Um chatbot treinado para entender intenções de estudantes em diálogos sobre teste de software

Thiago Musico[1], **Silvana Morita Melo**[2],
Leo Natan Paschoal[3], **Simone do Rocio Senger de Souza**[1]

[1] Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, (USP) - São Carlos - SP - Brasil

[2] Universidade Federal da Grande Dourados, (UFGD) - Dourados – Mato Grosso do Sul - Brasil

[3] Pontifícia Universidade Católica do Paraná, (PUC-PR) - Curitiba - Paraná - Brasil

thiago.musico50@gmail.com, silvanamelo@ufgd.edu.br

leo.paschoal@pucpr.br, srocio@icmc.usp.br

Abstract. *This paper presents the evolution of TOB-STT, a chatbot designed to support the software testing education. The chatbot assists students in resolving doubts, illustrating concepts, and demonstrating the application of testing criteria for generating test cases. The initial version, based on pattern matching, was promising but showed limitations in terms of response precision and understanding students' intentions. Seeking to improve the quality of interactions and the effectiveness of educational support, its architecture was redesigned using natural language processing and natural language understanding techniques, resulting in a new version named TOB-STT 2.0. To evaluate it, an experimental study was conducted between the two versions. The results showed that the new version was considerably more adequate in correctly addressing students' intentions. These findings indicate that the chatbot's evolution has successfully improved its comprehension capabilities and, consequently, the quality of educational support it provides.*

Resumo. *Este artigo apresenta a evolução do TOB-STT, um chatbot desenvolvido para apoiar o ensino de teste de software. O chatbot auxilia os estudantes a sanar dúvidas, exemplificar conceitos e demonstrar a aplicação de critérios de teste para a geração de casos de teste. A versão inicial, baseada em casamento de padrões, embora promissora, revelou limitações na precisão das respostas e na compreensão das intenções dos estudantes. Com o propósito de aprimorar a qualidade das interações e a eficácia do suporte educacional, sua arquitetura foi reformulada com técnicas de processamento de linguagem natural e entendimento de linguagem natural, resultando em uma nova versão, denominada TOB-STT 2.0. Para avaliá-la, foi conduzido um estudo experimental entre as duas versões. Os resultados demonstraram que a nova versão foi consideravelmente mais eficaz em responder corretamente às intenções dos estudantes. Essas evidências indicam que a evolução do chatbot foi bem-sucedida em aprimorar sua capacidade de compreensão e, consequentemente, a qualidade do suporte educacional oferecido.*

1. Introdução

Os chatbots, ou agentes conversacionais, são sistemas de software com potencial para auxiliar estudantes em diversas atividades educacionais [Roca et al. 2024, Kuhail et al. 2023]. Eles podem ser aplicados para esclarecer dúvidas sobre conteúdos, fornecer feedback durante sessões de tutoria e até mesmo contribuir para o desenvolvimento de habilidades linguísticas. Uma das principais vantagens da utilização desses

DOI: 10.5753/sbie.2025.12704

sistemas na educação é a sua disponibilidade ininterrupta, que oferece suporte aos alunos 24 horas por dia, sete dias por semana[Nee et al. 2023]. Devido a essa característica, os chatbots têm sido amplamente explorados no ensino de diversas áreas do conhecimento, como matemática [Lee and Yeo 2022], línguas [Mohamed 2024] e computação [Paschoal et al. 2018, Groothuijsen et al. 2024, Haldar et al. 2025].

Na área de computação, especificamente no contexto da engenharia de software, o chatbot TOB-STT foi projetado para apoiar estudantes no aprendizado de teste de software [Paschoal et al. 2019, Paschoal et al. 2023]. O objetivo do TOB-STT é, por meio de diálogos em linguagem natural na língua inglesa, explicar conceitos de teste de software, fornecer exemplos testes e demonstrar a geração de casos de teste utilizando técnicas específicas (*e.g.*, teste funcional e teste estrutural). Contudo, apesar dos resultados promissores de um estudo de viabilidade, a primeira versão do TOB-STT (*i.e.*, TOB-STT 1.0), implementada com uma abordagem baseada em casamento de padrões (*pattern matching*), apresentou limitações. Os resultados de um experimento controlado revelaram que o chatbot tinha dificuldades em compreender determinadas mensagens dos alunos [Paschoal et al. 2023]. Esse problema de interpretação limita sua capacidade de fornecer respostas precisas e identificar adequadamente as perguntas formuladas — uma limitação inerente ao casamento de padrões [Adamopoulou and Moussiades 2020].

Considerando os avanços em Processamento de Linguagem Natural *(Natural Language Processing* - NLP) e Compreensão da Linguagem Natural (*Natural Language Understanding*- NLU) [Attigeri et al. 2024], aliados ao potencial do TOB-STT como mecanismo de apoio ao ensino, o estudo de [Paschoal et al. 2023] sugeriu como trabalho futuro a evolução do TOB-STT. Nesse sentido, esta pesquisa parte, então, da premissa de que um modelo de inteligência artificial treinado para compreender as intenções dos estudantes pode aprimorar tanto a capacidade de interpretação do chatbot quanto a qualidade de suas respostas. Assim, o foco deste trabalho é a reformulação da arquitetura do TOB-STT para torná-lo mais eficaz na identificação das dúvidas e na geração de respostas adequadas.

O objetivo deste artigo é, portanto, relatar os resultados de um estudo que visou aprimorar a qualidade das respostas do TOB-STT, reduzindo a ocorrência de respostas incorretas. Para isso, propõe-se uma nova arquitetura que utiliza reconhecimento de entidades e intenções para aprimorar o modelo de classificação de perguntas, com o intuito de aumentar a precisão na identificação da pergunta feita pelo aluno e na adequação das respostas. Adicionalmente, o artigo apresenta um estudo experimental conduzido para avaliar se a nova versão do chatbot oferece um suporte de maior qualidade aos estudantes em comparação com a versão anterior.

Este trabalho contribui não apenas para a melhoria de um chatbot específico para o ensino de teste de software, mas também oferece um caminho para outros pesquisadores. Embora o domínio seja específico, acredita-se que a metodologia adotada pode ser útil para evoluir outros chatbots baseados em casamento de padrões que enfrentam problemas similares. Espera-se que este estudo sirva como uma referência para estabelecer chatbots de domínio mais eficazes, capazes de acolher melhor os estudantes e atender às suas necessidades.

2. Materiais e métodos

Esta seção apresenta a metodologia utilizada neste estudo, dividida em duas partes principais. A primeira descreve o processo de desenvolvimento da nova versão do chatbot, o TOB-STT 2.0. A segunda retrata o planejamento e a execução do experimento controlado conduzido para avaliar e comparar a nova versão com sua antecessora.

2.1. Desenvolvimento do chatbot

O desenvolvimento do TOB-STT 2.0 foi conduzido em cinco etapas principais, detalhadas a seguir.

Seleção da plataforma e arquitetura

A primeira decisão no processo de evolução do chatbot foi substituir a técnica de casamento de padrões, que utilizava a linguagem AIML (*Artificial Intelligence Markup Language*), por uma abordagem baseada em PLN e NLU. Após análise de diferentes tecnologias apresentadas no estudo de [Leifheit et al. 2023], o framework Rasa[1] foi escolhido para a construção do TOB-STT 2.0. A escolha foi fundamentada tanto em estudos que indicam o bom desempenho do framework Rasa quanto por sua adaptabilidade e implementação mais simples em comparação com o desenvolvimento de modelos de redes neurais recorrentes e LSTM (*Long Short-Term Memory*) [Pérez-Soler et al. 2020, Pérez-Soler et al. 2021]. O framework Rasa é dividido em dois componentes principais: o Rasa NLU, responsável por interpretar a mensagem do usuário e extrair suas intenções e entidades, e o Rasa Core, que gerencia o diálogo e decide a próxima ação do chatbot.

Mapeamento de intenções e entidades

Para migrar o conhecimento do chatbot para a nova arquitetura, foi realizada uma análise manual da base de conhecimento AIML do TOB-STT 1.0. O objetivo foi identificar e estruturar a base de conhecimento em intenções (o propósito do usuário) e entidades (informações-chave na mensagem). Essa etapa foi fundamental porque a arquitetura original, baseada em AIML, opera por casamento de padrões textuais, não possuindo uma estrutura explícita de intenções. Em contraste, o modelo de NLU do framework Rasa precisa ser treinado com exemplos de frases e suas respectivas intenções e entidades para aprender a classificar novas entradas.

A atividade foi conduzida de forma sistemática, com os pesquisadores revisando as bases de conhecimento do TOB-STT 1.0 para inferir as intenções subjacentes, utilizando como guia as categorias de perguntas já existentes (definição, demonstração e exemplos), retratadas no estudo de [Paschoal et al. 2019]. Essa analise resultou na definição de três intenções principais: (1) *define* (definir um conceito), (2) *how_to_use* (entender a aplicação de uma técnica) e (3) *cite_example* (solicitar um exemplo). Para gerenciar o fluxo do diálogo, também foram implementadas interações conversacionais como (i) *greet* (saudação), (ii) *thank* (agradecimento) e (iii) *bye* (despedida).

Para a estruturação das entidades, todos os assuntos de teste de software foram consolidados sob um único tipo de entidade principal, denominada *concept*. Nessa abordagem, análoga ao paradigma Orientado a Objetos, a entidade *concept* funciona como uma "classe", enquanto os termos específicos (*e.g.*, "*fault*" ou "*functional testing*") são tratados como "instâncias". Por exemplo, nas perguntas "*What is a fault?*" e "*What is functional testing?*", as palavras "*fault*" e "*functional testing*" são identificadas como valores distintos da mesma entidade *concept*.

Essa decisão de projeto possibilitou organizar o domínio de conhecimento de forma coesa. Por fim, a relação entre cada entidade e suas intenções correspondentes foi mapeada em toda a base de conhecimento do chatbot, conforme exemplificado parcialmente na Tabela 1, gerando os dados necessários para o treinamento da nova versão do chatbot.

[1]Mais informações disponíveis em: `https://rasa.com/`.

Tabela 1. Exemplo de mapeamento de entidades e intenções

Entidade	**Intenção**		
	Conceitualizar	Mostrar como usar	Fornecer exemplos
all nodes	✓	✓	
all edges	✓	✓	
all paths	✓		
cyclomatic complexity	✓	✓	
control flow graph	✓	✓	
functional testing	✓	✓	✓
equivalence partitioning	✓	✓	✓
boundary value	✓	✓	✓

Desenvolvimento do modelo de diálogo

Com as intenções e entidades definidas, a etapa seguinte foi o desenvolvimento dos dados de treinamento para o Rasa Core, o componente responsável por gerenciar o diálogo. Para determinar a melhor abordagem para essa estrutura, foi conduzido um estudo para avaliar três estratégias de implementação. Conforme ilustrado na Figura 1, este estudo seguiu três etapas: (1) treinamento do classificador (*rasa train*); (2) teste interativo por meio de uma interface de linha de comando (*rasa shell*); e (3) validação manual das respostas do chatbot.

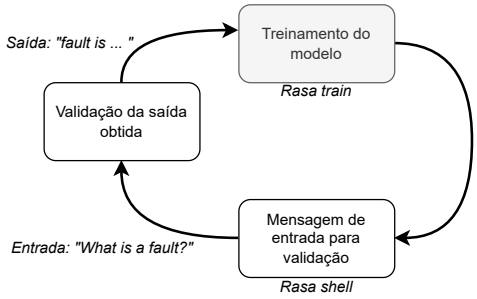

Figura 1. Fluxo de desenvolvimento e validação do modelo classificador

As três estratégias de implementação formuladas e avaliadas foram:

- A primeira estratégia envolveu o uso de estória padrão do framework Rasa. Uma estória é um exemplo de diálogo que serve como dado de treinamento. Contudo, essa abordagem falhou porque o modelo aprendeu a responder com base apenas no tipo da entidade (*concept*), sendo incapaz de diferenciar valor específico. Por exemplo, nas perguntas "*What is all nodes?*"e "*What is all paths?*", embora o chatbot identificasse corretamente as entidades, ele fornecia a mesma resposta para ambas as perguntas, pois não conseguia diferenciar os valores "*all nodes*" e "*all paths*".
- A segunda estratégia propunha o uso de um método em Python para centralizar a busca por todas as respostas. Após o Rasa NLU identificar a intenção e a entidade (e.g., *define* e *fault*), o Rasa Core acionaria essa função, que consultaria uma base de dados externa e devolveria a resposta. A estratégia foi descartada porque a implementação de um método genérico se mostrou complexa, pois a abordagem mais viável exigiria a criação de uma função em Python para cada resposta possível.
- A terceira estratégia, ilustrada na Figura 2, consistiu no uso de *slots*, que são unidades de memória que armazenam informações ao longo da conversa. Nessa

implementação, quando o chatbot identifica uma entidade (e.g., "*fault*"), ele armazena esse valor em um *slot*. Isso permite que o chatbot mantenha o contexto e responda corretamente a perguntas subsequentes que não contenham a entidade explicita. Por exemplo, após perguntar "*What is a fault?*", o usuário pode dizer "*Give me an example of that*", e o chatbot, ao consultar o *slot*, saberá que "*that*" se refere a "*fault*". Essa abordagem foi a que melhor se adequou aos requisitos do projeto.

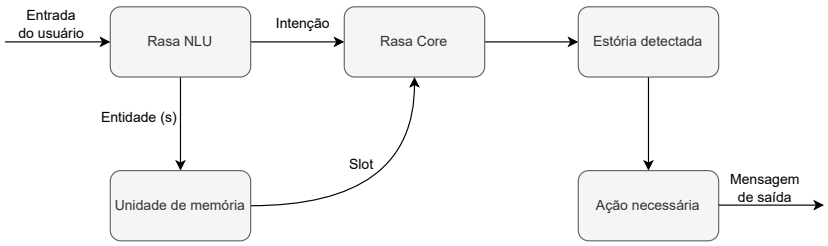

Figura 2. Estratégia de implementação com slots

Configuração da pipeline de NLU e treinamento

A etapa seguinte foi a configuração da *pipeline* de processamento do Rasa NLU e o treinamento do modelo. Essa configuração foi necessário porque o Rasa, sendo uma plataforma customizável, exige a definição explícita de como o modelo deve processar a linguagem e aprender com os dados.

A *pipeline*, ilustrada na Figura 3, é composta por uma sequência de componentes que realizam tarefas como tokenização e extração de características. Ressalta-se que, neste projeto, optou-se por utilizar configurações pré-definidas pelo próprio framework Rasa. O único componente não utilizado foi o *ResponseSelector*, pois ele é projetado para chatbots simples de pergunta-e-resposta, enquanto o TOB-STT 2.0 foi concebido para ter uma lógica de diálogo mais complexa e contextual. A *pipeline* final culmina no *DIETClassifier*, componente que efetivamente realiza a classificação da intenção e a extração da entidade.

O treinamento do chatbot foi realizado a partir de um conjunto de dados estruturados em três arquivos principais:

- Dados de NLU: O arquivo contém exemplos de frases anotadas com suas respectivas intenções e entidades. Estes dados, extraídos da base de conhecimento da primeira versão do chatbot, treinam o Rasa NLU para compreender o que o estudante diz ou pergunta.
- Estórias de diálogo: O arquivo contém exemplos de conversas que mapeiam sequências de intenções a ações do chatbot, treinando o Rasa Core para gerenciar o fluxo do diálogo.
- Arquivos do domínio: O arquivo funciona como um manifesto, listando todas as intenções, entidades, *slots* e ações que o chatbot conhece, conectando os componentes NLU e Core.

Ao final do treinamento, o framework Rasa gera um único arquivo de modelo compactado que encapsula os modelos NLU e Core treinados.

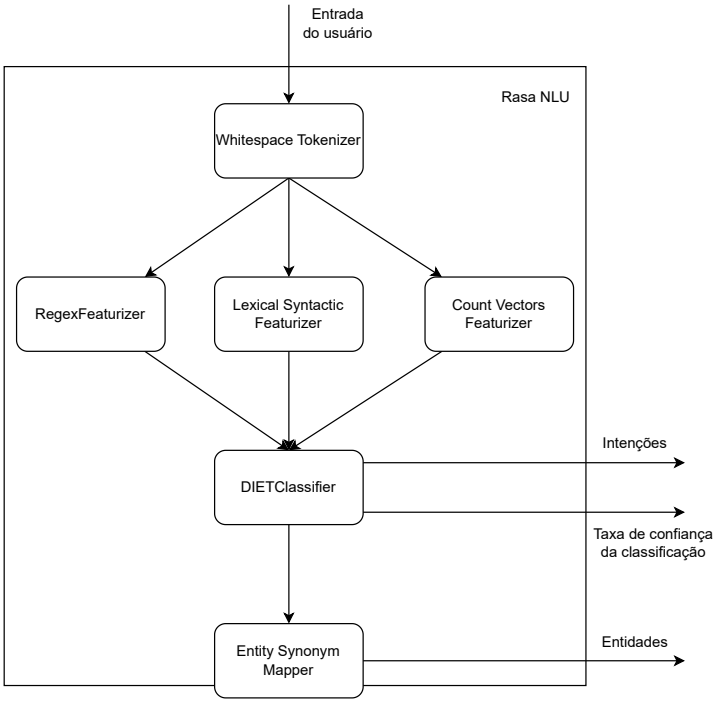

Figura 3. Pipeline de processamento do framework Rasa

Interface do chatbot

Após o treinamento do modelo, foi necessário disponibilizar o chatbot em uma interface gráfica. Para tanto, a plataforma de mensagens instantâneas Telegram[2] foi escolhida como a interface do TOB-STT 2.0. A decisão foi baseada em dois fatores principais: o framework Rasa oferece bibliotecas que facilitam a integração com o Telegram e a própria plataforma possui poucos pré-requisitos técnicos para a implementação de chatbots. A Figura 4 apresenta a interface do TOB-STT 2.0.

Figura 4. Interface do chatbot TOB-STT 2.0

[2]Mais informações disponíveis em: `https://web.telegram.org/`.

2.2. Planejamento do experimento

Para avaliar se a nova versão do TOB-STT fornece respostas de maior qualidade, foi planejado e conduzido um estudo experimental com o propósito de comparar o TOB-STT 2.0 com o TOB-STT 1.0. O experimento foi planejado seguindo o processo de [Wohlin et al. 2012] e suas etapas são detalhadas nas seções seguintes.

Definição do escopo

Conforme a abordagem GQM (*Goal*, *Question*, *Metric*) [Basili et al. 1994], o escopo do experimento pode ser descrito da seguinte forma: analisar o TOB-STT 2.0, com o propósito de verificar, com respeito a qualidade das respostas fornecidas pelo chatbot e a percepção dos alunos durante a interação, sob a perspectiva dos pesquisadores, no contexto de estudantes de um curso de graduação em Sistemas de Informação, matriculados na disciplina de Verificação, Validação e Teste de Software.

Para guiar esta investigação, foram formuladas duas questões de pesquisa (QP):

1. QP 1: "A nova versão do chatbot (TOB-STT 2.0), que utiliza NLP e NLU, fornece respostas mais corretas às perguntas dos alunos do que a versão baseada em casamento de padrões? "
   Esta questão visa em medir objetivamente a qualidade das respostas fornecidas pelo chatbot. A questão deu origem a um par de hipóteses estatísticas — a hipótese nula (H0) e a alternativa (HA) — que são detalhadas na Tabela 2.

Tabela 2. Questão de pesquisa e hipótese do experimento

Hipótese nula	Hipótese alternativa
H0: Não há diferença significativa na qualidade das respostas fornecidas pelo chatbot TOB-STT 2.0 em comparação com a versão baseada em casamento de padrões.	HA: Há diferença significativa na qualidade das respostas fornecidas pelo chatbot TOB-STT 2.0 em comparação com a versão baseada em casamento de padrões.

2. QP2: "Os alunos percebem a interação com o TOB-STT 2.0 como mais satisfatória e eficaz para o seu aprendizado em comparação com a versão baseada em casamento de padrões?"
   Esta questão visa compreender a experiência subjetiva dos estudantes.

Seleção das variáveis

Após a definição do escopo, as variáveis do estudo foram identificadas e definidas. O experimento contempla variáveis independentes e dependentes. A variável independente deste estudo é o tipo de classificador de diálogo, que corresponde a abordagem técnica utilizada pelo chatbot para interpretar e responder às mensagens dos usuários [Paschoal 2024]. Neste experimento, essa variável recebe dois tratamentos:

- TOB-SST 1.0: Versão inicial do chatbot de apoio ao ensino de teste de software, cujo classificador é baseado em casamento de padrões com a linguagem AIML.
- TOB-STT 2.0: Nova versão do chatbot, desenvolvida com o framework Rasa, cujo classificador utiliza Rasa NLU para compreensão da linguagem e Rasa Core para o gerenciamento do diálogo.

O objetivo do experimento e investigar os efeitos desses tratamentos e estratégias de desenvolvimento em duas variáveis dependentes, são elas:

- Qualidade da resposta: refere-se à adequação de cada resposta fornecida pelo chatbot a uma pergunta específica. Para operacionalizar essa variável, cada par pergunta-resposta é analisado por um juiz humano e classificado em uma das três categorias a seguir, com base no trabalho de [AbuShawar and Atwell 2016]:
  - Resposta correta: o estudante pergunta: "*What is a bug?*" e o chatbot responde corretamente: "*A bug is an error in a software program that causes it to produce an incorrect or unexpected result*".
  - Resposta incorreta: o chatbot não compreendeu corretamente a pergunta feita pelo estudante e, por isso, não conseguiu fornecer uma resposta apropriada, respondendo de maneira inadequada à interação. Por exemplo, o estudante pergunta: "*What is a bug?*" e o chatbot responde: "*A functional testing technique is used to verify specific system functionalities*".
  - Sem resposta: o chatbot não conseguiu responder à pergunta feita pelo estudante e, consequentemente, deixou o estudante sem retorno ou emitiu uma mensagem indicando que não sabia como responder à questão. Por exemplo, o estudante pergunta: "*What is a bug?*" e o chatbot responde: "*I'm sorry, I don't know the answer to that question*".
- Satisfação dos usuários: refere-se à percepção subjetiva do estudante sobre a experiência de interação com o chatbot. Para operacionalizar essa variável, utilizou-se um questionário composto por nove assertivas, proposto por [Herpich et al. 2020], que avalia diferentes dimensões da interação, como a adequação das respostas, a utilidade para a tarefa educacional e a satisfação geral. As assertivas são listadas a seguir:
  1. Ao interagir com o chatbot pela primeira vez, a experiência não foi animadora.
  2. As respostas fornecidas pelo chatbot foram sobre o tópico questionado.
  3. O chatbot não soube responder alguma pergunta realizada.
  4. Ao utilizar o chatbot, eu consegui obter o conhecimento pretendido.
  5. O chatbot não forneceu informações confiáveis em suas respostas.
  6. O chatbot contribuiu para a realização da tarefa.
  7. O chatbot demorou para fornecer as respostas.
  8. O chatbot possui uma interface fácil de usar.
  9. Eu fiquei insatisfeito com o chatbot.

Os participantes avaliaram cada assertiva em uma escala Likert variando de 1 (discordo totalmente) a 5 (concordo totalmente).

Amostragem e Design do experimento

O estudo experimental exigiu a seleção de estudantes para interagir com o chatbot e realizar uma atividade de teste de software. Para isso, foi necessário definir a amostragem. Por conveniência, foram selecionados alunos de graduação do Instituto De Ciências Matemáticas e de Computação da Universidade de São Paulo, matriculados na disciplina de Verificação, Validação e Teste de Software.

Optou-se por um design experimental independente (*i.e., between-subjects design*), no qual os estudantes são selecionados para participar do estudo e distribuídos aleatoriamente em um dos dois grupos de tratamentos, e cada grupo interage exclusivamente com a versão do chatbot que lhe é atribuída.

2.2.1. Instrumentos para a condução

Para a condução do experimento, foi preparado um conjunto de instrumentos com o objetivo de guiar os participantes, coletar os dados e viabilizar a análise posterior.

O principal objeto de estudo, a nova versão do chatbot (TOB-STT 2.0), foi desenvolvido e integrado ao Telegram para o acesso dos participantes. Para garantir que os estudantes utilizassem a ferramenta adequadamente, foi elaborado um conjunto de slides que explicava as funcionalidades, as limitações e a forma de interagir com o chatbot. Este material foi destinado ao grupo de usuários do TOB-STT 2.0 (grupo experimental). De forma análoga, para os alunos que utilizariam o TOB-STT 1.0 (grupo controle), também foi preparado um conjunto de slides contendo as informações necessárias para o acesso e uso do chatbot.

Para o experimento, foi estabelecida uma atividade que exigia dos alunos o uso dos chatbots como mecanismo de apoio à resolução de dúvidas sobre o conteúdo. Foi definida, então, uma tarefa educacional na qual alguns conceitos sobre teste de software eram apresentados, e os participantes precisavam identificar se estavam corretos ou não. Esta atividade foi elaborada com o propósito de estimular os alunos a se envolverem em diálogos com os chatbots.

Também foi elaborado um Termo de Consentimento Livre e Esclarecido (TCLE), que continha detalhes sobre o estudo (*i.e.*, objetivo, detalhes de execução) e tinha a finalidade de esclarecer aos alunos todos os possíveis benefícios, riscos e procedimentos a serem realizados. A elaboração deste documento foi necessária para garantir que os participantes recebessem todas as informações pertinentes à pesquisa antes de consentirem em participar.

Para a coleta dos dados, foram preparados dois instrumentos principais. O primeiro, um formulário online criado com o Google Forms[3], possuía a dupla função de receber a resolução da atividade educacional e medir a satisfação dos usuários por meio das 9 assertivas. O segundo instrumento foi um *script* em Python, desenvolvido para consultar e extrair os logs de diálogo da base de dados, permitindo a análise posterior da qualidade das respostas.

Todos os materiais e dados brutos do experimento, foram disponibilizados publicamente no pacote de laboratório, disponível no repositório Zenodo[4] por meio do seguinte link: `https://doi.org/10.5281/zenodo.16974205`.

Condução do experimento

O experimento foi executado durante uma aula da disciplina de Verificação, Validação e Teste de Software, ministrada por uma das pesquisadoras envolvidas neste estudo. Na ocasião, a proposta da pesquisa foi apresentada à turma e os alunos foram convidados a participar. No período, 42 alunos estavam matriculados na disciplina, e a proposta do estudo foi apresentado à turma. A participação foi voluntária, conforme explicitado no TCLE, sendo garantido que qualquer aluno poderia se recusar a participar ou desistir a qualquer momento, sem sofrer qualquer tipo de prejuízo.

Após a apresentação, os 42 alunos que consentiram em participar leram e concordaram com os termos do TCLE, sendo então divididos aleatoriamente em dois grupos. O primeiro grupo recebeu acesso ao TOB-STT 1.0, enquanto o segundo grupo foi direcionado a utilizar o TOB-STT 2.0.

A sessão experimental teve a duração total de 1 hora e 10 minutos. No início, os participantes de ambos os grupos foram instruídos a realizar, individualmente, uma atividade educacional. A principal orientação foi que, para solucionar quaisquer dúvidas sobre conceitos, definições ou exemplos necessários para a tarefa, eles deveriam utilizar o chatbot que lhes foi atribuído como única ferramenta de consulta.

[3]Mais informações disponíveis em: `https://docs.google.com/forms/u/0/`.
[4]Mais informações disponíveis em: `https://zenodo.org/`.

Ao completarem a tarefa, os participantes submeteram suas resoluções e registraram suas percepções sobre o chatbot por meio do formulário online.

Cuidados éticos

Este estudo foi conduzido em conformidade com os princípios éticos aplicáveis à pesquisa envolvendo seres humanos, conforme estabelecido pela Resolução 466/2012 do Conselho Nacional de Saúde. Todos os procedimentos foram rigorosamente adotados para garantir a proteção e o bem-estar dos participantes.

Antes do início da pesquisa, todos os participantes foram minuciosamente informados sobre os objetivos do estudo, a metodologia, os possíveis riscos e benefícios, bem como a garantia da confidencialidade de seus dados. Após receberem todas as informações e terem suas dúvidas esclarecidas, os participantes assinaram um TCLE, formalizando sua participação voluntária.

Para assegurar a privacidade e a confidencialidade, a identidade de todos os participantes foi preservada através da anonimização dos dados. As informações foram coletadas e armazenadas de forma segura, com acesso restrito apenas à equipe de pesquisa. Todos os resultados apresentados neste artigo são agregados, garantindo que nenhum participante possa ser individualmente identificado.

3. Resultados e discussões

Nesta seção, são apresentados os resultados obtidos a partir da execução do experimento. A exposição dos achados do estudo inicia-se com a descrição do processo de análise dos dados, seguida pela apresentação dos resultados referentes à qualidade das respostas e à satisfação dos estudantes.

3.1. Processo de análise de dados

A análise dos dados coletados foi realizada em duas frentes principais. Primeiramente, os *logs* dos diálogos, extraídos da base de dados, foram analisados manualmente por um especialista externo à execução do experimento, a fim de evitar vieses na classificação das interações. Cada par conversacional (pergunta do estudante e resposta do chatbot) foi avaliado e classificado em uma das três categorias pré-definidas: (1) resposta correta, (2) resposta incorreta e (3) sem resposta. Adicionalmente, as respostas dos estudantes ao formulário online foram compiladas e agregadas para permitir a comparação da percepção sobre o TOB-STT 2.0 e o TOB-STT 1.0.

3.2. Qualidade das respostas

Conforme ilustra a Figura 5, o TOB-STT 2.0 demonstrou um desempenho superior aos responder às dúvidas dos estudantes. A proporção de respontas corretas aumentou de 77% (TOB-STT 1.0) para 85.3% (TOB-STT 2.0). Complementarmente, houve uma redução na ocorrência de respostas inadequadas: a categoria de respostas incorretas caiu de 14% para 9.8%, e a de ausência de resposta diminuiu de 9% para 4.9% na nova versão.

Para verificar se essa diferença é estatisticamente significativa, foi executado o teste Qui-quadrado para os dados nominais categóricos do estudo. O resultado do teste indicou um *p-value* de 0.050. Adotando um nível de significância de $\alpha = 0.1$, o resultado sugere que a diferença observada é significativa. Com base nisso, a hipótese nula para a primeira questão de pesquisa do experimento é rejeitada, havendo evidências que suportam a hipótese alternativa: a nova versão do chatbot (TOB-STT 2.0) fornece respostas com qualidade superior em comparação com a versão baseada em casamento de padrões.

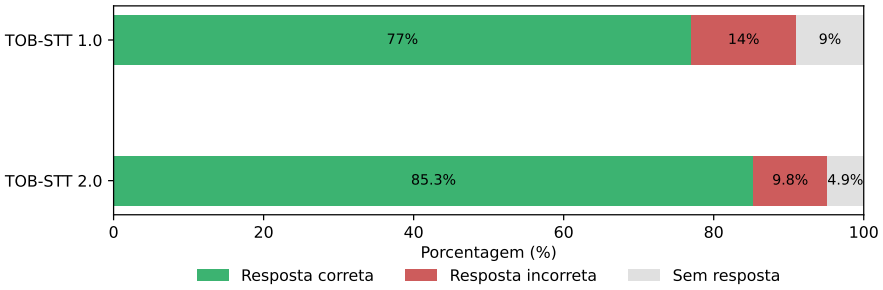

Figura 5. Comparativo da qualidade das respostas entre as versões do chatbot

3.3. Satisfação dos estudantes

A análise do formulário de satisfação indicou uma percepção majoritariamente mais positiva para o TOB-STT 2.0 em comparação com a versão anterior, conforme evidenciado pela análise comparativa das 9 assertivas apresentada na Figura 6. Os participantes relataram ter tido uma primeira impressão mais animadora com a nova versão (Assertiva 1) e demonstraram maior sucesso em obter o conhecimento pretendido (Assertiva 4). A relevância das respostas também foi um ponto forte: na Assertiva 2, houve uma percepção unânime (100% de concordância) de que as respostas do TOB-STT 2.0 foram sobre o tópico questionado. Além disso, o chatbot foi avaliado como mais útil para a realização da tarefa (Assertiva 6) e falhou menos em responder às perguntas (Assertiva 3), o que sugere uma melhor capacidade de interpretação por parte do TOB-STT 2.0.

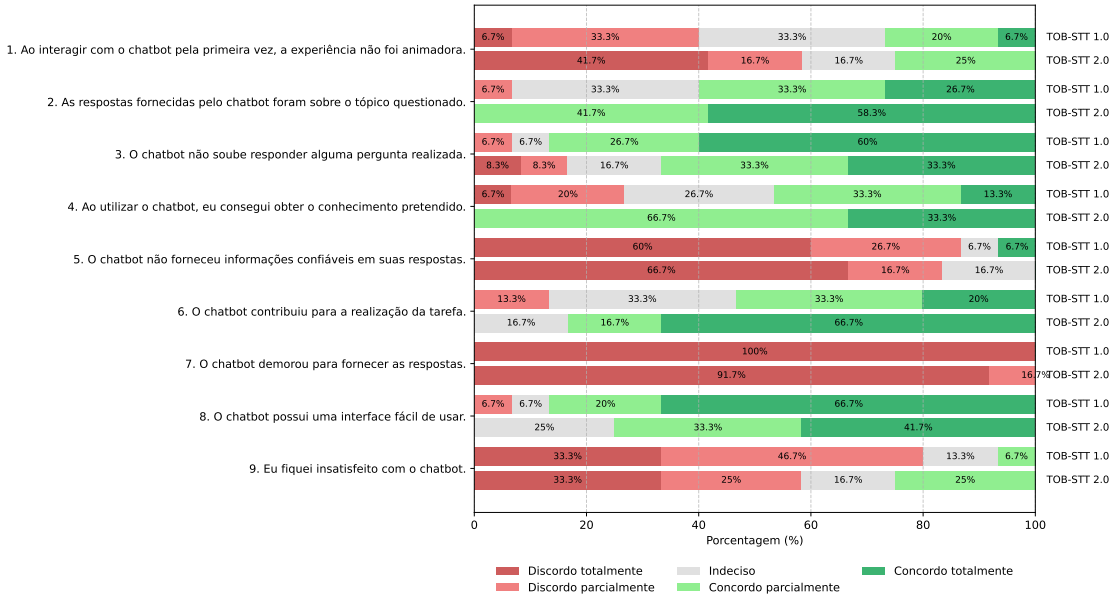

Figura 6. Satisfação dos estudantes com os chatbots

Quanto ao tempo de resposta percebido (Assertiva 7), ambas as versões do chatbot foram avaliadas positivamente pelos estudantes, conforme ilustrado na Figura 6. No que se refere à confiança na qualidade das respostas (Assertiva 5), os dois chatbots foram avaliados de forma similar, com os participantes considerando-as confiáveis em ambos os casos. Contudo, o principal ponto desfavorável para o TOB-STT 2.0 foi sua interface de

interação (Assertiva 8); os dados sugerem que a interface web da versão 1.0 foi percebida como mais simples de utilizar do que a do Telegram. Este resultado contrariou as expectativas iniciais, visto que os estudantes são usuários frequentes do Telegram e estão familiarizados com seus elementos de interface. Essa percepção pode ter influenciado o resultado da satisfação geral (Assertiva 9), que, embora majoritariamente positiva para o TOB-STT 2.0, apresentou uma parcela de insatisfação, possivelmente atribuída mais à plataforma de acesso do que à capacidade do chatbot em si.

4. Conclusões

Este artigo apresentou um estudo sobre a evolução do chatbot educacional TOB-STT, desenvolvido para sanar uma limitação central de sua versão inicial: a dificuldade em compreender as intenções dos estudantes devido a uma arquitetura baseada em casamento de padrões. O objetivo principal foi desenvolver e avaliar uma nova versão, o TOB-STT 2.0, cuja arquitetura foi reformulada com técnicas de PLN e NLU. A premissa era que essa nova abordagem aprimoraria a capacidade de interpretação do chatbot, resultando em respostas mais adequadas e, consequentemente, em um suporte educacional de maior qualidade. Este objetivo foi alcançado por meio do desenvolvimento de um novo modelo de diálogo no framework Rasa, validado em um estudo experimental.

Os resultados do estudo demonstram que o TOB-STT 2.0 alcançou uma proporção de respostas corretas superior à da versão anterior, ao mesmo tempo em que reduziu a frequência de respostas incorretas e de ausência de resposta. Adicionalmente, a análise da percepção dos usuários indicou que a nova versão foi considerada mais eficaz para a obtenção do conhecimento desejado. Em conjunto, essas evidências corroboram a hipótese de que a evolução da arquitetura foi bem-sucedida. Contudo, é importante assinalar que, apesar da melhoria, o chatbot ainda apresenta dificuldades pontuais em responder a certas perguntas, o que demonstra que há margem para evolução.

Com base nos resultados deste trabalho, delineiam-se duas principais frentes para pesquisas futuras. A primeira foca na otimização da interface e da experiência do usuário, uma necessidade que emerge dos resultados do estudo, os quais indicaram que 25% dos estudantes que utilizaram o TOB-STT 2.0 (via Telegram) mostraram-se indecisos quanto à facilidade de uso da interface. Para abordar este ponto, propõe-se uma dupla abordagem: por um lado, portar a interface para outra plataforma de mensagens, a fim de investigar se a usabilidade percebida está atrelada especificamente ao Telegram; por outro, motivado pela percepção positiva quanto à facilidade de uso da interface da versão anterior do chatbot, adaptar e reintegrar a interface web responsiva do TOB-STT 1.0 ao TOB-STT 2.0, conduzindo então uma nova análise comparativa de sua aceitação.

Além do estudo sobre os efeitos da interface do usuário, outra direção de pesquisa envolve a realização de um estudo comparativo entre o TOB-STT 2.0, um chatbot especialista de domínio, e um *Large Language Model* (LLM) generalista, como o ChatGPT[5] ou o Gemini[6]. Embora estudos recentes já explorem o uso de LLMs no apoio ao ensino de teste de software [Jalil et al. 2023, Haldar et al. 2025], há indicações de que, por sua natureza ampla, esses modelos podem cometer erros ou apresentar "alucinações" em questionamentos de nicho técnico. Portanto, um experimento futuro poderia comparar a eficácia das duas abordagens, avaliando métricas como a precisão das respostas e a satisfação dos alunos.

Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, processo nº 306719/2025-8), à Fundação Araucária de Apoio ao De-

[5]Mais informações disponíveis em: `https://openai.com/` .
[6]Mais informações disponíveis em: `https://gemini.google.com/`.

senvolvimento Científico e Tecnológico do Estado do Paraná (Fundação Araucária), e à PROPP/UFGD pelo apoio por meio do projeto SIGProj nº 322855.1174.8276.11032019.

Referências

AbuShawar, B. and Atwell, E. (2016). Usefulness, localizability, humanness, and language-benefit: additional evaluation criteria for natural language dialogue systems. *International Journal of Speech Technology*, 19:373–383.

Adamopoulou, E. and Moussiades, L. (2020). An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16*, pages 373–383. Springer.

Attigeri, G., Agrawal, A., and Kolekar, S. V. (2024). Advanced nlp models for technical university information chatbots: Development and comparative analysis. *IEEE Access*, 12:29633–29647.

Basili, V. R., Caldiera, G., and Rombach, D. H. (1994). *The Goal Question Metric Approach*, volume I. John Wiley & Sons.

Groothuijsen, S., van den Beemt, A., Remmers, J. C., and van Meeuwen, L. W. (2024). Ai chatbots in programming education: students' use in a scientific computing course and consequences for learning. *Computers and Education: Artificial Intelligence*, 7:100290.

Haldar, S., Pierce, M., and Fernando Capretz, L. (2025). Exploring the integration of generative ai tools in software testing education: A case study on chatgpt and copilot for preparatory testing artifacts in postgraduate learning. *IEEE Access*, 13:46070–46090.

Herpich, F., Nunes, F. B., Voss, G. B., and Medina, R. D. (2020). Three-dimensional virtual environment and npc: A perspective about intelligent agents ubiquitous. In *Natural Language Processing: Concepts, Methodologies, Tools, and Applications*, pages 912–938. IGI Global.

Jalil, S., Rafi, S., LaToza, T. D., Moran, K., and Lam, W. (2023). Chatgpt and software testing education: Promises & perils. In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 4130–4137.

Kuhail, M. A., Alturki, N., Alramlawi, S., and Alhejori, K. (2023). Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, 28(1):973–1018.

Lee, D. and Yeo, S. (2022). Developing an ai-based chatbot for practicing responsive teaching in mathematics. *Computers & Education*, 191:104646.

Leifheit, B. R., Basso, F. P., and Silva, W. (2023). Characterizing toolkits for platform independent chatbot development. In *Proceedings of the XIX Brazilian Symposium on Information Systems*, pages 28–36.

Mohamed, A. M. (2024). Exploring the potential of an ai-based chatbot (chatgpt) in enhancing english as a foreign language (efl) teaching: perceptions of efl faculty members. *Education and Information Technologies*, 29(3):3195–3217.

Nee, C. K., Rahman, M. H. A., Yahaya, N., Ibrahim, N. H., Razak, R. A., and Sugino, C. (2023). Exploring the trend and potential distribution of chatbot in education: A systematic review. *International Journal of Information and Education Technology*, 13(3):516–525.

Paschoal, L. N. (2024). *Um framework para o planejamento de experimentos controlados na pesquisa de chatbots educacionais*. Tese de doutorado, Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos.

Paschoal, L. N., de Oliveira, M. M., and Chicon, P. M. M. (2018). A chatterbot sensitive to student's context to help on software engineering education. In *Latin American Computer Conference*, pages 839–848.

Paschoal, L. N., Melo, S. M., Neves, V. d. O., Conte, T. U., and de SOUZA, S. d. R. S. (2023). An experimental study on a conversational agent in software testing lessons. *Informatics in Education*, 22(1):99–120.

Paschoal, L. N., Turci, L. F., Conte, T. U., and Souza, S. R. (2019). Towards a conversational agent to support the software testing education. In *Brazilian Symposium on Software Engineering*, pages 57–66.

Pérez-Soler, S., Guerra, E., and De Lara, J. (2020). Model-driven chatbot development. In *International Conference on Conceptual Modeling*, pages 207–222. Springer.

Pérez-Soler, S., Juárez-Puerta, S., Guerra, E., and de Lara, J. (2021). Choosing a chatbot development tool. *IEEE Software*, 38(4):94–103.

Roca, M. D. L., Chan, M. M., Garcia-Cabot, A., Garcia-Lopez, E., and Amado-Salvatierra, H. (2024). The impact of a chatbot working as an assistant in a course for supporting student learning and engagement. *Computer Applications in Engineering Education*, 32(5):e22750.

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., and Wessln, A. (2012). *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated.