não relacionadas a nenhum nó de diálogo, conforme indicado na Figura 6.6a. Para resolver este tipo de situação há duas alternativas, excluir as intenções órfãs ou associá-las a nós de diálogos (novos ou pré-existentes). A decisão sobre a melhor intervenção a ser feita é de responsabilidade da equipe de desenvolvimento, e deve ser tomada segundo a estratégia do projeto. Para o experimento, a intervenção escolhida foi a exclusão dessas intenções. Com isso, na segunda rodada de avaliação (R2), este *chatbot* passou a estar adequado a essa métrica, sem intenções órfãs restantes, como ilustrado na Figura 6.6b.



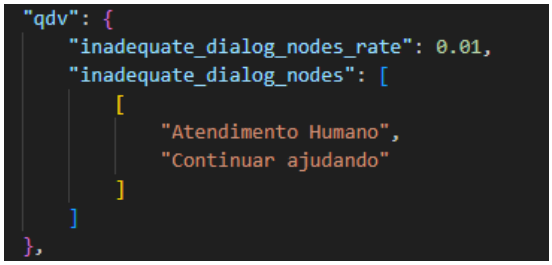
(a) Trecho do relatório DU para a métrica QIO do “Bot-3” na R1.



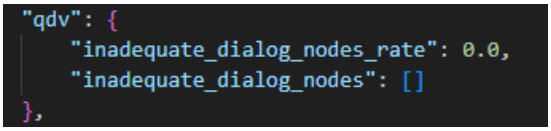
(b) Trecho do relatório DU para a métrica QIO do “Bot-3” na R2.

Figura 6.6: Comparação da métrica QIO entre as R1 e R2 do “Bot-3”

Como último exemplo, destaca-se ainda a intervenção realizada para corrigir a métrica QDV, também do “Bot-3”. Como apresentado na Figura 6.7a, o relatório da R1 revelou que dois nós de diálogo estavam configurados para sempre serem ativados, visto que suas condições de entrada estavam definidas como *“true”* (verdadeiro). Para corrigir essa falha de modelagem, optou-se por eliminar esses dois nós do fluxo conversacional. Consequentemente, na rodada de avaliação subsequente (R2), o *chatbot* foi considerado compatível com os padrões da métrica QDV, conforme mostrado na Figura 6.7b.



(a) Trecho do relatório DU para a métrica QDV do “Bot-3” na R1.



(b) Trecho do relatório DU para a métrica QDV do “Bot-3” na R2.

Figura 6.7: Comparação da métrica QDV entre as R1 e R2 do “Bot-3”

Por fim, é importante salientar que a escolha de quais intervenções realizar para aprimorar um *chatbot* deve ser baseada no contexto específico do projeto, além de ser uma

decisão que cabe à equipe de desenvolvimento. Contudo, a utilização de dados concretos, como os fornecidos pelo relatório de avaliação estática do *framework* DUBI, permite que essa decisão seja orientada por informações precisas, em vez de conjecturas ou experiência pessoais de integrantes das equipes. Assim, com base nos exemplos apresentados, fica evidenciada a eficácia e objetividade do *framework* DUBI ao avaliar a modelagem dos assistentes virtuais e, principalmente, indicar potenciais partes a serem melhoradas. Possibilitando que os desenvolvedores realizem ajustes informados e orientados por dados, garantindo intervenções mais assertivas e eficientes.

Nas seções seguintes deste capítulo, o detalhamento e a análise dos resultados de cada um dos *chatbots* estudados são apresentados.

6.3 Resultados experimentais do "Bot-1"

Nesta seção, os resultados obtidos pelo "Bot-1" serão apresentados e discutidos. Neste cenário, relembrando o que foi apresentado no capítulo anterior, este *chatbot* foi desenvolvido para servir como um assistente virtual para esclarecer dúvidas dos cidadãos sobre serviços municipais fornecidos por uma prefeitura. A equipe que o desenvolveu já possuia experiência em projetos de modelagem de *chatbots*, e o público-alvo deste assistente virtual são pessoas alfabetizadas com, pelo menos, o ensino fundamental completo.

Os resultados das métricas aferidas durante os testes desse *chatbot*, são apresentados na Tabela 6.1. Nela, as linhas representam as métricas observadas, enquanto as colunas discriminam os resultados obtidos em cada uma das rodadas de avaliação.

Neste contexto, a análise dos resultados após as quatro rodadas de avaliação do "Bot-1" fornece indícios sobre a relação entre as métricas de qualidade do *design* (fornecidas pelo módulo DU) e as de desempenho (aferidas pelo BI). De maneira geral, observa-se uma melhora progressiva nas métricas de BI correlacionada com melhorias em várias métricas de DU, sugerindo que as intervenções na modelagem do *chatbot* tiveram um impacto positivo em seu desempenho.

Inicialmente, com seu conteúdo original, o "Bot-1" apresentou acurácia de 0,7474 na R1, que foi aumentada para 0,8264 na terceira rodada, antes de sofrer uma leve queda para 0,8198 na rodada 4. Estes valores representam um ganho de, aproximadamente, 10,5%. Essa tendência foi acompanhada pelas métricas de precisão, *F1-score* e taxa de consistência das respostas, que apresentaram ganhos respectivos de 5,4%, 6% e 10,7%. Paralelamente, as métricas *recall*, taxa de *fallback* e taxa de compreensão melhoraram constantemente a cada rodada de avaliação, alcançando, respectivamente, ganhos de 8,6%, 59,6% e 13,8%. Já a taxa de ambiguidade apresentou melhor resultado na segunda rodada de avaliação, quando atingiu seu mínimo de 0,0692, após iniciar em 0,1093. Nas avaliações

Tabela 6.1: Resultados da avaliação do "Bot-1".

Métrica	**Rodada 1 (R1)**	**Rodada 2 (R2)**	**Rodada 3 (R3)**	**Rodada 4 (R4)**
Módulo BI				
Acurácia	0,7474	0,8236	**0,8264**	0,8198
Precisão	0,8264	0,8624	**0,8712**	0,8487
Recall	0,8824	0,9469	0,9400	**0,9584**
F1-score	0,8533	0,9023	**0,9040**	0,9000
tm_R	**1,2854**	1,2985	1,3177	1,3134
T_f	0,1081	0,0539	0,0606	**0,0436**
T_c	0,7484	0,8412	0,8391	**0,8522**
T_{CR}	0,7477	0,8253	**0,8280**	0,8202
ψ	0,1093	**0,0692**	0,0712	0,0850
Cobertura de intenções	0,9360	0,9460	0,9600	**0,9700**
Cobertura de nós de diálogo	0,7915	0,8447	0,8638	**0,8745**
Módulo DU				
QET	0,18	0,10	0,00	**0,00**
BET	0,50	0,50	0,42	**0,00**
TET	0,96	0,96	0,88	**0,50**
SID	0,12	0,12	0,12	**0,10**
LET	0,44	0,36	0,28	**0,28**
RET	0,92	0,92	0,90	**0,66**
QIO	0,00	0,00	0,00	0,00
TTR	0,00	0,00	0,00	0,00
LTR	0,42	0,42	0,42	0,42
RTR	0,02	0,02	0,02	0,02
STR	0,02	0,02	0,02	0,02
QID	0,00	0,00	0,00	0,00
QDV	0,00	0,00	0,00	0,00
QDF	0,00	0,00	0,00	0,00
QDR	0,00	0,00	0,00	0,00
QCF	0	0	0	0

seguintes (R3 e R4), houve um leve aumento, mas ainda assim abaixo do valor inicial, resultando em ganho de 22,2% na rodada 4 em relação ao seu índice inicial.

As métricas relacionadas à cobertura dos testes, tanto de intenções quanto de nós de diálogo, também mostraram um aumento gradual e consistente. Ao final da rodada 4 de avaliação, a cobertura do teste alcançou 97% das intenções do *chatbot* e 87,45% dos nós de diálogos existentes.

No contexto do módulo *Design Understanding*, as métricas que tiveram melhorias foram as BET, LET, QET, RET, SID e TET. A elas é atribuída a responsabilidade pela melhora das métricas de desempenho. As demais métricas do módulo DU não foram afetadas, por dois motivos: (i) o "Bot-1"já está ajustado aos valores de referência, incluindo QIO, TTR e todas as relacionadas aos fluxos de conversa, conforme detalhado na Tabela 4.1; e (ii) não foram feitas alterações no *chatbot* que impactassem métricas específicas, como LTR, RTR e STR, já que o objetivo do experimento não era o de obter a versão ótima da modelagem do assistente virtual.

Os dados da Tabela 6.1 sugerem haver relação entre as métricas BET e QET com as de desempenho. A melhora em QET (quantidade de exemplos de treinamento), refletida pelas intervenções realizadas no assistente virtual e exemplificadas na Seção 6.2.1, representa um aumento da diversidade de exemplos de treinamento, o que provavelmente ajudou o *chatbot* a reconhecer uma gama maior de variações nas intenções dos usuários, resultando na melhora da acurácia e da *F1-score*.

Ao melhorar a QET, consequentemente, há impactos positivos na BET (balanceamento dos exemplos de treinamento). Esta última, por sua vez, atingiu o valor ótimo da rodada 4, indicando um equilíbrio ideal entre as intenções. Isto é importante, pois um bom balanceamento evita o viés do assistente virtual para intenções com mais exemplos, resultando em uma melhor precisão e, consequentemente, um *F1-score* mais alto. Com menos viés, há também uma queda na taxa de *fallback*, uma vez que o *chatbot* se torna mais capaz de identificar a intenção correta.

Já a redução das métricas relacionadas à legibilidade, à representatividade e ao tamanho dos textos dos exemplos de treinamento (LET, RET e TET) indicam que os exemplos se tornaram mais relevantes e significativos, o que provavelmente ajudou a aprimorar as taxas de compreensão e de consistência das respostas do *chatbot*.

Entende-se que as mudanças implementadas na modelagem do "Bot-1" viabilizaram uma melhor estruturação do seu conteúdo de treinamento, resultando em textos mais bem formatados. Provavelmente, essa situação impactou positivamente a etapa de geração de mensagens de testes via modelo de linguagem, tendo em vista que o conteúdo de entrada para este modelo foi melhorando a cada rodada de avaliação. Como consequência, a cobertura dos testes apresentou incrementos constantes.

Neste cenário, entre as rodadas de testes R3 e R4, observou-se uma melhora em torno de 1% nas métricas *recall*, taxa de compreensão (T_C) e na cobertura dos testes. Esses ganhos indicam um avanço na capacidade do *chatbot* de entender e responder corretamente às intenções dos usuários, além de uma maior abrangência dos testes realizados. Por outro lado, houve uma leve redução de 0,4% no *F1-score*, o que é, para este contexto, estatisticamente insignificante. Isso sugere que, apesar do aumento da cobertura e dos desafios adicionais trazidos pelos testes mais extensivos, o desempenho geral do assistente virtual se manteve estável.

Portanto, é possível concluir que as intervenções baseadas no módulo DU tiveram um impacto benéfico nas métricas de BI, otimizando a eficácia geral do *chatbot* e evidenciando a eficiência do *framework* DUBI na avaliação da qualidade de *chatbots*.

6.4 Resultados experimentais do "Bot-2"

Esta seção apresenta e analisa os resultados das avaliações realizadas sobre o "Bot-2". Este *chatbot* foi construído para ser utilizado em um evento de treinamento e, por isso, apresenta uma estrutura mais simplificada quando comparado aos demais. Ele foi desenvolvido por uma equipe com experiência no desenvolvimento de *chatbots*, tendo como público-alvo usuários com o ensino superior completo. Interessante notar que esse assistente era o que ofertava o menor desempenho em seu conteúdo original, quando comparado aos outros dois estudados no experimento, sendo talvez uma consequência do objetivo para o qual foi criado, ou seja, ser um *chatbot* a ser utilizado em uma capacitação e não um projeto de produção real.

A Tabela 6.2 apresenta os resultados das quatro rodadas de avaliação deste assistente. Assim como a tabela da seção anterior, nesta, as linhas também representam as métricas observadas, enquanto as colunas informam os resultados obtidos em cada uma das rodadas de avaliação.

Neste cenário, ao examinar os resultados das quatro rodadas de avaliação do "Bot-2" percebe-se uma melhoria contínua nas métricas de desempenho aferidas pelo módulo BI. Esta evolução está em consonância com as otimizações realizadas nos aspectos estruturais e linguísticos da modelagem do assistente virtual, como as intervenções nas métricas BET e TET apresentadas na Seção 6.2.2.

Na rodada inicial, o "Bot-2" demonstrou um desempenho modesto nos indicadores BI, com uma precisão de 0,6494 e acurácia de 0,5675. Esse comportamento foi acompanhado por taxas de *fallback* (T_f) e de ambiguidade (ψ) altas, respectivamente, 0,1542 e 0,2593, sinalizando espaço para aprimoramentos na capacidade de resposta do *chatbot*. No módulo *Design Understanding*, métricas como o balanceamento dos exemplos de treinamento

(BET), a quantidade de exemplos de treinamento (QET) e o tamanho dos exemplos de treinamento (TET) apresentaram valores altos, sugerindo desequilíbrios no conjunto de treinamento.

Após a primeira intervenção, baseada nas recomendações do módulo DU, todas as métricas do *chatBot Intelligence* registraram ganhos significativos na segunda rodada (R2). A acurácia, que havia apresentado um valor muito baixo, alcançou o patamar de 0,8557, representando uma evolução 50,78% entre a R2 e a R1. Já as taxas de *fallback* e ambiguidade, que haviam registrado altos valores, também passaram a apresentar índices bem satisfatórios, reduzindo-se para 0,0447 e 0,0955 respectivamente. Estas reduções representaram impressionantes melhorias de 71% na taxa de *fallback*, e de 63% para a taxa de ambiguidade.

Os dados da Tabela 6.2 mostram que todas as outras métricas de desempenho seguiram esta mesma tendência, como a *F1-score* que melhorou aproximadamente 28% e a taxa de compreensão (T_C) que evoluiu 47%. Essa melhoria coincidiu com a normalização das métricas de QET e BET para zero, e TET que reduziu de 0,78 para 0,04, indicando uma otimização na distribuição e no tamanho dos exemplos de treinamento. Esta correlação sugere que o *chatbot* se beneficiou de um treinamento mais balanceado e representativo, resultando em respostas mais precisas.

Na terceira rodada de avaliação (R3), as métricas BI mostraram um novo avanço, desta vez mais discretos. Embora o *recall* e a taxa de *fallback* (T_f) tenham sofrido uma pequena piora, não se considera significativa, visto que as mudanças dos valores ocorreram a partir da terceira casa decimal. A exceção a esta situação foi a taxa de ambiguidade (ψ), que nesta rodada de avaliação diminuiu de 0,0955 para 0,0799, representando um ganho de 16% aproximadamente. Do ponto de vista do módulo DU, destaca-se o alcance do valor ótimo da métrica similaridade entre intenções distintas (SID), ocasionado pela alteração dos textos dos exemplos de treinamento das intenções, o que provavelmente contribuiu para esta redução interessante da taxa de ambiguidade. Esta melhoria sugere que estas modificações ajudaram ao assistente virtual se tornar mais hábil em discernir entre diferentes intenções.

Finalmente, na quarta rodada (R4), as métricas do *chatBot Intelligence* indicaram uma estabilidade ou melhoria leve como as acurácia, precisão e *F1-score*. Notavelmente, a taxa de *fallback* (T_f) caiu para 0,0166, representando novo ganho de 66%. Paralelamente, as métricas do *Design Understanding* ainda mostraram uma progressão na legibilidade dos textos de respostas (LTR) e na representatividade dos exemplos de treinamento (RET), indicando um material de treinamento mais claro e alinhado com o perfil do público-alvo. Estas novas melhorias nos exemplos de treinamento, resultantes de alterações realizadas em seus textos, foram a razão pela qual o desempenho do *chatbot* evoluiu nesta última

Tabela 6.2: Resultados da avaliação do "Bot-2".

Métrica	**Rodada 1 (R1)**	**Rodada 2 (R2)**	**Rodada 3 (R3)**	**Rodada 4 (R4)**
Módulo BI				
Acurácia	0,5675	0,8557	0,8708	**0,8836**
Precisão	0,6494	0,8799	0,8986	**0,9179**
Recall	0,8019	**0,9661**	0,9631	0,9570
F1-score	0,7169	0,9207	0,9294	**0,9369**
tm_R	1,3866	**1,2593**	1,2654	1,3973
T_f	0,1542	0,0447	0,0493	**0,0166**
T_c	0,5923	0,8713	0,8747	**0,9041**
T_{CR}	0,5705	0,8588	0,8710	**0,8843**
ψ	0,2593	0,0955	0,0799	**0,0736**
Cobertura de intenções	0,9407	1,0	**1,0**	0,9815
Cobertura de nós de diálogo	0,6222	0,9481	**0,9482**	0,9407
Módulo DU				
QET	0,48	0,00	0,00	**0,00**
BET	0,67	0,00	0,00	**0,00**
TET	0,78	0,04	0,00	**0,00**
SID	0,15	0,15	0,00	**0,00**
LET	0,00	0,00	0,00	0,00
RET	0,48	0,22	0,22	**0,07**
QIO	0,00	0,00	0,00	0,00
TTR	0,00	0,00	0,00	0,00
LTR	0,23	0,20	0,17	**0,10**
RTR	0,00	0,00	0,00	0,00
STR	0,00	0,00	0,00	0,00
QID	0,00	0,00	0,00	0,00
QDV	0,00	0,00	0,00	0,00
QDF	0,00	0,00	0,00	0,00
QDR	0,00	0,00	0,00	0,00
QCF	0	0	0	0

rodada, visto que as demais métricas da avaliação estática já estavam em seus valores ótimos.

Assim, a análise detalhada das quatro rodadas de avaliação sugere uma relação causal entre o refinamento da modelagem do *chatbot*, conforme avaliado pelo módulo DU, e o aprimoramento de seu desempenho interativo, medido pelo módulo BI. À medida que o *design* e o conteúdo do assistente virtual foram melhorados — com exemplos de treinamento mais equilibrados, representativos e textos de resposta mais compreensíveis —, observou-se um impacto positivo nas métricas da avaliação interativa. Esses resultados corroboram a expectativa desta pesquisa, e indicam que as sugestões de melhorias propostas pelo *framework* DUBI possibilitam de fato o atingimento um desempenho superior por parte do assistente virtual.

6.5 Resultados experimentais do "Bot-3"

A presente seção é dedicada à apresentação e à análise dos resultados do "Bot-3", obtidos durante a execução das quatro rodadas de avaliação deste *chatbot* no experimento. Estes resultados são detalhados na Tabela 6.3, a qual segue a mesma estrutura das tabelas anteriores deste capítulo.

Como explicado na Seção 5.3, no momento do experimento, este assistente virtual ainda se encontrava em desenvolvimento, possuindo como escopo o atendimento a usuários de um serviço público federal. A expectativa da equipe de desenvolvimento, que estava atuando em seu primeiro projeto de modelagem de *chatbots*, é que o público-alvo do assistente tenha escolaridade mínima equivalente ao ensino médio completo.

Em relação à análise dos resultados das avaliações do "Bot-3", os dados sugerem que ele segue o mesmo sentido apontado nos dois assistentes avaliados anteriormente. Ou seja, há indícios de uma correlação entre as melhorias estruturais do *chatbot* e seu desempenho. De modo que, mais uma vez, as alterações implementadas e orientadas pelo módulo DU refletiram positivamente na eficiência do sistema ao responder às mensagens geradas.

A Tabela 6.3 mostra que as métricas do módulo BI apresentaram uma tendência ascendente durante as rodadas de avaliação. Destaca-se, em especial, a acurácia e a *F1-score* que alcançaram ganhos respectivos de 9% e 6%. Esse aumento é paralelo à redução nas métricas QET (quantidade de exemplos de treinamento) e TET (tamanho dos exemplos de treinamento), que alcançaram o valor ótimo de zero na quarta rodada de avaliação (R4). Isto indica uma possível influência da melhoria da qualidade do treinamento no desempenho do *chatbot*.

Outro ponto de destaque foi a melhora da taxa de *fallback* (T_f), que na rodada R2 diminuiu de 0,0833 para 0,0598, representando um ganho de 28% após a primeira inter-

venção. Esse avanço pode ser atribuído, além da melhoria nos exemplos de treinamento, à eliminação das intenções órfãs, aferidas pela métrica QIO. Os resultados da rodada R1 indicaram a existência de 4% de intenções sem relação com algum nó de diálogo, o que pode levar o *chatbot* a compreender a necessidade do usuário, mas não encontrar em seu fluxo uma resposta para retornar ao usuário. Essa situação foi corrigida eliminando as intenções com essa característica, conforme apresentado na Seção 6.2.3.

Tabela 6.3: Resultados da avaliação do "Bot-3".

Métrica	**Rodada 1 (R1)**	**Rodada 2 (R2)**	**Rodada 3 (R3)**	**Rodada 4 (R4)**
Módulo BI				
Acurácia	0,5728	0,5973	0,6167	**0,6257**
Precisão	0,6178	0,6314	0,6508	**0,6536**
Recall	0,8804	0,9138	0,9179	**0,9324**
F1-score	0,7260	0,7465	0,7614	**0,7682**
tm_R	1,3190	1,6144	1,7187	**1,3159**
T_f	0,0833	0,0598	0,0594	**0,0502**
T_c	0,5691	0,6068	0,6004	**0,6132**
T_{CR}	0,5724	0,5960	0,6167	**0,6260**
ψ	0,1356	0,1070	0,1006	**0,0974**
Cobertura de intenções	0,9513	0,9612	0,9512	**0,9962**
Cobertura de nós de diálogo	0,4023	0,4101	0,4149	**0,4344**
Módulo DU				
QET	0,20	0,15	0,04	**0,00**
BET	0,50	0,33	0,44	**0,04**
TET	0,59	0,41	0,35	**0,00**
SID	0,29	0,28	**0,23**	0,44
LET	0,11	**0,10**	0,10	0,11
RET	0,39	0,39	0,35	**0,34**
QIO	0,04	0,00	0,00	**0,00**
TTR	0,00	0,00	0,00	0,00
LTR	0,07	0,07	0,07	0,07
RTR	0,00	0,00	0,00	0,00
STR	0,01	0,01	0,01	0,01
QID	0,01	0,00	0,00	**0,00**
QDV	0,01	0,00	0,00	**0,00**
QDF	0,00	0,00	0,00	0,00
QDR	0,00	0,00	0,00	0,00
QCF	0	0	0	0

Observaram-se ainda consistentes nas taxas de compreensão (T_C) e de consistência das respostas (T_{CR}) do assistente virtual durante as rodadas de avaliação. Pelos dados do

experimento, não é possível relacionar essa melhora diretamente a alguma métrica do módulo DU. No entanto, considera-se que as intervenções realizadas nos textos dos exemplos de treinamento de diversas intenções, entre as rodadas R1 e R4, visando melhorá-los em termos de legibilidade e representatividade, podem ter ocasionado uma melhor habilidade do *chatbot* para compreender as intenções e manter a consistência das respostas.

A taxa de ambiguidade (ψ) também foi reduzida, implicando em uma clareza maior na seleção de respostas por parte do assistente virtual. Esta métrica também apresentou um avanço de, aproximadamente, 28%. Curiosamente, a métrica SID (similaridade entre intenções distintas), após duas reduções nas R2 e R3, aumentou na última rodada, o que poderia sugerir um potencial aumento de confusão na seleção intenções. No entanto, isso não parece ter afetado negativamente na taxa de ambiguidade. Embora possa ser uma razão pela qual este *chatbot* não tenha alcançados níveis de assertividade tão altos quanto os "Bot-1" e "Bot-2", conforme discutido na Seção 6.6.

É importante notar que algumas métricas do módulo DU, como TTR (tamanho dos textos de resposta), LTR (legibilidade dos textos de resposta), RTR (representatividade dos textos de resposta) e STR (sentimento negativo dos textos das respostas), permaneceram constantes ao longo das rodadas de avaliação, indicando que estes aspectos já estavam otimizados ou não foram o foco das intervenções.

Em resumo, os dados coletados ao longo das quatro rodadas de avaliação fornecem evidências de que as melhorias na estrutura e conteúdo do *chatbot*, conforme avaliado pelo módulo *Design Understanding* (DU), tiveram um impacto positivo sobre o desempenho do assistente virtual, conforme medido pelo módulo *chatBot Intelligence* (BI). A análise sugere que focar em aprimoramentos estruturais pode ser uma estratégia eficiente para elevar o desempenho geral de *chatbots*. Estando assim em consonância com a proposta do *framework* DUBI.

6.6 Discussão geral

Com base nos resultados apresentados anteriormente, esta seção realiza uma análise geral sobre as percepções que são possíveis de se obter ao observar os dados coletados ao longo das quatro rodadas de avaliação dos *chatbots* "Bot-1", "Bot-2" e "Bot-3", objetos do experimento de validação do *framework* DUBI. Aqui, buscou-se identificar padrões e relações entre as métricas do módulo DU e as do módulo BI, para validar se o DUBI está apto ao que se propõe, ou seja, avaliar e indicar melhorias automaticamente em assistentes virtuais, independente do contexto de negócio destes sistemas.

Neste cenário, observando os resultados das métricas do módulo DU e BI, foi possível identificar uma relação consistente entre a qualidade da modelagem do *chatbot* e seu

desempenho durante as interações simuladas. Por exemplo, no "Bot-1", melhorias nas métricas de quantidade de exemplos de treinamento (QET) e balanceamento dos exemplos de treinamento (BET) do módulo DU refletiram diretamente em aumentos na acurácia e *F1-score*, conforme avaliado pelo módulo BI. Este comportamento foi semelhante nos "Bot-2" e "Bot-3", sugerindo que um conteúdo de treinamento mais robusto e equilibrado fornece uma base sólida para o assistente virtual, melhorando sua capacidade de responder corretamente às mensagens dos usuários.

Além disso, ao analisar os resultados individuais dos três *chatbots*, é possível inferir que métricas que não alteram especificamente a estrutura, mas sim a qualidade dos textos dos exemplos de treinamento, como TET (tamanho dos exemplos de treinamento), SID (similaridade entre intenções distintas), LET (legibilidade dos exemplos de treinamento) e RET (representatividade dos exemplos de treinamento), geram um impacto positivo direto no desempenho do assistente virtual. Analisando estes resultados, observa-se que melhorias nessas métricas resultaram em uma redução nas taxas de *fallback* (T_f) e de ambiguidade (ψ), bem como um aumento nas taxas de compreensão (T_C) e de consistência das respostas (T_{CR}). Por exemplo, no "Bot-1", ao aprimorar a legibilidade e representatividade dos exemplos de treinamento, houve uma significativa diminuição na taxa de *fallback* e um aumento na compreensão, indicando que o *chatbot* pôde entender as intenções das mensagens recebidas de forma mais eficaz. Da mesma forma, no "Bot-2", a redução na similaridade entre intenções diminuiu a ambiguidade, permitindo respostas mais precisas e consistentes. Esses resultados sugerem que investir na qualidade textual dos exemplos de treinamento pode melhorar substancialmente o desempenho do assistente virtual, tornando suas respostas mais claras e adequadas às necessidades dos usuários.

Outra análise interessante que os dados permitiram fazer foi relacionada à QIO (quantidade de intenções órfãs). No caso do "Bot-3", a melhoria nesta métrica implicou na diminuição da taxa de *fallback* e ao aumento do *F1-score*, por exemplo. Isso se justifica porque, ao reduzir as intenções órfãs, o *chatbot* fica menos propenso a encontrar situações onde entende a necessidade do usuário, mas não possui uma resposta adequada em seu fluxo de diálogo, resultando em menor *fallback*. Além disso, sem intenções órfãs, o "Bot-3" apresentou um *F1-score* mais alto, refletindo um desempenho geral aprimorado, ao conseguir evitar respostas inadequadas e aumentar a precisão e *recall* das suas respostas.

Do ponto de vista das métricas relacionadas aos textos de respostas de um *chatbot* (TTR, LTR, RTR e STR), conforme discutido na Seção 4.3, elas não têm uma influência direta em seu desempenho, mas impactam significativamente a experiência do usuário. Por isso, no experimento, essas métricas não foram priorizadas, mas isto não diminui sua importância. Visto que o *framework* DUBI visa ser uma ferramenta que possibilite a melhoria contínua dos assistentes virtuais em diversos aspectos, não apenas no desempenho.

Em relação às métricas relacionadas aos fluxos de conversa, i.e. QID, QDV, QDF, QDR e QCF, parte delas não foi possível de se observar, devido ao fato dos *chatbots* estudados já estarem adequados a elas, conforme esclarecido na Seção 6.1. As exceções foram as QID (quantidade de nós de diálogo com condições de entrada iguais) e QDV (quantidade de nós de diálogo com condição de entrada sempre verdadeiro), que estavam inadequadas no conteúdo original do "Bot-3". Entretanto, por essa situação ter se caracterizada apenas em um dos assistentes virtuais, não é possível validar um comportamento padrão entre essas métricas e as de desempenho. Embora seja sabido que uma alta QID está associada a uma maior redundância no fluxo de diálogo, o que pode reduzir a precisão das respostas. Por outro lado, a QDV pode resultar em respostas inadequadas, visto que será desprezada a intenção do usuário na examinação do referido nó de diálogo com tal condição.

6.6.1 Conclusões sobre os resultados

Os resultados apresentados neste experimento corroboram com a viabilidade conceitual e técnica do *framework* DUBI e validam as suposições desta pesquisa no contexto dos testes realizados. Primeiramente, demonstram ser possível automatizar o ciclo de avaliação de *chatbots* baseados em recuperação de informação e de domínio fechado. Além disso, as métricas coletadas pelo módulo *Design Understanding* (DU) forneceram informações objetivas sobre pontos de melhoria na estrutura e no conteúdo dos assistentes virtuais, que, ao serem implementadas, resultaram em melhorias mensuráveis no desempenho, conforme registrado pelo módulo *chatBot Intelligence* (BI).

Este fato evidencia que o *feedback* automatizado sobre a modelagem do *chatbot* é uma estratégia eficaz para promover melhorias pontuais e impactantes. Como exemplificado pelas métricas de desempenho do "Bot-2", que, ao final das quatro rodadas de avaliação, destacou-se com resultados notáveis, apresentando um salto de 55% em acurácia e de 31% em *F1-score*, além de uma redução substancial de 89% na taxa de *fallback* e de 71% na taxa de ambiguidade. Por sua vez, o "Bot-1" e o "Bot-3" também exibiram avanços importantes, com ganhos em patamares similares entre si. O "Bot-1" alcançou um aumento de 9,7% em acurácia e de 5,5% em *F1-score*, juntamente com uma significativa redução de 59,67% na taxa de *fallback* e de 22,23% na taxa de ambiguidade. Paralelamente, o "Bot-3" apresentou um incremento de 9,2% em acurácia e de 5,81% em *F1-score*, com diminuição de 39,73% e de 28,17% na taxas de *fallback* e ambiguidade, respectivamente. Esses resultados sugerem que a eficácia das melhorias pode variar conforme a qualidade inicial do *chatbot* e a complexidade inerente às suas intenções e fluxos de conversa, mas o potencial de otimização é claro em todos os casos.

Em conclusão, a análise dos resultados obtidos através das avaliações individuais dos três assistentes virtuais confirma o êxito do *framework* DUBI em automatizar a avaliação de *chatbots* e em fornecer diretrizes objetivas para melhorias na modelagem. As relações identificadas entre as métricas do módulo DU e as métricas do módulo BI reforçam a importância de uma estrutura de treinamento e de diálogos bem definida e balanceada, evidenciando que melhorias estruturais e de conteúdo refletem diretamente no desempenho dos *chatbots*. Finalmente, apesar de não ter sido testado exaustivamente em diversos cenários, os resultados obtidos fornecem indícios promissores de que o DUBI pode ser aplicado a outros *chatbots* que tenham o escopo abordado na pesquisa, sugerindo seu potencial para se tornar uma ferramenta auxiliar no processo de desenvolvimento desses sistemas.

6.7 Resumo do capítulo

Este capítulo apresentou e analisou os resultados obtidos no experimento de validação do *framework* DUBI. Inicialmente foram realizadas considerações prévias sobre a apresentação dos resultados, visando preparar o leitor para a leitura das análises individuais dos *chatbots* estudados. Também foram exemplificadas algumas das intervenções realizadas nos *chatbots* experimentados, para ilustrar a objetividade das indicações de melhorias do módulo DU. Na sequência, para cada assistente virtual, foram apresentados os dados de cada rodada de avaliação, permitindo avaliar a correlação entre as métricas de desempenho e as melhorias implementadas sugeridas pelo módulo DU. Por fim, foi realizada uma discussão geral sobre esses resultados, buscando-se identificar padrões ao observar os dados dos “Bot-1”, “Bot-2” e “Bot-3” em conjunto. Esta análise permitiu concluir sobre a viabilidade e eficácia do *framework* DUBI em sua missão de avaliar automaticamente *chatbots*.

No próximo capítulo, serão apresentadas as conclusões desta dissertação, juntamente com propostas para trabalhos futuros. Estas seções visam encerrar o documento proporcionando uma visão clara do que foi produzido, destacando a contribuição da pesquisa para a literatura atual e identificando áreas para evolução e melhoria no futuro.

Capítulo 7

Conclusões e trabalhos futuros

Este trabalho abordou a problemática da avaliação de *chatbots*, especificamente aqueles de domínio fechado e baseados em intenções e recuperação de informação. Para esse contexto, é sabido que uma avaliação eficaz desses sistemas é fundamental para garantir sua qualidade e desempenho, porém, a literatura existente até então, conforme apresentado no Capítulo 3, não oferecia uma solução que considerasse simultaneamente os aspectos estáticos de modelagem e os aspectos interativos de desempenho. Essa falta de abordagem abrangente poderia resultar em avaliações enviesadas e limitadas, geralmente dependentes do conhecimento e da experiência dos avaliadores humanos.

Para preencher essa lacuna, foi proposto o *framework* DUBI, que combina a avaliação estática da estrutura e modelagem dos *chatbots* com a avaliação interativa, que observa, dentre outras, a assertividade, a consistência e a ambiguidade das respostas a mensagens submetidas. Para isso, o DUBI foi desenhado para automatizar esse processo, tornando os testes menos enviesados e mais abrangentes, pois não dependem do conhecimento específico de um humano. Assim, ele proporciona uma avaliação mais objetiva, célere e consistente.

Entende-se que o ponto forte da proposta reside na sua capacidade de identificar e quantificar automaticamente as áreas que necessitam de melhorias na modelagem dos *chatbots*, proporcionando assim um diagnóstico preciso das deficiências. Isso permite às equipes de desenvolvimento realizar uma análise detalhada das causas dos problemas de desempenho, ainda em fase de pré-implantação. Desta forma, durante o processo de curadoria dos assistentes virtuais, é possível obter linhas-base comparativas entre as diferentes versões do sistema. Assim, o *framework* DUBI pode oferecer evidências concretas de que as alterações realizadas nos assistentes virtuais não prejudicarão o desempenho após a implantação. Com isso, aumenta-se a garantia de que a implantação de uma nova versão do *chatbot* não resultará em perdas de imagem ou em quebras de expectativa do ponto de vista da experiência dos usuários.

Este ponto forte foi comprovado por meio de um experimento realizado durante este trabalho. Nele, três *chatbots* reais foram submetidos às avaliações do *framework* DUBI. Os resultados deste experimento confirmaram que as recomendações fornecidas pelo DUBI, após implementadas, resultaram em melhorias significativas no desempenho dos *chatbots*. Portanto, para o contexto avaliado, verificou-se a eficácia do *framework* em proporcionar uma análise objetiva e detalhada, essencial para o aprimoramento contínuo dos assistentes virtuais.

Por exemplo, um dos *chatbots* testados obteve ganhos de 55% em sua acurácia e 31% na *F1-score*, além de uma redução de 89% na taxa de *fallback* e 71% na taxa de ambiguidade. Os outros assistentes virtuais seguiram numa linha parecida, obtendo ganhos superiores a 10% em outras métricas, como as taxas de compreensão e consistência das respostas. Esses resultados destacam a importância de uma modelagem bem estruturada para o sucesso dos *chatbots* em aplicações práticas.

Ainda sobre as conclusões obtidas do experimento de validação, foi possível comprovar a relação direta entre a qualidade da modelagem dos *chatbots* e seu desempenho operacional, evidenciando a importância de uma modelagem bem estruturada para o sucesso dos assistentes virtuais em aplicações práticas. Além disso, também foi possível comprovar que o *framework* DUBI tende a proporcionar testes menos enviesados, visto que independem do conhecimento humano para serem realizados, potencializando a identificação de falhas e a proposição de melhorias que realmente impactam a qualidade final do sistema.

Em relação às contribuições deste trabalho, considera-se serem significativas tanto para o meio acadêmico quanto para o profissional, em especial para o SERPRO.

Do ponto de vista acadêmico e científico, o DUBI representa um avanço em relação ao que estava disponível na literatura atual. Ele oferece uma forma automatizada de avaliar *chatbots*, tanto em termos de sua modelagem quanto de seu desempenho, permitindo às equipes de desenvolvimento monitorar e melhorar seus sistemas de forma mais eficaz. A publicação do artigo *"Chatbot Design Understanding: a framework for automating chatbot modeling quality assessment"*, na conferência CISTI 2024 (19ª Conferência Ibérica de Sistemas e Tecnologias de Informação), corrobora com a conclusão sobre a relevância e a inovação desta pesquisa no campo da avaliação de *chatbots*.

Para o SERPRO, o *framework* DUBI traz uma evolução significativa para a plataforma Serprobots. A implementação de indicadores claros e padronizados de qualidade permitirá às equipes da empresa entregar soluções superiores aos seus clientes. Além disso, a automação proporcionada pelo DUBI poderá aumentar a produtividade nos projetos de desenvolvimento e curadoria de *chatbots*, permitindo uma maior capacidade produtiva da empresa e redução do custo operacional destes projetos, ambos aspectos relevantes para a administração pública.

Como perspectivas futuras, três caminhos promissores foram identificados. Primeiro, é necessário entender mais claramente quais métricas do módulo *Design Understanding* (DU) têm maior influência no desempenho do *chatbot*, avaliado pelo módulo *chatBot Intelligence* (BI). Entende-se que isso pode ser alcançado por meio de experimentos adicionais com uma maior variedade de assistentes virtuais, permitindo traçar correlações mais precisas entre as métricas dos dois módulos.

A segunda possibilidade de trabalho futuro é a evolução do DUBI para não apenas avaliar e indicar melhorias, mas também automatizar intervenções na modelagem dos *chatbots* com base nas recomendações geradas. Para isso, o uso de modelos de linguagem de grande porte (do inglês, LLMs — *Large Language Models*) pode ser explorado para gerar conteúdos de melhoria automaticamente, apresentando sugestões que podem ser aprovadas e implementadas pela equipe de curadoria.

E, finalmente, a terceira perspectiva para o futuro é a expansão do DUBI para incluir a capacidade de avaliar *chatbots* generativos. Para tal, será necessário aperfeiçoar os módulos de avaliação para considerarem a coerência e relevância das respostas geradas. Em particular, será importante observar a relevância das respostas em relação à pergunta recebida e ao contexto da conversa em andamento, assegurando que as interações se mantêm coesas e significativas. Esta proposta ampliará significativamente o escopo e a aplicabilidade do DUBI, atendendo às necessidades emergentes de avaliação de *chatbots* mais avançados.

Em resumo, com base nos testes realizados no escopo desta pesquisa, o *framework* DUBI se mostrou uma ferramenta eficaz e inovadora para a avaliação automática de *chatbots*, oferecendo tanto à comunidade científica quanto ao SERPRO um meio mensurável e objetivo de aferir e melhorar a qualidade e o desempenho desses sistemas. As contribuições deste trabalho não só avançam o estado da arte na avaliação de assistentes virtuais, mas também oferece uma solução prática que pode ser aplicada para otimizar o desenvolvimento e a curadoria desses sistemas no ambiente corporativo, com menos esforço e dependência de humanos especializados no tema. Por fim, os resultados obtidos fornecem fortes indícios de que o DUBI pode ser aplicado a outros *chatbots*, proporcionando um diagnóstico detalhado e confiável das suas capacidades e deficiências.

Referências

[1] Caldarini, Guendalina, Sardar Jaf e Kenneth McGarry: *A Literature Survey of Recent Advances in Chatbots*. Information, 13(1):41, janeiro 2022, ISSN 2078-2489. https://www.mdpi.com/2078-2489/13/1/41. 1, 2, 5, 9, 10, 32, 34, 35, 37, 38, 39, 40

[2] Motger, Quim, Xavier Franch e Jordi Marco: *Software-Based Dialogue Systems: Survey, Taxonomy, and Challenges*. ACM Computing Surveys, 55(5):1–42, maio 2023, ISSN 0360-0300. https://dl.acm.org/doi/10.1145/3527450. 1, 10, 35, 40

[3] Bavaresco, Rodrigo, Diórgenes Silveira, Eduardo Reis, Jorge Barbosa, Rodrigo Righi, Cristiano Costa, Rodolfo Antunes, Marcio Gomes, Clauter Gatti, Mariangela Vanzin, Saint Clair Junior, Elton Silva e Carlos Moreira: *Conversational agents in business: A systematic literature review and future research directions*. Computer Science Review, 36:100239, maio 2020, ISSN 15740137. https://linkinghub.elsevier.com/retrieve/pii/S1574013719303193. 1, 33

[4] Extra: *Uma em cada três vagas do serviço público federal está desocupada*, março 2023. https://tinyurl.com/2rkze5uk, (acesso em 24 Mar. 2023). 1

[5] Brasil: *Lei Nº 14.129, de 29 de março de 2021*. Diário Oficial da República Federativa do Brasil, 2021. http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2021/lei/l14129.htm. 1

[6] Serpro: *Serpro - Inovação que pulsa*, junho 2023. https://campanhas.serpro.gov.br/institucional/inovacao-que-pulsa/?utm_source=portal&utm_medium=banner&utm_campaign=inovacao-que-pulsa&utm_content=20230130-, (acesso em 22 Jun. 2023). 1

[7] Serpro: *Serprobots*, junho 2024. https://serprobots.ia.serpro.gov.br/, (acesso em 22 Jun. 2024). 2

[8] Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre e Mark Cieliebak: *Survey on evaluation methods for dialogue systems*. Artificial Intelligence Review, 54(1):755–810, janeiro 2021, ISSN 0269-2821. https://link.springer.com/10.1007/s10462-020-09866-x. 2, 4, 5, 18, 40, 46

[9] Mohammad Forkan, Abdur Rahim, Prem Prakash Jayaraman, Yong Bin Kang e Ahsan Morshed: *ECHO: A Tool for Empirical Evaluation Cloud Chatbots*. Em *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, páginas 669–672. IEEE, maio 2020, ISBN 978-1-7281-6095-5. https://ieeexplore.ieee.org/document/9139701/. 2, 15, 19, 20, 39, 43, 46, 47

[10] Bravo-Santos, Sergio, Esther Guerra e Juan de Lara: *Testing chatbots with charm*. Communications in Computer and Information Science, 1266 CCIS:426–438, 2020, ISSN 18650937. https://link.springer.com/chapter/10.1007/978-3-030-58793-2_34. 2, 3, 44, 47, 50

[11] Gao, Mingkun, Xiaotong Liu, Anbang Xu e Rama Akkiraju: *Chatbot or Chat-Blocker: Predicting Chatbot Popularity before Deployment*. Em *Designing Interactive Systems Conference 2021*, páginas 1458–1469, New York, NY, USA, junho 2021. ACM, ISBN 9781450384766. https://dl.acm.org/doi/10.1145/3461778.3462147. 2, 21, 22, 23, 38, 41, 46, 47, 54

[12] Cañizares, Pablo C., Sara Pérez-Soler, Esther Guerra e Juan De Lara: *Automating the measurement of heterogeneous chatbot designs*. Proceedings of the ACM Symposium on Applied Computing, páginas 1491–1498, abril 2022. https://dl.acm.org/doi/10.1145/3477314.3507255. 2, 3, 21, 22, 23, 24, 39, 40, 41, 42, 46, 47, 50, 54

[13] Yang, Ruolan, Zitong Li, Haifeng Tang e Kenny Q Zhu: *ChatMatch: Evaluating Chatbots by Autonomous Chat Tournaments*. Em *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, páginas 7579–7590. Long Papers, 2022. 2, 18, 19, 20, 21, 45, 47, 68

[14] IBM: *watsonx Assistant - IBM Cloud API Docs*. https://cloud.ibm.com/apidocs/assistant-v2, (acesso em 12 Jan. 2024). 3, 13, 22, 26, 34, 44, 50, 76, 78

[15] Google: *Dialogflow*. https://cloud.google.com/dialogflow?hl=pt-br, (acesso em 02 Abr. 2024). 3, 13, 22, 42, 44, 50

[16] Amazon Web Services: *Amazon Lex*. https://docs.aws.amazon.com/lex/index.html, (acesso em 10 Mar. 2024). 3, 13, 44, 50

[17] Rasa Technologies Inc: *RASA - Conversational AI Platform*, maio 2023. https://rasa.com/, (acesso em 05 Mai. 2023). 3, 14, 42, 50

[18] Meta Platforms: *WhatsApp*, junho 2023. https://www.whatsapp.com/?lang=pt_br, (acesso em 28 Jun. 2023). 3

[19] Meta Platforms: *Messenger*, junho 2023. https://www.messenger.com/, (acesso em 22 Jun. 2023). 3

[20] X Corp: *Direct Message (DM) on Twitter*, junho 2023. https://help.twitter.com/en/using-twitter/direct-messages, (acesso em 22 Jun. 2023). 3

[21] Deriu, Jan, Don Tuggener, Pius von Däniken, Jon Ander Campos, Alvaro Rodrigo, Thiziri Belkacem, Aitor Soroa, Eneko Agirre e Mark Cieliebak: *Spot The Bot: A Robust and Efficient Framework for the Evaluation of Conversational Dialogue Systems*. EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, páginas 3971–3984, 2020. https://aclanthology.org/2020.emnlp-main.326. 4, 17, 38, 43, 46, 47

[22] Maroengsit, Wari, Thanarath Piyakulpinyo, Korawat Phonyiam, Suporn Pongnumkul, Pimwadee Chaovalit e Thanaruk Theeramunkong: *A Survey on Evaluation Methods for Chatbots*. Em *Proceedings of the 2019 7th International Conference on Information and Education Technology*, volume Part F148391, páginas 111–119, New York, NY, USA, março 2019. ACM, ISBN 9781450366397. https://dl.acm.org/doi/10.1145/3323771.3323824. 5, 9, 10, 14, 17, 18, 35, 36, 39, 40

[23] Silva, João Quirino Machado e e Irene Pimenta Rodrigues: *Desenvolvimento de Chatbots para responder a perguntas frequentes*. Tese de Doutoramento, Universidade de Évora, janeiro 2021. http://hdl.handle.net/10174/29039. 5, 35, 36, 37

[24] Mohamad Suhaili, Sinarwati, Naomie Salim e Mohamad Nazim Jambli: *Service chatbots: A systematic review*. Expert Systems with Applications, 184:115461, dezembro 2021, ISSN 09574174. https://linkinghub.elsevier.com/retrieve/pii/S0957417421008745. 5, 10, 32, 35, 37

[25] Agarwal, Ritu e Mani Wadhwa: *Review of State-of-the-Art Design Techniques for Chatbots*. SN Computer Science, 1(5):246, setembro 2020, ISSN 2662-995X. https://link.springer.com/10.1007/s42979-020-00255-3. 5, 10, 35, 38

[26] Luo, Bei, Raymond Y. K. Lau, Chunping Li e Yain-Whar Si: *A critical review of state-of-the-art chatbot designs and applications*. WIREs Data Mining and Knowledge Discovery, 12(1):e1434, janeiro 2022, ISSN 1942-4787. https://onlinelibrary.wiley.com/doi/10.1002/widm.1434. 5, 10, 32, 35, 38

[27] Ma, Longxuan, Mingda Li, Wei Nan Zhang, Jiapeng Li e Ting Liu: *Unstructured Text Enhanced Open-Domain Dialogue System: A Systematic Survey*. ACM Transactions on Information Systems, 40(1):1–44, janeiro 2022, ISSN 1046-8188. https://dl.acm.org/doi/10.1145/3464377. 5, 35

[28] Agra, Ronaldo e Jacir Bordim: *Chatbot design understanding: a framework for automating chatbot modeling quality assessment*. Em *2024 19th Iberian Conference on Information Systems and Technologies (CISTI)*, Salamanca/ES, junho 2024. To appear. 6

[29] Eisenstein, Jacob: *Introduction to Natural Language Processing*. MIT Press, Cambridge, 2019. 10

[30] Chowdhary, K R: *Natural Language Processing*. Em *Fundamentals of Artificial Intelligence*, páginas 603–649. Springer India, New Delhi, 2020, ISBN 978-81-322-3972-7. https://doi.org/10.1007/978-81-322-3972-7_19. 10

[31] Nigam, Amber, Prashik Sahare e Kushagra Pandya: *Intent detection and slots prompt in a closed-domain chatbot*. Em *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, páginas 340–343, 2019. 10

[32] Adamopoulou, Eleni e Lefteris Moussiades: *An Overview of Chatbot Technology*. Em Maglogiannis, Ilias, Lazaros Iliadis e Elias Pimenidis (editores): *Artificial Intelligence Applications and Innovations*, páginas 373–383, Cham, 2020. Springer International Publishing, ISBN 978-3-030-49186-4. 11

[33] Chowdhary, K R: *Information Retrieval*. Em *Fundamentals of Artificial Intelligence*, páginas 557–602. Springer India, New Delhi, 2020, ISBN 978-81-322-3972-7. https://doi.org/10.1007/978-81-322-3972-7_18. 11

[34] Cao, Yihan, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu e Lichao Sun: *A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT*. J. ACM, 37(111), março 2023. https://arxiv.org/abs/2303.04226v1. 11, 62

[35] OpenAI: *Introducing ChatGPT*, maio 2023. https://openai.com/blog/chatgpt, (acesso em 05 Mai. 2023). 12

[36] Brabra, Hayet, Marcos Baez, Boualem Benatallah, Walid Gaaloul, Sara Bouguelia e Shayan Zamanirad: *Dialogue Management in Conversational Systems: A Review of Approaches, Challenges, and Opportunities*. IEEE Transactions on Cognitive and Developmental Systems, 14(3):783–798, setembro 2022, ISSN 2379-8920. https://ieeexplore.ieee.org/document/9447005/. 12, 35

[37] Casas, Jacky, Marc Olivier Tricot, Omar Abou Khaled, Elena Mugellini e Philippe Cudré-Mauroux: *Trends & methods in chatbot evaluation*. ICMI 2020 Companion - Companion Publication of the 2020 International Conference on Multimodal Interaction, páginas 280–286, outubro 2020. https://dl.acm.org/doi/10.1145/3395035.3425319. 14, 17

[38] Wikipedia: *Confusion matrix*. https://en.wikipedia.org/wiki/Confusion_matrix, (acesso em 02 Mai. 2023). 15

[39] Jebb, Andrew T, Vincent Ng e Louis Tay: *A Review of Key Likert Scale Development Advances: 1995–2019*. Frontiers in Psychology, 12, 2021, ISSN 1664-1078. https://www.frontiersin.org/articles/10.3389/fpsyg.2021.637547. 17

[40] Finch, Sarah E. e Jinho D. Choi: *Towards unified dialogue system evaluation: A comprehensive analysis of current evaluation protocols*. Em Pietquin, Olivier, Smaranda Muresan, Vivian Chen, Casey Kennington, David Vandyke, Nina Dethlefs, Koji Inoue, Erik Ekstedt e Stefan Ultes (editores): *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, páginas 236–245, 1st virtual meeting, julho 2020. Association for Computational Linguistics. https://aclanthology.org/2020.sigdial-1.29. 18, 21, 39, 41

[41] Jurafsky, Daniel e James H. Martin: *N-gram Language Models*. Em *Speech and Language Processing*, capítulo 3, páginas 31–57. Stanford University Press, 3rd edição, janeiro 2023. 18

[42] Wikipedia: *Flesch–Kincaid readability tests*, maio 2023. https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests, (acesso em 02 Mai. 2023). 18

[43] Wikipedia: *Indice Gulpease*, maio 2023. https://it.wikipedia.org/wiki/Indice_Gulpease, (acesso em 05 Mai. 2023). 18

[44] Moreno, Gleice Carvalho de Lima, Marco P. M. de Souza, Nelson Hein e Adriana Kroenke Hein: *ALT: um software para análise de legibilidade de textos em Língua Portuguesa*. Policromias-Revista de Estudos do Discurso, Imagem e Som, 8:91–128, 2022, ISSN 2448-2935. 19, 78

[45] Qiu, Lisong, Juntao Li, Wei Bi, Dongyan Zhao e Rui Yan: *Are Training Samples Correlated? Learning to Generate Dialogue Responses with Multiple References*. ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, páginas 3826–3835, 2019. https://aclanthology.org/P19-1372. 21

[46] Lin, Zhaojiang, Andrea Madotto, Jamin Shin, Peng Xu e Pascale Fung: *MoEL: Mixture of Empathetic Listeners*. EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, páginas 121–132, 2019. https://aclanthology.org/D19-1012. 21

[47] Moore, Robert J e Raphael Arar: *Conversational UX Design: A Practitioner's Guide to the Natural Conversation Framework*. Association for Computing Machinery, New York, NY, USA, 2019, ISBN 9781450363013. 23

[48] RiveScript: *Artificial Intelligence Scripting Language - RiveScript.com*. https://www.rivescript.com/, (acesso em 10 Abr. 2024). 26, 37

[49] OpenAI: *OpenAI*, março 2023. https://openai.com/, (acesso em 02 Mar. 2023). 26, 34, 63

[50] CAPES: *Portal periodicos CAPES*, maio 2023. https://www-periodicos-capes-gov-br.ez54.periodicos.capes.gov.br/, (acesso em 05 Mai. 2023). 31

[51] Clarivate: *Coleção principal da Web of Science*. https://www.webofscience.com/wos/woscc/basic-search, (acesso em 02 Mai. 2023). 31

[52] Institute of Electrical and Electronics Engineers: *IEEE Xplore digital library*, maio 2023. https://ieeexplore.ieee.org/Xplore/home.jsp, (acesso em 05 Mai. 2023). 31

[53] Association for Computing Machinery: *ACM Digital Library*, maio 2023. https://dl.acm.org/. 31

[54] Safi, Zeineb, Alaa Abd-Alrazaq, Mohamed Khalifa e Mowafa Househ: *Technical Aspects of Developing Chatbots for Medical Applications: Scoping Review*. J Med Internet Res 2020;22(12):e19127 https://www.jmir.org/2020/12/e19127, 22(12):e19127, dezembro 2020, ISSN 14388871. https://www.jmir.org/2020/12/e19127. 33

[55] Følstad, Asbjørn, Theo Araujo, Effie Lai Chong Law, Petter Bae Brandtzaeg, Symeon Papadopoulos, Lea Reis, Marcos Baez, Guy Laban, Patrick McAllister, Carolin Ischen, Rebecca Wald, Fabio Catania, Raphael Meyer von Wolff, Sebastian Hobert e Ewa Luger: *Future directions for chatbot research: an interdisciplinary*

research agenda. Computing, 103(12):2915–2942, dezembro 2021, ISSN 14365057. https://link.springer.com/article/10.1007/s00607-021-01016-7. 33

[56] Motger, Quim, Xavier Franch e Jordi Marco: *Conversational Agents in Software Engineering: Survey, Taxonomy and Challenges*, junho 2021. http://arxiv.org/abs/2106.10901. 33

[57] Weizenbaum, Joseph: *ELIZA a computer program for the study of natural language communication between man and machine*. Communications of the ACM, 26(1):23–28, janeiro 1983, ISSN 15577317. https://dl.acm.org/doi/10.1145/357980.357991. 33

[58] Wallace, Richard S.: *The anatomy of A.L.I.C.E.* Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer, páginas 181–210, 2009. https://link.springer.com/chapter/10.1007/978-1-4020-6710-5_13. 33

[59] AIML Foundation: *AIML Foundation*. http://www.aiml.foundation/, (acesso em 07 Abr. 2024). 34

[60] Apple: *Siri - Apple (BR)*, abril 2023. https://www.apple.com/br/siri/, (acesso em 28 Abr. 2023). 34

[61] Google: *Google Assistant, your own personal Google default*, março 2023. https://assistant.google.com/, (acesso em 02 Mar. 2023). 34

[62] Amazon: *Amazon Alexa Official Site: What is Alexa?* https://developer.amazon.com/pt-BR/alexa, (acesso em 07 Mar. 2023). 34

[63] Radford, Alec e Karthik Narasimhan: *Improving Language Understanding by Generative Pre-Training*, 2018. https://api.semanticscholar.org/CorpusID:49313245. 34

[64] OpenAI: *GPT-4 Technical Report*, março 2023. https://arxiv.org/abs/2303.08774v3, (acesso em 02 Mar. 2023). 34

[65] Du, Nan, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen e Claire Cui: *GLaM: Efficient scaling of language models with mixture-of-experts*. Em Chaudhuri, Kamalika, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu e Sivan Sabato (editores): *Proceedings of the 39th International Conference on Machine Learning*, volume 162 de *Proceedings of Machine Learning Research*, páginas 5547–5569. PMLR, 17–23 Jul 2022. https://proceedings.mlr.press/v162/du22c.html. 34

[66] Thoppilan, Romal, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee Huaixiu, Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping

Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel, Morris Tulsee, Doshi Renelito, Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi e Quoc Le Google: *LaMDA: Language Models for Dialog Applications*, janeiro 2022. https://arxiv.org/abs/2201.08239v3. 34

[67] Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave e Guillaume Lample: *LLaMA: Open and Efficient Foundation Language Models*, fevereiro 2023. https://arxiv.org/abs/2302.13971v1. 34

[68] Hussain, Shafquat, Omid Ameri Sianaki e Nedal Ababneh: *A Survey on Conversational Agents/Chatbots Classification and Design Techniques*. Advances in Intelligent Systems and Computing, 927:946–956, 2019, ISSN 21945365. https://link.springer.com/chapter/10.1007/978-3-030-15035-8_93. 35

[69] Tai, André Gilberto, Maria Luísa Torres, Ribeiro Marques e Silva Coheur: *PALbot: a Plug&(Almost)pLay chatbot*. Tese de Doutoramento, Instituto Universitário de Lisboa, julho 2019. https://repositorio.iscte-iul.pt/handle/10071/19933. 35

[70] Almansor, Ebtesam H. e Farookh Khadeer Hussain: *Survey on Intelligent Chatbots: State-of-the-Art and Future Research Directions*. Advances in Intelligent Systems and Computing, 993:534–543, 2020, ISSN 21945365. https://link.springer.com/chapter/10.1007/978-3-030-22354-0_47. 35

[71] Ni, Jinjie, Tom Young, Vlad Pandelea, Fuzhao Xue e Erik Cambria: *Recent advances in deep learning based dialogue systems: a systematic survey*. Artificial Intelligence Review, páginas 1–101, agosto 2022, ISSN 0269-2821. https://link.springer.com/10.1007/s10462-022-10248-8. 35

[72] Silva, Wallinson de Lima e Eiji Adachi Medeiros Barbosa: *Uso de testes metamórficos para verificação de aplicação chatbot*. Tese de Doutoramento, Universidade Federal do Rio Grande do Norte, julho 2022. https://repositorio.ufrn.br/handle/123456789/49527. 35, 36

[73] Gupta, Pranav, Anand A Rajasekar, Amisha Patel, Mandar Kulkarni, Alexander Sunell, Kyung Hyuk Kim, Ganapathy Krishnan e Anusua Trivedi: *Answerability: A custom metric for evaluating chatbot performance*. Em *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, páginas 316–325, 2022. https://aclanthology.org/2022.gem-1.27. 40

[74] Deriu, Jan e Mark Cieliebak: *Towards a Metric for Automated Conversational Dialogue System Evaluation and Improvement*. INLG 2019 - 12th International Conference on Natural Language Generation, Proceedings of the Conference, páginas 432–437, 2019. https://aclanthology.org/W19-8654. 42, 46, 47

[75] Benesty, Jacob, Jingdong Chen, Yiteng Huang e Israel Cohen: *Pearson Correlation Coefficient*. Em *Noise Reduction in Speech Processing*, páginas 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, ISBN 978-3-642-00296-0. https://doi.org/10.1007/978-3-642-00296-0_5. 43

[76] Wikipedia: *Coeficiente de correlação de postos de Spearman*. https://pt.wikipedia.org/wiki/Coeficiente_de_correla%C3%A7%C3%A3o_de_postos_de_Spearman, (acesso em 02 Mai. 2023). 43

[77] Li, Yanran, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, Shuzi Niu e Hong Kong: *DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset*. Em Kondrak, Greg e Taro Watanabe (editores): *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, páginas 986–995, Taipei, Taiwan, novembro 2017. Asian Federation of Natural Language Processing. https://aclanthology.org/I17-1099. 44

[78] Solaiman, Irene, Miles Brundage, Openai Jack, Clark Openai, Amanda Askell Openai, Ariel Herbert-Voss, Jeff Wu Openai, Alec Radford Openai, Gretchen Krueger Openai, Jong Wook, Kim Openai, Sarah Kreps, Miles Mccain Politiwatch, Alex Newhouse, Jason Blazakis, Kris Mcguffie e Jasmine Wang: *Release Strategies and the Social Impacts of Language Models*, agosto 2019. https://arxiv.org/abs/1908.09203v2. 44

[79] Zhan, Jingtao, Jiaxin Mao, Yiqun Liu, Min Zhang e Shaoping Ma: *An Analysis of BERT in Document Ranking*. SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, páginas 1941–1944, julho 2020. https://dl.acm.org/doi/10.1145/3397271.3401325. 44

[80] Wikipedia: *Kronecker delta*, julho 2023. https://en.wikipedia.org/wiki/Kronecker_delta, (acesso em 05 Jul. 2023). 57

[81] Di Gennaro, Giovanni, Amedeo Buonanno e Francesco AN Palmieri: *Considerations about learning word2vec*. The Journal of Supercomputing, páginas 1–16, 2021, ISSN 1573-0484. https://doi.org/10.1007/s11227-021-03743-2. 59

[82] Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jéssica Rodrigues e Sandra Aluísio: *Portuguese word embeddings: Evaluating on word analogies and natural language tasks*. Em *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, páginas 122–131, Porto Alegre, RS, Brasil, 2017. SBC. https://sol.sbc.org.br/index.php/stil/article/view/4008. 59

[83] Myers, Glenford J, Corey Sandler e Tom Badgett: *The art of software testing*. John Wiley & Sons, Hoboken and N.J, 3rd ed edição, 2012. 66

[84] Kim, Jintae, Shinhyeok Oh, Oh Woog Kwon e Harksoo Kim: *Multi-turn chatbot based on query-context attentions and dual wasserstein generative adversarial networks.* Applied Sciences, 9:3908, setembro 2019. 76

[85] Haque, Sazzadul: *A question-answering machine learning system for faqs.* Tese de Mestrado, Universidade de Évora, 2021. 78, 117

[86] Real, Livy, Erick Fonseca e Hugo Goncalo Oliveira: *The assin 2 shared task: a quick overview.* Em *International Conference on Computational Processing of the Portuguese Language*, páginas 406–412. Springer, 2020. 116

[87] Pradhan, Nitesh, Manasi Gyanchandani, Rajesh Wadhvani *et al.*: *A review on text similarity technique used in ir and its application.* International Journal of Computer Applications, 120(9):29–34, 2015. 117

[88] Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jéssica Rodrigues e Sandra Aluísio: *Portuguese word embeddings: Evaluating on word analogies and natural language tasks.* Em *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, páginas 122–131, Porto Alegre, RS, Brasil, 2017. SBC. `https://sol.sbc.org.br/index.php/stil/article/view/4008`. 117

[89] SentenceTransformers: *all-MiniLM-L6-v2*, abril 2024. `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`, (acesso em 21 Abr. 2024). 117

[90] Souza, Fábio, Rodrigo Nogueira e Roberto Lotufo: *BERTimbau: pretrained BERT models for Brazilian Portuguese.* Em *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020. 117

[91] Ali Abd Al-Hameed, Khawla: *Spearman's correlation coefficient in statistical analysis.* International Journal of Nonlinear Analysis and Applications, 13(1):3249–3255, 2022, ISSN 2008-6822. `https://ijnaa.semnan.ac.ir/article_6079.html`. 118

Apêndice A

Modelo do relatório da avaliação estática do módulo DU

Este apêndice tem como finalidade apresentar o modelo criado para o relatório de avaliação estática do módulo *Design Understanding* (DU) do *framework* DUBI, desenvolvido como parte integrante do presente trabalho.

Conforme detalhado na Seção 4.3, o módulo DU avalia a estrutura e o conteúdo de treinamento dos *chatbots*. Para isso, um conjunto de 16 métricas são observadas e aferidas, com base nos aspectos relacionados à estrutura, ao relacionamento e às características textuais dos componentes do assistente virtual. Essas métricas são apresentadas na Tabela 4.1 deste documento.

Desta forma, o relatório de avaliação do módulo *Design Understanding* foi pensado para poder apresentar uma visão ampla e detalhada da qualidade do conteúdo estático dos *chatbots*. Por ser parte de um processo automatizado, decidiu-se por utilizar o formato *JavaScript Object Notation* (JSON) para estruturar este relatório, de modo a possibilitar sua interpretação por sistemas de computador.

Assim, o relatório está organizado em seções distintas, cada uma representando uma das métricas avaliadas pelo módulo DU. Essa estrutura possibilita uma análise sistemática e abrangente do assistente virtual avaliado. A Tabela A.1 ilustra o conteúdo do documento gerado no formato JSON e contém todas as seções do modelo do relatório de avaliação estática, acompanhadas de seus atributos, tipos e descrições.

Tabela A.1: Descrição das seções e atributos do relatório DU.

Seção	Atributo	Tipo	Descrição
general_info	*chatbot*	Alfanumérico	Identificação do chatbot.
	num_intents	Inteiro	Número total de intenções presentes no chatbot.
	num_training_examples	Inteiro	Número total de exemplos de treinamento no chatbot.
	avg_training_examples_per_intent	Real	Média de exemplos de treinamento por intenção.
	num_dialog_nodes	Inteiro	Número total de nós de diálogo no chatbot.
qet	*recommended_min_examples*	Inteiro	Número mínimo esperado de exemplos de treinamento por intenção.
	recommended_max_examples	Inteiro	Número máximo esperado de exemplos de treinamento por intenção.
	inappropriate_intent_rate	Real	Percentual de intenções inadequadas em relação ao total.
	inappropriate_intents	Lista	Nomes das intenções inadequadas.
bet	*imbalance_threshold*	Real	Limite aceitável de desequilíbrio.
	imbalanced_intents_rate	Real	Percentual de intenções desequilibradas em relação ao total.
	imbalanced_intents	Lista	Intenções desequilibradas com seus respectivos atributos: - *intent* (Identificação da intenção). - *imbalance* (Valor de desbalanceamento da intenção).
sid	*similarity_threshold*	Real	Limiar de similaridade para detecção de intenções similares.
	similar_intents_rate	Real	Percentual de intenções similares em relação ao total.
	similar_intents	Lista	Intenções similares com seus respectivos atributos: - *intent* (Identificação da intenção). - *similar_intents* (Dicionário contendo as intenções similares e suas similaridades).
tet	*min_words_per_training_examples*	Inteiro	Número mínimo esperado de palavras por exemplo de treinamento.
	max_words_per_training_examples	Inteiro	Número máximo esperado de palavras por exemplo de treinamento.
	inadequate_intents_rate	Real	Percentual de intenções com exemplos inadequados em relação ao total.
	inappropriate_intents	Lista	Intenções com exemplos inadequados e seus respectivos atributos: - *intent* (Identificação da intenção). - *examples_few_words* (Número de exemplos de treinamento com poucas palavras). - *examples_many_words* (Número de exemplos de treinamento com muitas palavras).
let	*idx_min_expected_readability*	Inteiro	Índice mínimo esperado de legibilidade.
	desc_min_expected_readability	Alfanumérico	Descrição do índice mínimo esperado de legibilidade.
	inadequate_intents_rate	Real	Percentual de intenções com legibilidade inadequadas.
	inappropriate_intents	Lista	Intenções com respostas inadequadas e seus respectivos atributos: - *intent* (Identificação da intenção). - *idx_readability* (Valor do índice de legibilidade). - *desc_readability* (Descrição do índice de legibilidade).
ret	*min_expected_representativeness*	Real	Valor mínimo esperado de representatividade.
	intents_low_representativeness_rate	Real	Percentual de intenções com representatividade baixa em relação ao total.
	intents_low_representativeness	Lista	Intenções com representatividade baixa e seus respectivos atributos: - *intent* (Identificação da intenção). - *examples_low_representativeness* (Exemplos com baixa representatividade).
qio	*unlinked_intents_rate*	Real	Percentual de intenções não vinculadas a nenhum nó de diálogo.
	unlinked_intents	Lista	Intenções não vinculadas a nenhum nó de diálogo.
ttr	*max_characters_answer*	Inteiro	Tamanho máximo de texto aceito para as respostas do chatbot.
	inadequate_answers_rate	Real	Percentual de respostas inadequadas em relação ao total.
	inappropriate_answers	Lista	Respostas inadequadas com seus respectivos atributos: - *dialog_node_id* (Identificação do nó de diálogo da resposta inadequada). - *dialog_node_title* (Título do nó de diálogo da resposta inadequada). - *text_size* (Tamanho do texto da resposta inadequada).
ltr	*idx_min_expected_readability*	Inteiro	Índice mínimo esperado de legibilidade.
	desc_min_expected_readability	Alfanumérico	Descrição do índice mínimo esperado de legibilidade.
	inadequate_answers_rate	Real	Percentual de respostas inadequadas em relação ao total.
	inappropriate_answers	Lista	Respostas inadequadas com seus respectivos atributos: - *answer_text* (Título do nó de diálogo da resposta inadequada). - *idx_readability* (Valor do índice de legibilidade). - *desc_readability* (Descrição do índice de legibilidade).
rtr	*min_expected_representativeness*	Real	Valor mínimo esperado de representatividade.
	dialogs_low_representativeness_rate	Real	Percentual de respostas com representatividade baixa em relação ao total.
	answers_low_representativeness	Lista	Respostas com representatividade baixa e seus respectivos atributos: - *answer_text* (Título do nó de diálogo com baixa representatividade). - *representativeness_score* (Valor do índice de representatividade).
str	*inadequate_dialog_nodes_rate*	Real	Percentual de nós de diálogo com sentimento negativo.
	inadequate_dialog_nodes	Lista	Nós de diálogo com sentimento negativo.
qid	*inadequate_dialog_nodes_rate*	Real	Percentual de nós de diálogo com a mesma condição de entrada.
	inadequate_dialog_nodes	Lista	Nós de diálogo com a mesma condição de entrada.
qdv	*inadequate_dialog_nodes_rate*	Real	Percentual de nós de diálogo com condição de entrada verdadeira.
	inadequate_dialog_nodes	Lista	Nós de diálogo com condição de entrada verdadeira.
qdf	*inadequate_dialog_nodes_rate*	Real	Percentual de nós de diálogo com condição de entrada falsa.
	inadequate_dialog_nodes	Lista	Nós de diálogo com condição de entrada falsa.
qdr	*inadequate_dialog_nodes_rate*	Real	Percentual de nós de diálogo com respostas vazias.
	inadequate_dialog_nodes	Lista	Nós de diálogo com respostas vazias.
qcf	*amount_detected_cycles*	Inteiro	Número de ciclos detectados.
	detected_cycles	Lista	Descrição dos caminhos que formam ciclos.

Apêndice B

Modelo do relatório da avaliação interativa do módulo BI

O objetivo deste apêndice é apresentar a estrutura do relatório de avaliação interativa do módulo *chatBot Intelligence* (BI) do *framework* DUBI, desenvolvido como parte integrante do presente trabalho.

Conforme detalhado na Seção 4.4, o módulo BI é responsável por gerar mensagens simuladas e interagir com o *chatbot* em avaliação. Com base nisso, ele consegue coletar e aferir as métricas de desempenho, qualidade de diálogo e cobertura dos testes, como especificado na Seção 4.4.3 deste documento.

Desta forma, o relatório de execução do módulo BI contém todas as informações essenciais para avaliar o desempenho e a qualidade do assistente virtual sob teste, fornecendo dados valiosos para o aprimoramento contínuo do sistema. Assim, como o relatório do módulo DU, aqui também será utilizado o formato *JavaScript Object Notation* (JSON) para estruturar as informações, de modo a possibilitar sua interpretação computacional.

Com isso, o relatório do *chatBot Intelligence* está organizado em seções que representam as classes de métricas aferidas. Essa estrutura possibilita uma análise objetiva do desempenho do *chatbot* e da efetividade do teste realizado. A Tabela B.1 apresenta a estrutura do relatório de avaliação interativa, detalhando as seções e seus respectivos atributos, tipos e descrições.

Tabela B.1: Descrição das seções e atributos do relatório BI.

Seção	**Atributo**	**Tipo**	**Descrição**
-	*chatbot*	Alfanumérico	Identificação do chatbot.
performance_metrics	*accuracy*	Real	Acurácia do *chatbot* medida no teste.
	precision	Real	Precisão do *chatbot* medida no teste.
	recall	Real	A revocação do *chatbot* medida no teste.
	f1score	Real	O F1-score do *chatbot* medido no teste.
dialog_metrics	tmR	Real	O tempo médio de resposta, em segundos, do *chatbot*.
	tf	Real	A taxa de *fallback* do *chatbot* para o teste.
	tc	Real	A taxa de compreensão do *chatbot* aferida no teste.
	tcr	Real	A taxa de consistência das respostas do *chatbot* no teste.
	tamb	Real	A taxa de ambiguidade do *chatbot* aferida no teste.
test_coverage	*intents*	Real	Percentual de cobertura do teste quanto às intenções do *chatbot*.
	dialog_nodes	Real	Percentual de cobertura do teste em relação aos nós de diálogo.
failed_interactions	-	Lista	As unidades de teste que falharam, cada uma contendo:
	id	Alfanumérico	Identificação da unidade de teste que falhou.
	type	Inteiro	O tipo de mensagem simulada.
	binding_intent	Alfanumérico	A identificação da intenção testada.
	simulated_message	Alfanumérico	A mensagem simulada utilizada no teste.
	expected_answer	Alfanumérico	A resposta esperada para esta interação.
	received_answer	Alfanumérico	A resposta recebida do *chatbot*.
	response_time	Real	O tempo de resposta, em segundo, desta interação.

Apêndice C

Experimento para avaliar técnicas de similaridade de textos

Este apêndice apresenta o experimento conduzido para explorar e determinar a técnica mais adequada para calcular a similaridade entre textos curtos. O objetivo foi estabelecer qual o método seria utilizado para calcular a métrica SID (Similaridade entre intenções distintas), usada pelo módulo *Design Understanding* (DU) do *framework* DUBI, que mais se correlacionasse com a avaliação humana.

Para isso, o experimento foi estruturado em etapas que possibilitassem obter e preparar os dados a serem utilizados, bem como calcular as similaridades e analisar a correlação com o julgamento humano. Desta forma, a metodologia deste experimento é assim resumida: (i) inicialmente, foi necessário identificar e obter uma base de pares de textos previamente anotadas por humanos, em relação a similaridade entre eles; (ii) na sequência, foram selecionadas diferentes técnicas para avaliar a similaridade dos textos, conforme apresentado na Seção C.2; (iii) em seguida, as similaridades dos textos foram calculadas utilizando cada uma das técnicas; (iv) uma análise de correlação entre os valores obtidos e as avaliações humanas foi efetuada; (v) por fim, a técnica com a maior correlação foi então identificada como a mais adequada para integrar o *framework* DUBI.

C.1 Base de dados utilizada

A base de dados "ASSIN 2" [86] é uma referência no contexto da linguística computacional, particularmente na avaliação de similaridade semântica e inferência textual em língua portuguesa. Ela consiste em um conjunto de pares de sentenças anotadas por avaliadores humanos, que forneceram medidas de similaridade semântica entre as sentenças. Essas medidas são categorizadas de forma que cada par de sentenças é acompanhado por uma pontuação que reflete o grau de similaridade percebido pelos avaliadores, possibilitando

uma comparação objetiva com as métricas calculadas automaticamente pelas técnicas computacionais.

Esta base é composta por três conjuntos de dados, divididos em arquivos distintos para representar os dados de treino, validação e teste. Juntos, estes conjuntos de dados somam 9.448 pares de textos curtos. Cada par de textos do conjunto de dados é rotulado com o valor de similaridade, com escala contínua de 1 a 5, sendo:

1. Sentenças completamente diferentes, sobre diferentes temas;
2. Sentenças não relacionadas, mas sobre o mesmo tema;
3. Sentenças de certa forma relacionadas: podem descrever fatos diferentes, mas compartilham de alguns detalhes;
4. Sentenças fortemente relacionadas, mas que diferem em alguns detalhes;
5. Sentenças que significam essencialmente a mesma coisa.

Como o objetivo deste experimento não era o de criar um modelo específico de cálculo de similaridade, mas sim ter valores de referência rotulados por humanos para comparar com os valores calculados computacionalmente, os conjuntos de dados de treino, validação e testes foram agrupados num único conjunto, e utilizado no experimento.

C.2 Técnicas testadas

As técnicas de cálculo de similaridade testadas incluíram tanto o método de distância entre textos quanto a distância de cosseno, a partir de vetorização dos textos [85]. Para o primeiro, foram utilizadas as distâncias de Levenshtein e Jaro [87]. Já para o segundo, os textos foram vetorizados utilizando TF-IDF, Word2Vec e BERT *embeddings*.

Para a vetorização a partir da abordagem Word2Vec, utilizou-se modelos de 300 e de 600 dimensões treinados em português [88]. Já a vetorização via arquitetura BERT foi realizada com uso dos modelos "all-MiniLM-L6-v2" [89], um modelo multi-idiomas que gera vetores densos de 384 dimensões, e "BERTimbau Base" [90], um modelo treinado com textos exclusivamente em português.

C.3 Processo de avaliação e resultados

Cada uma das técnicas mencionadas anteriormente foi aplicada aos pares de textos da base "ASSIN 2", resultando em um conjunto de valores de similaridade. Como resultado desta etapa, foi gerado um arquivo JSON contendo, para cada par de texto, os respectivos valores de similaridade de cada técnica, conforme exemplificado na Listagem C.1.

Listagem C.1: Trecho do JSON gerado após o cálculo de similaridade.


```
{
 "id": 6,
 "source": "train",
 "similarity_human": 4.8,
 "similarity_levenshtein": 0.5833,
 "similarity_jaro": 0.7681,
 "similarity_tfidf": 0.7327,
 "similarity_word2vec_300": 0.9368,
 "similarity_word2vec_600": 0.9224,
 "similarity_all-MiniLM-L6-v2": 0.9078,
 "similarity_bertimbau": 0.5764,
 "text1": "surfista esta pegando grande onda da agua verde escura",
 "text2": "onda grande esta sendo surfada surfista da agua verde escura"
}
```


Na fase subsequente, procedeu-se à segregação dos valores de similaridade em conjuntos distintos, correspondentes a cada uma das técnicas avaliadas, incluindo a avaliação humana oriunda da base "ASSIN 2". O objetivo desta separação dos valores em diferentes conjuntos foi verificar a distribuição dos dados de cada conjunto. Nesta verificação observou-se que nenhum dos conjuntos de dados apresentou distribuição normal.

A análise de correlação foi então realizada para comparar os valores de similaridade gerados pelas técnicas computacionais com aqueles anotados por avaliadores humanos. Devido à ausência de normalidade nos dados, optou-se pela utilização do coeficiente de correlação de Spearman [91], um método não-paramétrico apropriado para tais condições.

Como resultado, os valores de correlação obtidos foram os seguintes, em comparação com a avaliação humana:

- Distância de Levenshtein: 0.4283;
- Distância de Jaro: 0.4383;
- TF-IDF: 0.5859;
- Word2Vec 300 dimensões: 0.6035;
- **Word2Vec 600 dimensões: 0.6047**;
- BERT all-MiniLM-L6-v2: 0.6031
- BERTimbau: 0.3218

A análise indicou que a técnica de distância de cosseno com vetorização a partir do modelo Word2Vec de 600 dimensões, treinado em português, alcançou a maior correlação

com as avaliações humanas. Outras técnicas de vetorização também exibiram valores próximos, mas as abordagens baseadas em distâncias textuais revelaram-se menos correlatas, evidenciando a eficácia das técnicas baseadas em vetores para a análise de similaridade semântica em textos curtos.

Curiosamente, a técnica que utilizou o modelo BERTimbau para vetorização apresentou uma correlação significativamente mais baixa do que as demais técnicas, incluindo as abordagens de distância textual. Ao avaliar exemplos dos casos em que houve uma alta similaridade segundo a avaliação humana, mas baixa segundo o modelo BERTimbau, identificou-se uma tendência de discrepância nos casos de alteração da voz ativa para a passiva entre os pares de textos, como ilustrado nos textos "text1" e "text2" da Listagem C.1. Nestas situações, as similaridades calculadas com base nos vetores do BERTimbau frequentemente divergiram da avaliação humana e das outras técnicas de vetorização. Embora a exploração dessa inconsistência não fosse o foco do experimento, o padrão observado foi suficiente para explicar o desempenho inferior deste modelo.

C.4 Conclusão do experimento

A técnica de distância de cosseno utilizando o modelo Word2Vec, treinado em português e com 600 dimensões, foi identificada como a mais adequada para o cálculo da métrica SID do *framework* DUBI. Este método se mostrou não só próximo à avaliação humana, mas também eficiente do ponto de vista computacional, exigindo menos recursos para execução e apresentando melhores tempos de resposta. Por este motivo, decidiu-se por calcular a similaridade de textos na métrica SID fazendo uso da desta técnica. Por fim, destaca-se que os resultados obtidos são específicos para o contexto deste trabalho, não sendo necessariamente aplicáveis a outros cenários, que podem requerer adaptações ou novas avaliações.

Apêndice D

Detalhamento dos relatórios gerados no experimento de validação do DUBI

O presente apêndice tem como propósito fornecer o detalhamento dos relatórios das avaliações estáticas e interativas gerados durante o experimento de validação do *framework* DUBI. Eles complementam os resultados sumarizados e discutidos no Capítulo 6 desta dissertação, os quais ratificam a eficácia do DUBI no processo de avaliação da qualidade de *chatbots*.

É importante destacar que, apesar dos assistentes virtuais utilizados como objetos do estudo não possuírem dados pessoais, seus conteúdos continham referências a organizações, órgãos de governo, endereços de URL, cidades, dentre outros. Mesmo sendo dados não considerados sensíveis, optou-se por omiti-los para garantir a privacidade dos projetos, uma vez que são informações irrelevantes para a apresentação dos resultados. Assim, parte dos textos dos exemplos de treinamento das intenções, nomes de intenções, respostas dos nós de diálogo e mensagens de testes geradas pelo DUBI podem aparecer com o termo "OPP", significando *"omitido por privacidade"*. Cabe ressaltar, no entanto, que essa modificação foi realizada apenas para esta apresentação. Durante o experimento, todas as informações foram utilizadas como originalmente modeladas pelas equipes de desenvolvimento.

Destaca-se ainda que essa modificação se deu apenas nos textos, sem qualquer alteração nos valores das métricas aferidas, seja pela avaliação estática ou interativa. Assim, a integridade e a fidelidade dos resultados obtidos durante o experimento foram preservadas.

Para facilitar a consulta e o acesso aos relatórios, assim como a legibilidade deste documento, optou-se por disponibilizá-los em um repositório do GitHub, através do endereço https://github.com/ronaldoagra/dubi-experimento-relatorios.

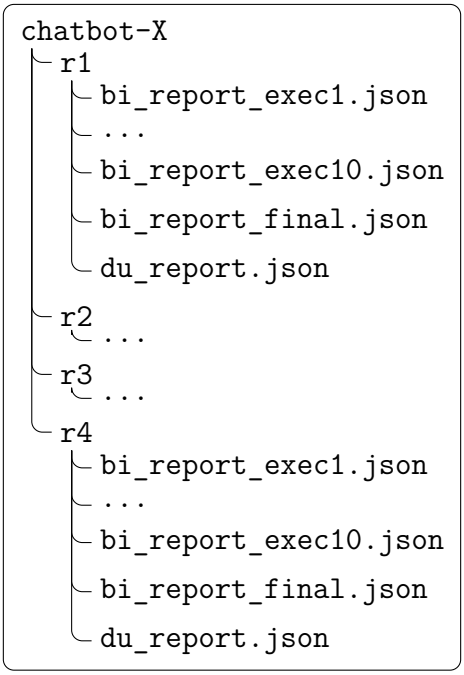


Figura D.1: Exemplo da estrutura de diretórios e arquivos para os relatórios de avaliação de um *chatbot*.

Este repositório está estruturado para apresentar os relatórios conforme os *chatbots* utilizados no experimento, disponibilizados, para cada assistente virtual, os relatórios das avaliações estáticas e interativas segmentados pelas rodadas de avaliação, conforme ilustrado na Figura D.1. Assim, dentro de cada diretório de rodada de avaliação (representados na figura por "r1", "r2", "r3" e "r4"), encontram-se três tipos de arquivos:

- **"du_report.json":** representa o relatório de avaliação estática produzido pelo módulo *Design Understanding* (DU), seguindo as especificações do Apêndice A.
- **"bi_report_execN.json":** são os relatórios da avaliação interativa, produzidos pelo módulo *chatBot Intelligence* (BI) e que seguem a estrutura apresentada no Apêndice B. Estes arquivos possuem o sufixo 'execN', onde 'N' é um número de 1 a 10, indicando a iteração específica da avaliação interativa.
- **"bi_report_final.json":** trata-se de um relatório contendo os resultados agregados das 10 iterações da avaliação interativa, naquela rodada de avaliação. A estrutura deste arquivo é semelhante à do relatório BI de uma interação, entretanto apresentando os valores médios, desvios padrão e intervalos de confiança para cada uma das métricas.

Desta maneira, acredita-se que esta organização dos arquivos facilita o acesso aos detalhes de cada etapa do processo de avaliação conduzido no experimento. Dentro de

cada diretório, é possível analisar tanto o relatório do módulo *Design Understanding* (DU), que apresenta suas métricas e indicações de melhoria pertinentes, quanto os indicadores de desempenho gerados pelo módulo *chatBot Intelligence* (BI), disponíveis nos relatórios da avaliação interativa de cada uma das execuções realizadas. Esses relatórios, portanto, complementam a análise apresentada no Capítulo 6, permitindo uma compreensão mais profunda dos resultados obtidos durante o experimento realizado para validar a viabilidade do *framework* DUBI.