Proposta de Chatbot Inteligente baseado na Organização Acadêmica do Instituto Federal de Pernambuco

Proposal for an Intelligent Chatbot based on the Academic Organization of the Federal Institute of Pernambuco

Alex Emanuel Barbosa de Souza[1], **Flavio Rosendo da Silva Oliveira**[1]

[1] Análise e Desenvolvimento de Sistemas – Instituto Federal de Pernambuco (IFPE)
Paulista – PE – Brasil

aebs@discente.ifpe.edu.br, flavio.oliveira@paulista.ifpe.edu.br

Resumo. *O Instituto Federal de Pernambuco abriga uma variedade de documentos norteadores. Entretanto, acessar as informações contidas nesses documentos nem sempre é uma tarefa simples e rápida, devido à sua extensão e complexidade. Diante desse desafio, este artigo propõe o desenvolvimento de um chatbot inteligente destinado a facilitar o acesso às informações institucionais contidas no documento da Organização Acadêmica do Instituto Federal de Pernambuco. O chatbot utiliza HTML, CSS e ReactJs para a interface do cliente, FastAPI para a aplicação do servidor, MySQL como banco de dados e o modelo BERTimbau para a inteligência do sistema. Adicionalmente, foi criado o conjunto de dados OrgAcadQA, baseado no documento da Organização Acadêmica do Instituto, utilizado juntamente com a base de dados SQuAD v1.1-PT-BR no treinamento e avaliação dos modelos na tarefa de Resposta a Perguntas. O modelo BERTimbau$_{Large}$ demonstrou os resultados mais promissores, alcançando uma Correspondência Exata de 0,78 e uma pontuação F1 de 0,88 na base OrgAcadQA. Esses resultados evidenciaram a eficácia dos modelos BERTimbau na construção de sistemas de Resposta a Perguntas no contexto da Organização Acadêmica do Instituto Federal de Pernambuco.*
Palavras-chave: Processamento de Linguagem Natural; Resposta a Perguntas; Chatbot; BERT.

Abstract. *The Federal Institute of Pernambuco houses a variety of guiding documents. However, accessing the information contained within these documents is often neither simple nor quick due to their length and complexity. To address this challenge, this paper proposes the development of an intelligent chatbot designed to facilitate access to institutional information contained within the Academic Organization document of the Federal Institute of Pernambuco. The chatbot uses HTML, CSS, and ReactJS for the client interface, FastAPI for the server application, MySQL as the database, and the BERTimbau model for system intelligence. Additionally, the OrgAcadQA dataset was created, based on the Academic Organization document of the Institute, and used in conjunction with the SQuAD v1.1-PT-BR dataset for training and evaluating models in the Question Answering task. The BERTimbau$_{Large}$ model achieved the most promising results, reaching an Exact Match of 0.78 and an F1 score of 0.88 on the OrgAcadQA dataset. These results highlight the effectiveness of BERTimbau models in building Question Answering systems within the context of the Academic Organization at the Federal Institute of Pernambuco.*
Keywords: Natural Language Processing; Questions Answering; Chatbot; BERT.

1. Introdução

Grandes empresas, como Google, Facebook, Uber e eBay, têm adotado chatbots inteligentes para aprimorar a comunicação com seus clientes (Csaky, 2019). Os chatbots são sistemas projetados para compreender e interagir com seres humanos por meio da linguagem natural. A adoção desses sistemas proporciona uma série de benefícios, tais como automação e personalização do atendimento, disponibilidade constante em tempo real e capacidade de atender simultaneamente a um grande volume de usuários. A aplicabilidade dessa tecnologia não se limita apenas ao atendimento ao cliente. Atualmente, chatbots são utilizados no suporte a operações financeiras (Wube *et al.*, 2022), facilitando o acesso a informações médicas (Athota *et al.*, 2020) e auxiliando estudantes no processo de aprendizado e na localização de informações (Clarizia *et al.*, 2018).

O Instituto Federal de Pernambuco (IFPE) possui um grande volume de documentos institucionais que normatizam e norteiam vários dos seus processos. Um exemplo desses documentos é a Organização Acadêmica[1]. Este documento visa definir normas, procedimentos e diretrizes para a organização da vida acadêmica de todos os envolvidos no processo educativo nos campi do IFPE (Educação, 2015). Apesar da vasta gama de informações contidas na Organização Acadêmica, a obtenção dessas informações nem sempre é uma tarefa rápida e prática, devido à extensão e complexidade do documento.

Considerando as inúmeras vantagens proporcionadas pelo uso de sistemas de chatbot inteligentes e a possibilidade de estabelecer uma forma mais eficiente de acesso às informações presentes na Organização Acadêmica, este artigo tem como objetivo apresentar e detalhar as etapas do desenvolvimento de um sistema de chatbot inteligente capaz de responder a perguntas relacionadas às informações contidas na Organização Acadêmica do IFPE. O sistema foi implementado como uma solução web, tornando o chatbot acessível em diversas plataformas e dispositivos. Para o desenvolvimento da inteligência do chatbot foram utilizados os modelos BERTimbau, treinados especificamente para a tarefa de Resposta a Perguntas, utilizando a base de dados OrgAcadQA. Dessa forma, o sistema é capaz de localizar precisamente o trecho do documento que contém a resposta à pergunta do usuário, facilitando o acesso às informações desejadas.

O restante desse artigo é estruturado da seguinte forma: A Seção 2 apresenta os conceitos fundamentais que formam a base teórica e contextual do trabalho, além de discutir abordagens adotadas em estudos anteriores sobre o desenvolvimento de chatbots inteligentes e sistemas de Resposta a Perguntas. A Seção 3 detalha os principais aspectos do desenvolvimento dos componentes do sistema e o treinamento realizado para a construção da inteligência do chatbot. A Seção 4 fornece uma descrição detalhada dos experimentos realizados com os modelos BERTimbau, incluindo a exploração de diferentes configurações de parâmetros e estratégias de treinamento, bem como uma análise abrangente dos resultados obtidos em diferentes bases de dados. Por fim, a Seção 5 sumariza os principais resultados alcançados, discute suas implicações, destaca as contribuições do estudo e sugere possíveis direções para futuras pesquisas.

2. Referencial Teórico

2.1. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é um subcampo da Inteligência Artificial (IA) e da Linguística Computacional focado no estudo e desenvolvimento de sistemas que compreendem, interpretam e geram textos em linguagem natural por meio da aplicação de algoritmos e modelos computacionais (Chowdhary, 2020).

[1] https://portal.ifpe.edu.br/wp-content/uploads/repositoriolegado/recife/documentos/organizacao-academica.pdf

Devido à natureza dos dados utilizados na construção dos modelos e à complexidade das tarefas associadas, o desenvolvimento de sistemas baseados em PLN enfrenta diversos desafios. Entre eles, destacam-se a escassez de dados rotulados para determinados domínios e linguagens (Chen *et al.*, 2023), as ambiguidades linguísticas e a diversidade de expressões e termos presentes em uma língua (Khurana *et al.*, 2023).

Em resposta a esses desafios, as técnicas e algoritmos no campo do PLN têm evoluído constantemente. Nos últimos anos, a adoção de Modelos de Linguagem Pré-treinados (Wang *et al.*, 2023) tem alcançado o estado da arte em diversas áreas do PLN, como Análise de Sentimentos, Reconhecimento de Entidades Nomeadas, Sumarização, Tradução Automática e Resposta a Perguntas.

2.2. BERT: Bidirectional Encoder Representations for Transformers

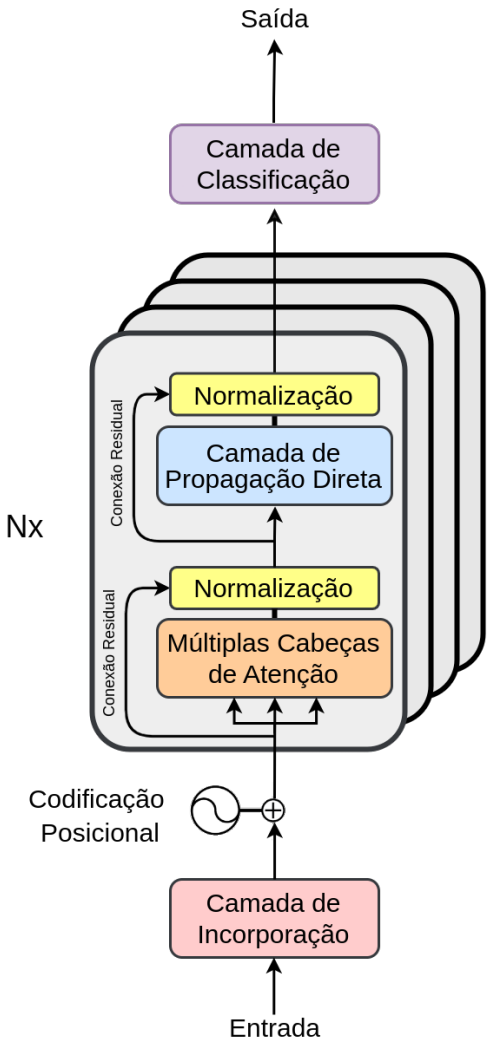
As Representações de Codificador Bidirecional de Transformadores (BERT, do inglês *Bidirectional Encoder Representations from Transformers*) (Devlin *et al.*, 2019) constituem um modelo de linguagem baseado na arquitetura Transformer (Vaswani *et al.*, 2017), pré-treinado para as tarefas de Modelagem de Linguagem Mascarada e Previsão da Próxima Frase em um grande volume de dados textuais não rotulados. Diferentemente dos modelos de linguagem unidirecionais, que processam o texto em uma única direção, o BERT adota uma abordagem bidirecional, permitindo a geração de representações semânticas das palavras com base no contexto em que elas estão inseridas.

A técnica de *fine-tuning* permite adaptar modelos de linguagem, como o BERT, para tarefas específicas de PLN (Howard; Ruder, 2018). Esse processo ajusta os parâmetros do BERT com base em um conjunto de dados especializado, sem a necessidade de mudanças significativas em sua arquitetura. A capacidade de especializar o BERT para tarefas específicas através do *fine-tuning* elevou-o ao status de estado da arte em diversas aplicações de PLN.

A Figura 1 ilustra a arquitetura do modelo BERT. A entrada do modelo é composta por *tokens*, que são as unidades fundamentais de um texto, podendo representar palavras, sub-palavras ou caracteres. A camada de incorporação (do inglês *embedding layer*) transforma os *tokens* de entrada em representações vetoriais de alta dimensionalidade, que capturam o significado semântico de cada *tokens*. Em seguida, aplica-se uma codificação posicional, que atribui uma representação baseada no posicionamento relativo dos *tokens* dentro das sentenças. As camadas do codificador Transformer — 12 no $\text{BERT}_{\text{Base}}$ e 24 no $\text{BERT}_{\text{Large}}$ — desempenham um papel crucial na captura de informações contextuais de forma bidirecional. Essas camadas são compostas por subcamadas de Múltiplas Cabeças de Atenção e de Propagação Direta. Ao redor de cada subcamada, existe uma conexão residual que, junto com as saídas das subcamadas, passam por um processo de normalização. A saída da última camada de codificação é então processada por uma camada de classificação, que é ajustada para a tarefa específica em que o modelo BERT está sendo utilizado, como Classificação de Texto ou Resposta a Perguntas, gerando assim a predição final do modelo.

Embora exista uma versão multilíngue do BERT, consideráveis esforços têm sido dedicados à criação de versões pré-treinadas específicas para diferentes idiomas. Frequentemente, esses modelos demonstram desempenhos superiores à versão multilíngue, como evidenciado pelos modelos BERTimbau (Souza; Nogueira; Lotufo, 2020). O BERTimbau, desenvolvido pela empresa NeuralMind, é uma versão do BERT pré-treinada com dados textuais em português, utilizando o corpus brWaC (Wagner *et al.*, 2018). Disponível nas versões $\text{BERTimbau}_{\text{Base}}$ e $\text{BERTimbau}_{\text{Large}}$, o modelo tem se destacado em tarefas como Reconhecimento de Entidades Nomeadas, Reconhecimento de Implicação Textual, Similaridade Semântica Textual (Mello *et al.*, 2024) e Resposta a Perguntas (Silva; Laterza; Faleiros, 2022) na língua portuguesa.

Figura 1. Representação da arquitetura base do modelo BERT. O modelo é formado por uma camada de incorporação, N camadas de codificadores do Transformer e uma camada de classificação.



Fonte: Adaptado de (Vaswani *et al.*, 2017)

2.3. Resposta a Perguntas

Resposta a Perguntas (QA, do inglês *Question Answering*) é um subcampo do PLN e da Recuperação de Informação (IR, do inglês *Information Retrieval*) que se dedica ao desenvolvimento de sistemas capazes de compreender perguntas formuladas em linguagem natural e fornecer respostas relevantes e precisas (Allam; Haggag, 2012). A área de QA tem ganhado destaque devido à sua ampla aplicabilidade, que inclui sistemas de busca avançados, análise de documentos e chatbots inteligentes.

Diversas abordagens têm sido empregadas no desenvolvimento de sistemas de QA (Calijorne Soares; Parreiras, 2020). Entre elas, a abordagem de Compressão de Leitura de Máquina (MRC, do inglês *Machine Reading Comprehension*) se destaca por sua capacidade de criar modelos robustos e altamente adaptáveis. Essa abordagem utiliza modelos avançados de aprendizado de máquina para entender perguntas e extrair respostas precisas a partir de um contexto (Zeng *et al.*, 2020).

2.4. SQuAD: The Stanford Question Answering Dataset

O Conjunto de Dados de Resposta a Perguntas de Stanford (SQuAD, do inglês *The Stanford Question Answering Dataset*) é uma das principais bases de dados de domínio aberto utilizados para o treinamento e avaliação de modelos de PLN, especialmente em tarefas de MRC. Seu objetivo central é, dado um contexto e uma pergunta, localizar o trecho exato no texto que contenha a resposta correta (Rajpurkar *et al.*, 2016).

O SQuAD v1.1 é composto por 107.785 perguntas elaboradas por colaboradores a partir de 536 artigos da Wikipédia. Esses artigos foram segmentados em parágrafos, que serviram como referência para a formulação das perguntas. Cada pergunta possui como resposta um ou mais trechos do parágrafo correspondente. O conjunto de dados é dividido em 80% para treinamento, 10% para validação e 10% para teste

Desde seu lançamento, o SQuAD v1.1 tem sido amplamente utilizado como um *benchmark* para a avaliação de modelos de QA. Visando possibilitar o uso desse conjunto de dados para a avaliação de modelos na língua portuguesa, em 2020 foi disponibilizado o SQuAD v1.1-PT-BR[2] pela comunidade inter-institucional Deep Learning Brasil. Essa versão foi criada através da tradução automática do SQuAD v1.1 para o português, utilizando uma API do Google Cloud.

A Figura 2 apresenta uma amostra dos dados da base SQuAD v1.1-PT-BR. Cada registro na base inclui um título que descreve o tema principal do artigo de onde os parágrafos foram extraídos, além de uma lista de parágrafos. Cada parágrafo é composto por um contexto e um conjunto de perguntas e respostas. O contexto refere-se ao conteúdo textual de um parágrafo do artigo, que serve como base para a formulação das perguntas. Para cada pergunta, são fornecidas uma ou mais respostas, que consistem no texto da resposta e um atributo que indica a posição inicial da resposta dentro do contexto.

2.5. Trabalhos Relacionados

(Neto *et al.*, 2022) apresentam o desenvolvimento de um chatbot com o objetivo de auxiliar estudantes no acesso a informações relacionadas a uma universidade pública. Para a construção do chatbot, foi utilizada a biblioteca RASA, que facilita a estruturação do sistema e oferece diversos algoritmos para as etapas de pré-processamento, além de utilizar o modelo DIET para a classificação de intenções das perguntas. Diante de uma questão formulada pelo usuário, o sistema deve retornar a resposta previamente cadastrada no banco de dados que melhor corresponda à pergunta realizada. Para avaliação dos resultados, foi desenvolvida uma base de 476 perguntas, baseadas na seção de ”Perguntas Frequentes”do site do curso de Ciência da Computação, onde o sistema alcançou uma precisão de 90,9% nas respostas retornadas. De forma semelhante ao trabalho de (Neto *et al.*, 2022), o chatbot proposto neste estudo visa auxiliar os usuários no acesso a informações relacionadas a uma instituição de ensino. Contudo, ao contrário da solução anterior, optou-se por uma abordagem baseada em MRC no desenvolvimento do chatbot, conferindo-lhe a capacidade de responder perguntas com base no contexto das informações, eliminando assim a necessidade de uma base de perguntas e respostas previamente cadastradas.

Em (Collarana *et al.*, 2018), propõe-se o desenvolvimento de um sistema de QA para acesso a informações contidas em textos regulatórios. O sistema é dividido em dois módulos. No primeiro módulo, ocorre a seleção dos parágrafos mais relevantes de acordo com a pergunta, a partir de um conjunto de documentos. Posteriormente, no módulo de seleção de respostas, os parágrafos selecionados na etapa anterior são analisados para encontrar o exato trecho que responde à pergunta. O sistema

[2]https://huggingface.co/datasets/ArthurBaia/squad_v1_pt_br

Figura 2. Amostra da base SQuAD v1.1-PT-BR


```
{
    "title": "Software_testing",
    "paragraphs": [
      {
        "context": "Como o número de testes possíveis, mesmo para componentes simples de software,
é praticamente infinito, todos os testes de software usam alguma estratégia para selecionar testes viáveis
pelo tempo e pelos recursos disponíveis. Como resultado, o teste de software normalmente (mas não
exclusivamente) tenta executar um programa ou aplicativo com a intenção de encontrar bugs de
software (erros ou outros defeitos). O trabalho de teste é um processo iterativo, pois quando um
bug é corrigido, ele pode iluminar outros erros mais profundos ou até criar novos.",
        "qas": [
          {
            "id": "57290d3a6aef0514001549f7",
            "question": "Por que é tão difícil localizar erros no software?",
            "answers": [
              {
                "answer_start": 17,
                "text": "testes possíveis, mesmo para componentes simples de software, é praticamente infinito"
              }
            ]
          },
          {
            "id": "57290d3a6aef0514001549f6",
            "question": "Qual é o objetivo de testar o software?",
            "answers": [
              {
                "answer_start": 359,
                "text": "encontrar bugs de software"
              }
            ]
          },
          {
            "id": "57290d3a6aef0514001549f8",
            "question": "O que pode resultar de um bug sendo corrigido?",
            "answers": [
              {
                "answer_start": 491,
                "text": "ele pode iluminar outros erros mais profundos"
              }
            ]
          }
        ]
      }
    ]
}
```


foi avaliado em uma base com 631 pares de perguntas e respostas geradas a partir do documento regulatório MaRisk. Através de diferentes configurações de treinamento, os autores constataram uma melhoria significativa na performance do sistema ao realizar um treinamento inicial em uma base ampla de domínio aberto, o SQuAD v1.1, seguido de um *fine-tuning* com a base de domínio regulatório criada com o MaRisk. O modelo treinado em ambas as bases alcançou um F1 de 0,59 e uma Correspondência Exata de 0,34. Similarmente ao trabalho apresentado, este estudo investiga o ganho de performance ao treinar um modelo de MRC com múltiplas etapas de *fine-tuning*, utilizando uma base ampla de domínio aberto, seguida por uma segunda etapa de *fine-tuning* em uma base de domínio alvo no contexto da Organização Acadêmica.

Quanto à aplicação do BERT na tarefa de QA, o estudo realizado por (Chen; Zulkernine, 2021) oferece uma análise abrangente por meio de diversos experimentos, explorando múltiplas variantes dos modelos BERT e ALBERT no desenvolvimento de um sistema de QA específico para um determinado domínio. O autor investiga diversas técnicas de pré-processamento e estratégias de treinamento, avaliando seus impactos durante o treinamento dos modelos. O objetivo principal do estudo é desenvolver um sistema de QA capaz não apenas de localizar o trecho exato de uma resposta,

mas também de determinar se a informação necessária para responder à pergunta está presente nos documentos utilizados pelo modelo. Para alcançar esse objetivo, a base SQuAD v2.0 foi empregada no treinamento e na avaliação dos modelos. Os modelos treinados com a estratégia de *fine-tuning* e o pré-processamento do tipo Segmentação apresentaram os resultados mais promissores, alcançando aproximadamente 75% de Correspondência Exata e 78% de F1 na base SQuAD v2.0. Semelhante ao trabalho mencionado, este estudo também adotará variações do modelo BERT no desenvolvimento de um sistema de QA de domínio específico. A combinação da técnica de pré-processamento de Segmentação e da estratégia de treinamento *fine-tuning* será adotada, fundamentada nos resultados promissores apresentados por (Chen; Zulkernine, 2021).

A utilização de dados de domínio específico no treinamento de modelos de QA é crucial para assegurar uma compreensão aprofundada e precisa do contexto em que o sistema será aplicado. (Sayama; Araujo; Fernandes, 2019) apresenta o desenvolvimento do FAQUAD, um conjunto de dados para a tarefa de MRC no domínio de instituições brasileiras de ensino superior. A base de dados é composta por 900 perguntas, derivadas de 249 parágrafos, extraídos de 18 documentos oficiais do curso de Ciência da Computação de uma faculdade federal brasileira e 21 artigos da Wikipédia relacionados ao sistema de ensino superior brasileiro. Ao longo do artigo, o autor conduziu uma série de experimentos utilizando diferentes combinações de representações pré-treinadas com o modelo BIDAF. Utilizando as representações pré-treinadas geradas pelo ELMo, o modelo BIDAF alcançou aproximadamente 43% de F1 e 24% de Correspondência Exata na base FAQUAD. Semelhante ao estudo apresentado, neste trabalho será desenvolvida uma base de dados de domínio específico para a tarefa de MRC, fundamentada no contexto da Organização Acadêmica.

3. Metodologia

Nesta seção, serão explorados os principais aspectos do desenvolvimento do chatbot. Inicialmente, será apresentada uma visão geral da arquitetura do sistema, seguida por uma análise detalhada da implementação de cada um dos seus componentes. Em seguida, será descrito o processo de criação da base OrgAcadQA, utilizada para treinar e avaliar o modelo inteligente na tarefa de QA no contexto da Organização Acadêmica. Por fim, é descrito o processo de treinamento e avaliação dos modelos na tarefa de QA, abordando aspectos referentes a escolha dos modelos, bases de dados utilizadas, estrategias de treinamento e métricas de avaliação.

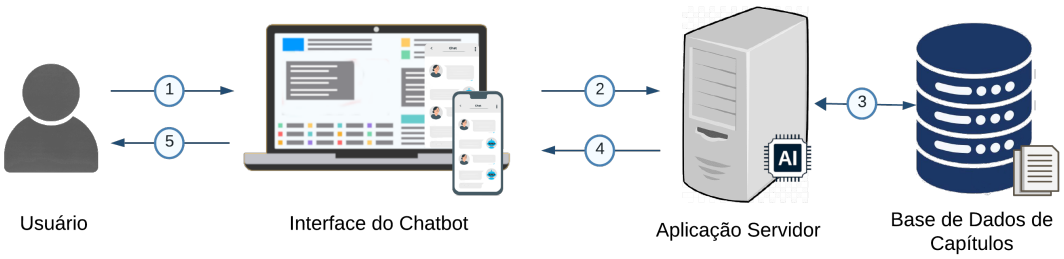
3.1. Desenvolvimento dos Componentes do Chatbot

A Figura 3 apresenta uma visão geral da arquitetura do sistema. A interação inicial ocorre entre o usuário e o chatbot por meio de uma interface *web*, onde o usuário informa a sua pergunta e indica a qual tópico ela pertence. Essas informações são então enviadas à aplicação servidor por meio de uma requisição HTTP. A aplicação servidor processa a requisição e utiliza o tópico escolhido pelo usuário na realização de uma consulta SQL a base dados de capítulos, afim de recuperar o contexto referente a aquele determinado tópico. Tanto o contexto quanto a pergunta são utilizados pelo modelo inteligente para extração do trecho do contexto que responde à pergunta do usuário. A aplicação cliente recebe a resposta da requisição, que contém a resposta para a pergunta do usuário. Por fim, a resposta é enviada para o usuário.

3.1.1. Interface Cliente do Chatbot

A interface do usuário desempenha um papel crucial no desenvolvimento de um chatbot, pois, é através dela que são realizadas as interações entre usuário e máquina. Uma interface bem projetada e

Figura 3. Visão geral do fluxo de execução do sistema. (1) Interação inicial entre o usuário e a interface cliente do chatbot. (2) Envio do tópico e pergunta para a aplicação servidor. (3) Consulta ao banco de dados de capítulos. (4) Retorno da resposta para a pergunta com base no contexto consultado. (5) Envio da resposta para o usuário.



intuitiva não apenas facilita a interação, mas também contribui significativamente para a experiência do usuário (Sharma; Tiwari, 2021).

No desenvolvimento do sistema proposto, optou-se pela implementação de uma interface *web*, devido as diversas vantagens que essa abordagem oferece. A adoção dessa estrategia proporciona grande acessibilidade, permitindo que os usuários interajam com o sistema através de qualquer dispositivo conectado à internet. Adicionalmente, facilita a integração com outras plataformas online, como portais de informação e sites universitários.

A interface cliente do chatbot foi desenvolvida utilizando as tecnologias HTML, CSS e a biblioteca React-chatbot-kit [3] da linguagem Javascript. A biblioteca React-chatbot-kit possibilita a criação de diversos elementos de interface para o desenvolvimento de chatbots, como o *template* do chat, caixas de diálogo e *widgets*. Além disso, através da biblioteca, também foi possível realizar a configuração de ações, responsáveis por controlar as respostas retornadas pelo chatbot com base nas interações com o usuário. Foram aplicados estilos e animações para melhorar a estética e usabilidade da interface, proporcionando uma experiência de conversação potencialmente mais natural para os usuários.

Na Figura 4 é representado o fluxo de interação entre o chatbot e o usuário. O chatbot inicia a interação saudando o usuário e realizando uma breve apresentação. Logo após, é solicitado ao usuário que escolha um tópico, que definirá o contexto da sua pergunta, a partir de uma lista previamente cadastrada. Após isso, o usuário realiza sua pergunta. A pergunta do usuário juntamente com o tópico escolhido são utilizados na construção de uma requisição HTTP enviada para aplicação servidor. O retorno dessa requisição contém a resposta para a pergunta do usuário. Por fim, é dado ao usuário a oportunidade de realizar novas perguntas ou finalizar a interação.

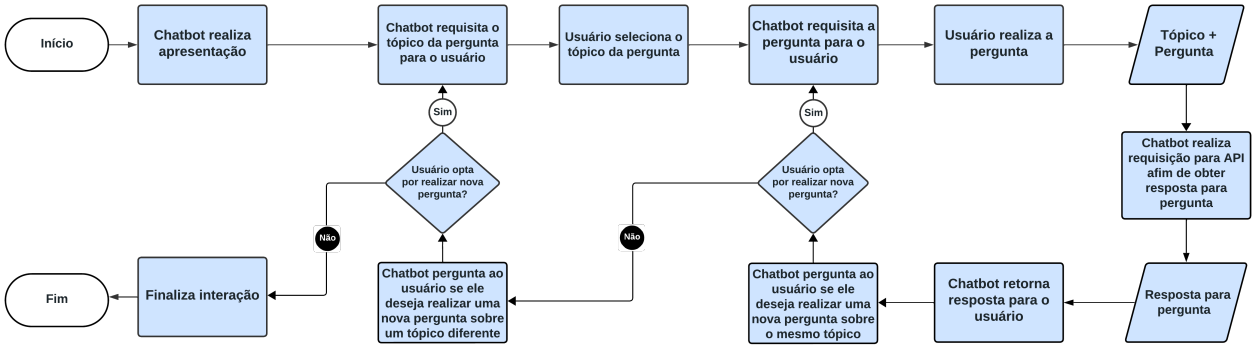
3.1.2. Aplicação Servidor

A aplicação servidor é o componente responsável por processar as requisições realizadas pela interface do usuário e retornar a resposta para a pergunta realizada com base no contexto do tópico indicado.

A Figura 5 apresenta um diagrama de sequência, construído a partir do *website* sequencedi-

[3] https://github.com/FredrikOseberg/react-chatbot-kit

Figura 4. Representação das etapas de interação entre o chatbot e o usuário



agram.org[4], que representa o comportamento interno da aplicações servidor e de que forma os seus componentes se relacionam. Inicialmente, a aplicação recebe o tópico e a pergunta realizada pelo usuário através de uma requisição HTTP a API REST da aplicação, representada pela classe APIRequestHandler. Em seguida, os dados da requisição são passados para a classe QAService, responsável por orquestrar a interação entre os demais componentes da aplicação para o retorno da resposta. Para isso, primeiro é necessário identificar o contexto, conjunto de parágrafos da Organização Acadêmica referentes ao tópico informado pelo usuário. Esse procedimento é realizado pela classe ContextDatabaseClient através de uma consulta a base de dados de capítulos, onde são armazenados os contextos para cada um dos tópicos cadastrados. A aplicação então utiliza a pergunta, juntamente do contexto, como entrada para o modelo inteligente, reapresentado pela classe BERTQAModel.

Por sua vez, o modelo retorna o exato trecho do contexto que contém a resposta para a pergunta realizada. Por fim, a aplicação devolve a resposta obtida para interface do usuário.

A API foi desenvolvida utilizando o *framework web* FastAPI[5] da linguagem Python, uma escolha fundamentada na simplicidade de implementação, robustez, performance e documentação automática proporcionada por esse *framework* (Lathkar, 2023). Para estabelecer a comunicação com o banco de dados, adotou-se a biblioteca mysql-connector-Python[6]. Por fim, para a interação e implementação do modelo inteligente, foi utilizada a biblioteca Transformers[7], que tem como principal objetivo disponibilizar modelos estado-da-arte baseados na arquitetura Transformer (Wolf *et al.*, 2020).

3.1.3. Banco de Dados de Capítulos da Organização Acadêmica

A base de dados de capítulos tem como propósito armazenar informações associados a cada capítulo presente no documento da Organização Acadêmica. Essa abordagem visa viabilizar o acesso eficiente aos contextos, ao mesmo tempo em que centraliza o armazenamento e a administração dessas informações. Todas as informações relativas aos capítulos foram consolidadas em uma tabela. A Tabela 1 apresenta a estrutura da tabela de capítulos.

[4]https://sequencediagram.org/
[5]https://fastapi.tiangolo.com/
[6]https://dev.mysql.com/doc/connector-python/en/
[7]https://huggingface.co/docs/transformers/index

Figura 5. Diagrama de sequência dos componentes internos da aplicação servidor.

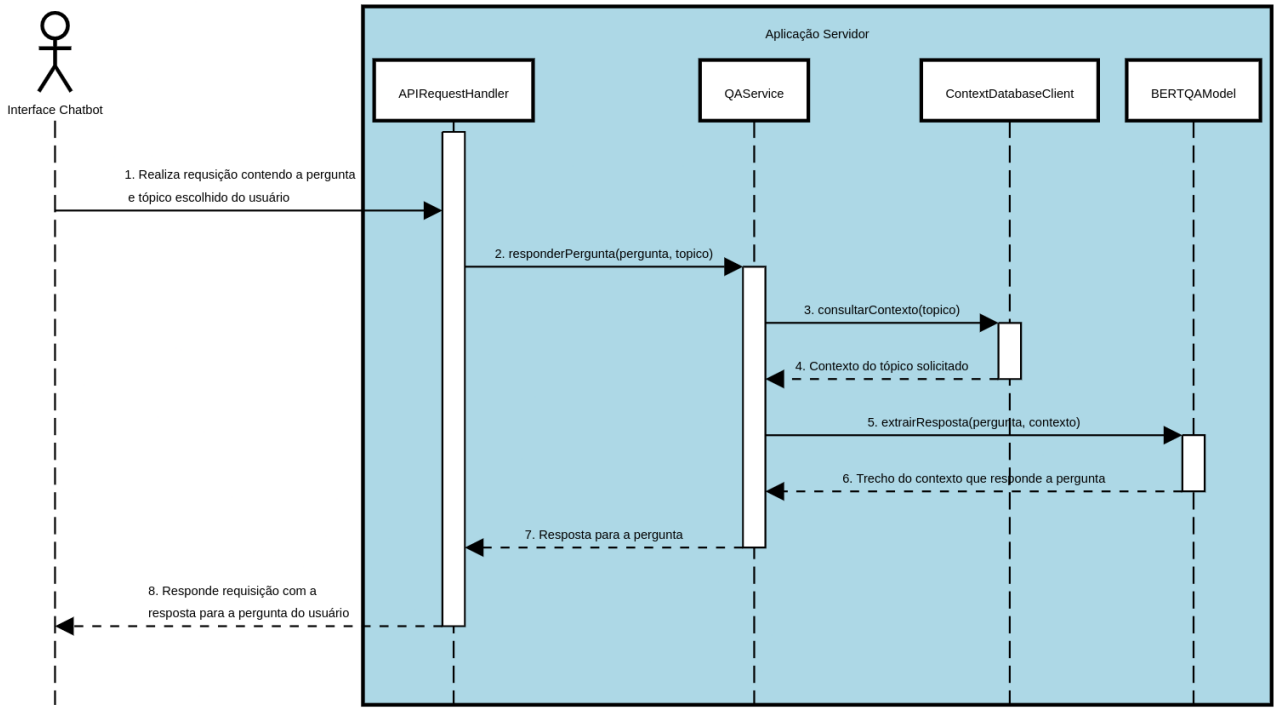


Tabela 1. Descrição da estrutura da tabela de capítulos

Coluna	Tipo	Descrição
id	TINYINT	Número do Capítulo. Chave primária da tabela.
titulo	VARCHAR(50)	Título do capítulo.
contexto	LONGTEXT	Conteúdo textual do capítulo.

A Organização Acadêmica compreende um extenso documento de 89 páginas, organizado em 19 capítulos. Cada capítulo explora um segmento distinto relacionada ao processo educacional e da vida acadêmica no IFPE. Esses capítulos são, por sua vez, subdivididos em seções, artigos e parágrafos, nos quais são abordados pontos específicos, tais como normas, procedimentos e orientações referentes a cada tema. A extensão do documento, sua formatação e a presença de marcações textuais são elementos que podem apresentar desafios na aplicação de modelos NLP. Diante disso, as seguintes etapas de pré-processamento foram realizadas no documento da Organização Acadêmica para geração da tabela de capítulos.

1. Conversão do documento da Organização Acadêmica do formato PDF para TXT utilizando a biblioteca PyPDF2[8], viabilizando a manipulação dos dados e a aplicação de técnicas de pré-processamento no texto.
2. Remoção de marcações e estruturas de formatação do texto, através de expressões regulares. Esse procedimento contribui para um texto mais uniforme e livre de elementos indesejados.
3. Segmentação do documento em capítulos, possibilitando a manipulação dos capítulos do documento de forma independente.

[8]https://pypdf2.readthedocs.io/en/3.0.0/

4. Criação dos registros da tabela de capítulos. O texto do capítulo, em conjunto com seu título e um identificador único, formam um registro na tabela de capítulos.

Para o desenvolvimento do banco de dados dos capítulos, foi escolhida a versão 8.0.21 do sistema de gerenciamento de bancos de dados (SGBD) MySQL. Esta escolha baseou-se na ampla adoção e consolidação no mercado do MySQL.

3.2. Construção da Base de Dados OrgAcadQA

No desenvolvimento de modelos de QA para domínios específicos, a inclusão de dados do domínio-alvo durante a etapa de treinamento é essencial para aprimorar a capacidade do modelo em compreender e fornecer respostas mais precisas a perguntas relacionadas a esse domínio particular. Nesse cenário, a base OrgAcadQA foi desenvolvida com o propósito de treinar e avaliar modelos para a tarefa de MRC no contexto da Organização Acadêmica.

A OrgAcadQA foi desenvolvida seguindo o mesmo formato do SQuAD v1.1, garantindo assim compatibilidade na lógica de pré-processamento dos dados e avaliação dos modelos. Para a criação da base OrgAcadQA, foram selecionados cinco capítulos da Organização Acadêmica, escolhidos como uma amostragem representativa do documento. A seleção dos capítulos foi realizada com base na experiência do autor como estudante, levando em consideração sua percepção sobre os capítulos que mais geram dúvidas entre os estudantes. O conteúdo dos capítulos foi utilizado como base para a formulação das perguntas e respontas da base de dados.

A Figura 6 apresenta uma amostra dos dados da base OrgAcadQA. Cada registro contém um título que indica a qual capítulo do documento da Organização Acadêmica o registro se refere, além de uma lista de parágrafos. Cada parágrafo é composto por um contexto e um conjunto de perguntas e respostas. O contexto corresponde ao conteúdo textual do capítulo, utilizado como base para a formulação das perguntas. Para cada pergunta, são fornecidas uma ou mais respostas, que incluem o texto da resposta e um atributo que indica a posição inicial da resposta dentro do contexto.

Um total de 100 perguntas foram elaboradas, de forma manual, contendo de uma a três respostas com base em trechos específicos presentes no determinado contexto. A distribuição de perguntas por capítulo pode ser visualizada na Tabela 2.

Tabela 2. Relação da quantidade de perguntas por capítulo

Capítulo	Quantidade de Perguntas
Trabalho de Conclusão de Curso (TCC)	20
Solenidades de Conclusão de Curso	20
Conclusão dos Cursos	20
Trancamento de Matrícula	15
Prática Profissional	25
Total	100

3.3. Treinamento e Avaliação dos Modelos Inteligentes

Para o desenvolvimento do modelo inteligente capaz de compreender e responder as perguntas realizadas pelos usuários, foram utilizados como base os modelos BERTimbau. Devido ao pré-treinamento realizado utilizando um grande volume de dados textuais não rotulados da língua portuguesa, o modelo é capaz de compreender representações semânticas da língua e generalizar esses conhecimentos para tarefas especificas do PLN, melhorando significativamente o seu desempenho. Com poucas

Figura 6. Amostra da base OrgAcadQA referente ao capitulo de Solenidades de Conclusão de Curso.


```
{
    "title": "Solenidades de Conclusão de Curso",
    "paragraphs": [
      {
        "context": "...Os estudantes concluintes dos Cursos Técnicos de Nível Médio são apresentados à
sociedade por meio  de uma solenidade de formatura, que possui caráter não obrigatório e constitui ato
simbólico...  A sessão solene será  agendada mediante solicitação do Coordenador do Curso à  Comissão
Institucional de Formatura com antecedência mínima de 60 (sessenta) dias do término do  período letivo de
conclusão do curso. A solicitação deverá ser encaminhada por meio de requerimento nomeando os
integrantes da Comissão de Formatura dos concluintes, contendo a proposta de data, horário, local e o
número de prováveis formandos. A solenidade de colação de grau será presidida pelo Reitor(a) ou
representante por ele designado...",
        "qas": [
          {
            "id": "647427ea-f490-4411-adfb-22e7e65b4b69",
            "question": "Com quantos dias de antecedência mínima deve ser solicitada a agendamento
                        de uma solenidade de colação de grau?",
            "answers": [
              {
                "answer_start": 3721,
                "text": "60 (sessenta) dias"
              },
              {
                "answer_start": 3694,
                "text": "com antecedência mínima de 60 (sessenta) dias"
              }
            ]
          },
          {
            "id": "5aa9c01b-dd46-47db-911b-09a8f5b42d87",
            "question": "Quem preside a solenidade de colação de grau?",
            "answers": [
              {
                "answer_start": 3999,
                "text": "A solenidade de colação de grau será presidida pelo Reitor(a) ou representante
                        por ele designado"
              },
              {
                "answer_start": 4051,
                "text": "Reitor(a) ou representante por ele designado"
              },
              {
                "answer_start": 4046,
                "text": "pelo Reitor(a) ou representante por ele designado"
              }
            ]
          }
        ]
      }
    ]
}
```


adaptações na sua arquitetura, os modelos BERTimbau podem ser aplicados a tarefa de QA (Devlin *et al.*, 2019). Afim de avaliar os modelos BERTimbau na tarefa de QA, foram adotadas as seguintes métricas:

- **Correspondência Exata**: A métrica de Correspondência Exata (EM, do inglês *Exact Match*) avalia se a resposta fornecida pelo modelo é exatamente igual a resposta esperada. Essa métrica atribui uma pontuação de 1 para respostas totalmente corretas e 0 para os demais casos. Em cenários com múltiplas respostas corretas, a métrica de EM considera a pontuação como 1 caso a resposta prevista pelo modelo seja igual a ao menos uma das respostas esperadas. Embora a EM forneça uma avaliação clara da precisão, ela pode ser menos flexível em

tarefas onde respostas parciais ou sinônimos são aceitáveis.

$$EM = \begin{cases} 1 & \text{se a resposta gerada é exatamente igual à resposta esperada} \\ 0 & \text{caso contrário} \end{cases} \tag{1}$$

- **Pontuação F1**: A métrica de Pontuação F1 fundamenta-se na sobreposição de *tokens* entre a resposta esperada e aquela predita pelo modelo. O F1 é derivado da média harmônica da precisão e da revocação. No contexto de QA, a precisão é definida como a proporção entre o número de *tokens* compartilhados pela resposta predita e a resposta esperada, em relação ao total de *tokens* presentes na resposta predita pelo modelo. Por outro lado, a revocação é a proporção entre o número de *tokens* compartilhados entre as respostas e o total de *tokens* na resposta esperada. Em cenários com múltiplas respostas corretas, considera-se o valor máximo de F1 entre todas as respostas possíveis. Nos sistemas de QA, a métrica F1 é especialmente útil por permitir a avaliação de respostas parcialmente corretas, diferentemente de métricas como o EM, que consideram apenas respostas totalmente exatas.

$$\text{Precisão} = \frac{\text{Número de } \textit{Tokens} \text{ Corretamente Preditos}}{\text{Total de } \textit{Tokens} \text{ na Resposta Predita}} \tag{2}$$

$$\text{Revocação} = \frac{\text{Número de } \textit{Tokens} \text{ Corretamente Preditos}}{\text{Total de } \textit{Tokens} \text{ na Resposta Esperada}} \tag{3}$$

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão + Revocação}} \tag{4}$$

3.3.1. Pré-processamento dos Dados para a Tarefa de QA

Inicialmente, o processo envolve a extração das informações de contextos, perguntas e respostas a partir da base de dados, as quais são então organizadas de maneira estruturada. Essa organização resulta na formação de unidades que relacionam cada pergunta ao seu respectivo contexto e as respostas correspondentes. Para cada resposta, é realizado o cálculo da posição final com base na soma da posição inicial e do comprimento do texto da resposta. As informações relacionadas à posição de início e fim da resposta desempenham um papel crucial na avaliação dos resultados gerados pelo modelo.

Posteriormente, inicia-se o processo de tokenização, no qual cada par de pergunta e resposta é dividido em unidades menores denominadas *tokens*. A técnica empregada para a tokenização é o WordPiece (Wu *et al.*, 2016), para garantir a compatibilidade com o vocabulário dos modelos BERTimbau. Após essa etapa, são adicionados *tokens* especiais, como [CLS], utilizado em tarefas de classificação, e [SEP], empregado para delimitar os *tokens* da pergunta e os do contexto. Cada *token* é então mapeado para um índice numérico correspondente no vocabulário do modelo BERTimbau.

Os modelos BERTimbau apresentam limitações em relação ao número máximo de *tokens* que podem ser processados em uma única sequência. Se a sequência de entrada, resultante da concatenação dos *tokens* da pergunta e do contexto, ultrapassar o limite predefinido, é implementada a estratégia de segmentação. Essa abordagem divide o contexto em segmentos, e os *tokens* de cada segmento, juntamente com os da pergunta, originam uma nova sequência de entrada. Por outro

lado, caso a sequência tenha um tamanho inferior ao estabelecido, adota-se a estratégia de preenchimento. Essa medida visa padronizar o tamanho das sequências de entrada dentro de um lote de treinamento.

Por fim, são gerados os vetores de segmento e a máscara de atenção. O vetor de segmento tem a função de informar ao modelo quais *tokens* da sequência de entrada pertencem à pergunta e quais são referentes ao contexto, atribuindo valores distintos conforme a origem do *tokens*. Essa distinção é essencial para o correto entendimento do modelo. A máscara de atenção, por sua vez, indica quais *tokens* fazem parte da sequência original e quais são resultados da estratégia de preenchimento. Essa máscara permite que o modelo se concentre apenas nos *tokens* relevantes durante o treinamento, otimizando assim o processo de aprendizado.

As etapas de pré-processamento descritas foram executadas para cada uma das bases utilizadas neste artigo, durante seus respectivos experimentos. O processo de tokenização foi implementado por meio da classe BertTokenizer [9] da biblioteca Transformers.

3.3.2. Treinamento e Avaliação dos Modelos em Dados de Domínio Aberto

O *fine-tuning* de modelos BERT para uma tarefa especifica demanda um grande volume de dados rotulados, o que nem sempre é possível para determinados domínios. A utilização de bases de domínio aberto, caracterizadas por seu amplo volume e representatividade, surge como uma solução para mitigar essa limitação, permitindo que esses modelos aprendam a desempenhar a tarefa especifica sem depender da existência de dados do domínio alvo. Neste artigo, para o treinamento dos modelos BERTimbau para a tarefa de QA, será utilizada a base SQuAD v1.1-PT-BR. Com isso, espera-se desenvolver um modelo capaz de conseguir responder perguntas de vários contextos realizadas na língua portuguesa.

3.3.3. Treinamento e Avaliação dos Modelos em Dados de Domínio Alvo

Com o intuito de avaliar os modelos BERTimbau em dados de domínio específico, foram utilizadas as bases FAQUAD e OrgAcadQA. Ambas as bases abrangem o contexto de instituições brasileiras de ensino superior. Entretanto, enquanto o FAQUAD foi desenvolvido a partir de um contexto mais amplo, abrangendo documentos oficiais e artigos do Wikipédia relacionados a instituições de ensino superior, a base OrgAcadQA foca exclusivamente nas informações contidas no documento da Organização Acadêmica do Instituto Federal de Pernambuco.

Para a otimização dos hiperparâmetros utilizados durante o treinamento dos modelos, foi empregada a técnica de *Grid Search*. Essa técnica explora de forma exaustiva todas as combinações possíveis de valores para os hiperparâmetros que se que deseja otimizar, avaliando o desempenho de cada combinação com base em uma determinada métrica (Liashchynskyi; Liashchynskyi, 2019). A técnica *Grid Search* foi implementado através da classe GridSampler[10] da biblioteca Optuna (Akiba *et al.*, 2019), devido a sua integração com a biblioteca Transformers.

A abordagem de busca exaustiva adotada pelo *Grid Search* pode demandar considerável tempo e recursos computacionais, especialmente em conjunto de dados extensos ou em modelos complexos. Afim de minimizar esses custos, adotou-se a técnica de Parada Antecipada (Prechelt, 2002).

[9] https://github.com/huggingface/transformers/blob/v4.37.2/src/transformers/models/bert/tokenization_bert.py
[10] https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.GridSampler.html

Ao incorporar a técnica de Parada Antecipada no *Grid Search*, a busca por hiperparâmetros é interrompida quando não há melhorias nos resultados do modelo acima de um certo percentual após um determinado número de rodadas de avaliação em um conjunto de dados. Dessa forma, evitando a exploração desnecessária do espaço de busca dos hiperparâmetros e, ao mesmo tempo, contribuindo para a obtenção de modelos mais generalizáveis. A técnica de Parada Antecipada foi implementado através da classe PatientPruner[11] da biblioteca Optuna.

Com o intuito de avaliar de maneira robusta e consistente o desempenho proveniente das diversas combinações de hiperparâmetros derivadas da aplicação do *Grid Search*, adotou-se a técnica de validação cruzada. Essa abordagem visa realizar a avaliação do modelo em diferentes conjuntos de dados, oferecendo uma análise robusta da sua capacidade de generalização para dados não observados. A implementação dessa técnica foi conduzida por meio da classe GroupKFold[12] do scikit-learn (Pedregosa *et al.*, 2011), uma variação da validação cruzada K-Fold, projetada para lidar com grupos específicos de dados. O objetivo é garantir que amostras de um mesmo grupo não estejam simultaneamente presentes nos conjuntos de treino e teste durante as iterações, prevenindo possíveis vieses introduzidos por dependências ou similaridades entre amostras do mesmo grupo. Nos experimentos conduzidos, os grupos foram definidos com base no contexto de cada pergunta.

Para cada modelo, foram avaliadas as performances de três abordagens de treinamento. Cada abordagem fundamenta-se na base de dados utilizada para realização do *fine-tuning* para a tarefa de QA. O principal objetivo é avaliar como cada configuração de treinamento impacta o resultado final da avaliação dos modelos nas bases de domínio alvo.

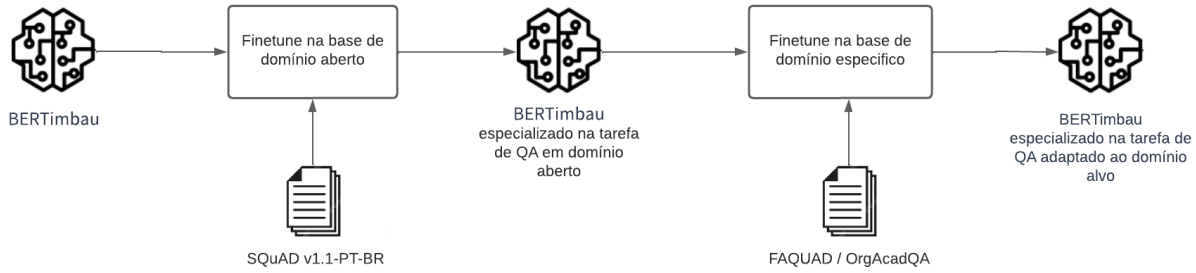
i) ***Fine-tuning* em Dados de Domínio Aberto**: A primeira abordagem caracteriza-se pelo treinamentos dos modelos em uma base ampla de dados de domínio aberto. O objetivo é avaliar a capacidade de generalização desses modelos quando expostos a contextos específicos. Para isso serão utilizados os modelos treinados na base SQuAD v1.1-PT-BR.

ii) ***Fine-tuning* em Dados de Domínio Específico**: Em contraposição à abordagem anterior, esta se concentra no treinamento exclusivo dos modelos utilizando dados do domínio alvo. O treinamento nos dados específicos do domínio pode proporcionar benefícios durante a avaliação, devido ao compartilhamento de características entre as bases de treinamento e validação. No entanto, devido às especificidades dessas bases, que derivam de seus contextos particulares, os modelos treinados nelas podem apresentar desafios relacionados à representatividade dos dados e à generalização do conhecimento adquirido durante o treinamento. Para essa abordagem, as bases FAQUAD e OrgAcadQA serão utilizadas para o treinamento dos modelos em seus respectivos experimentos.

iii) ***Fine-tuning* em Dados de Domínio Aberto e Específico**: Essa abordagem fundamenta-se na estratégia *two-stage fine-tuning* (Li; Rudzicz, 2021). Inicialmente, ocorre o *fine-tuning* do modelo em uma base de dados ampla de domínio aberto, contendo um volume significativo de informações. Posteriormente, é conduzida uma segunda etapa de *fine-tuning*, utilizando uma base de dados específica do domínio em questão. O propósito dessa estratégia é aprimorar o desempenho do modelo nos dados específicos do domínio, utilizando como base os conhecimentos adquiridos durante o treinamento na base de domínio aberto. A Figura 7 ilustra as etapas da execução da técnica *two-stage fine-tuning*.

[11] https://optuna.readthedocs.io/en/stable/reference/generated/optuna.pruners.PatientPruner.html
[12] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html

Figura 7. Etapas na aplicação da estrategia *two-stage fine-tuning*.



4. Experimentos e Resultados

Nesta seção, serão abordados os principais aspectos relacionados ao treinamento dos modelos BERTimbau para a tarefa de QA, explorando diferentes estratégias e conjuntos de treinamento. Adicionalmente, serão apresentados os resultados provenientes dos experimentos conduzidos, bem como a conclusão do desenvolvimento do chatbot.

Todos os experimentos descritos nesta seção foram realizados utilizando a plataforma Google Colab, com as seguintes especificações de hardware: GPU NVIDIA® T4 de 15 GB, CPU Intel(R) Xeon(R) @ 2,20 GHz, 12,7 GB de memória RAM e 107,7 GB de armazenamento em disco.

4.1. Treinamento dos Modelos na Tarefa de QA na Base SQuAD V1.1-PT-BR

Para o treinamento dos modelos BERTimbau na base SQuAD V1.1-PT-BR, foram utilizados como base os valores dos hiperparâmetros definidos por (Devlin *et al.*, 2019) no treinamento dos modelos BERT na base SQuAD v1.1.

O treinamento dos modelos foi realizado por 3 épocas, com os seguintes hiperparâmetros: *learn_rate* de 5e-5, *weight_decay* de 0,01 e *batch_size* de 8. Em relação aos hiperparâmetros utilizados na etapa de pré-processamento, foram definidos os valores de *max_seq_length* como 384 e *doc_stride* de 128. Os valores dos hiperparâmetros *batch_size* e *max_seq_length* precisaram ser reduzidos em relação aos utilizados originalmente no treinamento dos modelos BERT, devido à limitações nos recursos computacionais do ambiente de treinamento. Para assegurar a reprodutibilidade dos experimentos, os hiperparâmetros *seed* e *data_seed* foram fixadas como 42.

Na condução dos experimentos, utilizou-se o *script* run_qa.py[13], disponibilizado pela biblioteca Transformers. Esse *script* foi elaborado com o propósito de treinar e avaliar modelos destinados à tarefa de QA em bases que possuem um formato compatível com o da base SQuAD v1.1. Os resultados obtidos durante o treinamento dos modelos BERTimbau na base SQuAD v1.1-PT-BR estão apresentados na Tabela 3.

Apesar de o modelo $\text{BERTimbau}_{\text{Large}}$ possuir mais do que o dobro do número de parâmetros do $\text{BERTimbau}_{\text{Base}}$, os resultados de ambos são bastante próximos, variando aproximadamente 2% para ambas as métricas de EM e F1. Notavelmente, o $\text{BERTimbau}_{\text{Large}}$ apresenta um desempenho superior ao ser avaliado na base SQuAD v1.1-PT-BR.

[13]https://github.com/huggingface/transformers/blob/main/examples/pytorch/question-answering/run_qa.py

Tabela 3. Comparação dos resultados obtidos pelos modelos BERTimbau para a tarefa de QA na base SQuAD-v1.1-PT-BR

Modelo	F1	EM
$\text{BERTimbau}_{\text{Base}}$	0,8224	0,7023
$\text{BERTimbau}_{\text{Large}}$	**0,8438**	**0,7263**

4.2. Aplicação dos Modelos em Dados de Domínio do Ensino Superior

Os experimentos conduzidos utilizando o FAQUAD seguiram a mesma metodologia estabelecida por (Sayama; Araujo; Fernandes, 2019). Para isso, foram utilizados os *scripts*[14] disponibilizados por seus autores para realização do pré-processamento dos dados da base. Assim como realizado originalmente para avaliação do FAQUAD, foi utilizado o algoritmo GroupKFold para realização da validação cruzada, com o parâmetro *n_splits* definido como 10.

A otimização de hiperparâmetros foi realizada utilizando os espaços de busca definidos na Tabela 4. Os modelos foram treinados por 20 épocas para cada combinação de hiperparâmetros. O PatientPruner foi configurado com os parâmetros *patience* igual a 3 e um *min_delta* de 0,1. Por fim, os melhores resultados para o modelo $\text{BERTimbau}_{\text{Base}}$ foram obtidos com a combinação de *batch_size* de 2, *learn_rate* de 1e-5 e *weight_decay* de 0,01. Já para o modelo $\text{BERTimbau}_{\text{Large}}$, *batch_size* de 4, *learn_rate* de 2e-5 e *weight_decay* de 0,01.

Tabela 4. Configurações utilizada durante a otimização de hiperparâmetros utilizando a abordagem GridSearch

Hiperparâmetro	**Espaço de Busca**
batch_size	{2, 4, 8}
learn_rate	{1e-5, 2e-5, 5e-5}
weight_decay	{0,01, 0,1, 0,2}

As Tabelas 5 e 6 apresentam os resultados de média e desvio padrão para as métricas F1 e EM, obtidos a partir da avaliação de diferentes configurações de treinamento dos modelos $\text{BERTimbau}_{\text{Base}}$ e $\text{BERTimbau}_{\text{Large}}$ em diferentes conjuntos de treinamento e validação. Para as abordagens que utilizaram a base SQuAD v1.1-PT-BR na etapa de *fine-tuning*, foram empregados os modelos provenientes dos experimentos descritos na Seção 4.1. Para o *fine-tuning* e avaliação na base FAQUAD, foram realizadas 10 rodadas de avaliação utilizando diferentes combinações de conjuntos de treinamento e teste, aplicando a estratégia de validação cruzada. Em cada rodada, a melhor combinação de hiperparâmetros, identificada durante a otimização, foi utilizada para cada modelo, enquanto os demais hiperparâmetros permaneceram com os valores utilizados no experimento da Seção 4.1.

Semelhante aos resultados obtidos no treinamento na base SQuAD v1.1-PT-BR, ambas as variantes dos modelos BERTimbau demonstraram resultados bastante próximos em todas as configurações de treinamento, variando em média aproximadamente 2%. Destaque-se que o modelo $\text{BERTimbau}_{\text{Large}}$ apresentou as melhores performances.

Ao ser realizada a comparação dos resultados obtidos entre as diferentes configurações de treinamento, é possível observar que o modelo submetido ao *fine-tuning* exclusivo na FAQUAD exibiu os valores mais baixos para mas métricas F1 e EM. Isso pode ser atribuído ao volume restrito de dados

[14] https://github.com/liafacom/faquad

Tabela 5. Comparação da performance de diferentes configurações de treinamento do modelo BERTimbau$_{Base}$ avaliados na base FAQUAD, através da estratégia de validação cruzada

Base de *Fine-tuning*	F1	EM
SQuAD v1.1-PT-BR	$0{,}8689 \pm 0{,}0220$	$0{,}6722 \pm 0{,}0659$
FAQUAD	$0{,}7395 \pm 0{,}0388$	$0{,}5144 \pm 0{,}0463$
SQuAD v1.1-PT-BR + FAQUAD	$\mathbf{0{,}8998 \pm 0{,}0244}$	$\mathbf{0{,}7456 \pm 0{,}0514}$

Tabela 6. Comparação da performance de diferentes configurações de treinamento do modelo BERTimbau$_{Large}$ avaliados na base FAQUAD, através da estratégia de validação cruzada

Base de *Fine-tuning*	F1	EM
SQuAD v1.1-PT-BR	$0{,}8878 \pm 0{,}0233$	$0{,}6944 \pm 0{,}0475$
FAQUAD	$0{,}7565 \pm 0{,}0437$	$0{,}5622 \pm 0{,}0591$
SQuAD v1.1-PT-BR + FAQUAD	$\mathbf{0{,}9065 \pm 0{,}0203}$	$\mathbf{0{,}7656 \pm 0{,}0341}$

nessa base, limitando a representatividade dos dados e, consequentemente, reduzindo a capacidade de generalização do modelo treinado exclusivamente nesse conjunto.

Posteriormente, o modelo treinado na base SQuAD v1.1-PT-BR apresentou melhorias significativas, com um aumento de 17,35% no F1 e 23,51% no EM. Essa base, rica em dados de domínio aberto, permitiu ao modelo ajustar seus pesos de maneira satisfatória para a tarefa de QA, evidenciando sua capacidade de generalização mesmo quando exposto a uma base que possui características distintas daquela em que foi o modelo foi inicialmente treinado.

Por fim, o modelo que seguiu a estratégia de *two-stage fine-tuning* alcançou os melhores resultados em ambas as métricas. Este modelo apresentou um aumento de 19,04% no F1 e 36,33% no EM em comparação com o modelo treinado exclusivamente na SQuAD v1.1-PT-BR. Esses resultados destacam que, para esse contexto específico, o modelo se beneficiou significativamente de uma etapa adicional de *fine-tuning* em dados específicos do domínio após ter sido inicialmente treinado em uma base de domínio aberto com um amplo volume de dados e alta representatividade.

A Tabela 7 apresenta uma comparação dos melhores resultados alcançados pelos modelos BERTimbau em relação aos resultados apresentados no artigo do FAQUAD. Os resultados evidenciam que ambos os modelos BERTimbau superam as performances obtidas pelo autor ao utilizar o BIDAF em conjunto com as representações pré-treinadas do modelo ELMO.

Tabela 7. Comparação dos melhores resultados obtidos pelos modelos BERTimbau com os apresentados no artigo do FAQUAD

Base de *Fine-tuning*	F1	EM
BIDAF$_{ELMO}$(Sayama; Araujo; Fernandes, 2019)	$0{,}4398 \pm 0{,}0483$	$0{,}2456 \pm 0{,}0371$
BERTimbau$_{Base}$ *two-stage fine-tuning*	$0{,}8998 \pm 0{,}0244$	$0{,}7456 \pm 0{,}0514$
BERTimbau$_{Large}$ *two-stage fine-tuning*	$\mathbf{0{,}9065 \pm 0{,}0203}$	$\mathbf{0{,}7656 \pm 0{,}0341}$

4.3. Avaliação dos Modelos no Contexto da Organização Acadêmica

A base OrgAcadQA[15] foi construída a partir dos dados de 5 capítulos do documento da Organização Acadêmica, conforme detalhado na Seção 3.2. Durante a otimização dos hiperparâmetros dos modelos, adotou-se a estratégia de validação cruzada. Para isso, foi utilizado o GroupKFold, com o parâmetro *n_splits* igual a 5. Nesse contexto, a cada iteração do GroupKFold, o modelo foi treinado com dados provenientes de 4 capítulos e avaliado com o restante dos dados. Esse processo foi repetido até que o modelo fosse avaliado em cada um dos 5 capítulos da base, garantindo uma avaliação precisa e robusta. Os hiperparâmetros otimizados e o espaço de busca estão detalhados na Tabela 4. Destaca-se que a melhor combinação de hiperparâmetros foi consistente para ambos os modelos, sendo *batch_size* igual a 2, *learning_rate* de 1e-5 e *weight_decay* de 0,01.

Devido ao extenso tamanho dos capítulos da Organização Acadêmica, os contextos na base OrgAcadQA ultrapassam o número máximo de *tokens* estabelecido no hiperparâmetro *max_seq_length*. Diante dessa situação, a estratégia de segmentação torna-se essencial, permitindo que os modelos lidem com contextos longos sem perda de informação.

Os experimentos conduzidos na Seção 4.2 foram replicados para a base OrgAcadQA. Os resultados desse experimento estão apresentados nas Tabelas 8 e 9.

Tabela 8. Comparação da performance de diferentes configurações de treinamento do modelo BERTimbau$_{\text{Base}}$ avaliados na base OrgAcadQA, através da estratégia de validação cruzada

Base de *Fine-tuning*	F1	EM
SQuAD v1.1-PT-BR	$0{,}7514 \pm 0{,}1107$	$0{,}5860 \pm 0{,}1466$
OrgAcadQA	$0{,}4125 \pm 0{,}0419$	$0{,}1733 \pm 0{,}0641$
SQuAD v1.1-PT-BR + OrgAcadQA	$\mathbf{0{,}8533 \pm 0{,}0821}$	$\mathbf{0{,}7533 \pm 0{,}1210}$

Tabela 9. Comparação da performance de diferentes configurações de treinamento do modelo BERTimbau$_{\text{Large}}$ avaliados na base OrgAcadQA, através da estratégia de validação cruzada

Base de *Fine-tuning*	F1	EM
SQuAD v1.1-PT-BR	$0{,}7568 \pm 0{,}0903$	$0{,}6400 \pm 0{,}1084$
OrgAcadQA	$0{,}3013 \pm 0{,}1384$	$0{,}1200 \pm 0{,}1043$
SQuAD v1.1-PT-BR + OrgAcadQA	$\mathbf{0{,}8808 \pm 0{,}0553}$	$\mathbf{0{,}7820 \pm 0{,}0249}$

Assim como observado nos resultados dos experimentos com o FAQUAD, os modelos treinados exclusivamente na base de domínio específico exibiram os piores desempenhos em relação às métricas F1 e EM. Nessa configuração, o BERTimbau$_{\text{Base}}$ demonstrou resultados superiores e mais consistentes do que o BERTimbau$_{\text{Large}}$ em ambas as métricas. Esses resultados podem ser atribuídos ao reduzido volume de dados disponíveis na base, o que compromete a representatividade de seus dados, influenciando negativamente o processo de aprendizagem dos modelos.

Para os modelos treinados na base de domínio aberto, os resultados foram similaridade na pontuação F1, com ambos os modelos atingindo aproximadamente 0,75. A principal disparidade surge na métrica de EM, onde o BERTimbau$_{\text{Base}}$ registrou 0,58, enquanto o BERTimbau$_{\text{Large}}$ obteve 0,64, demonstrando ter conseguido responder de forma mais assertiva as respostas presentes na base

[15] https://huggingface.co/datasets/alexemanuel27/orgacadqa

Comparado à abordagem anterior, o modelo BERTimbau$_{\text{Large}}$ apresentou um aumento de cerca de 0,45 pontos no F1 e 0,52 no EM. Esses resultados indicam a habilidade do modelo em generalizar os conhecimentos adquiridos durante o treinamento, resultando em desempenho sólido mesmo quando avaliado em uma base com características e contextos distintos.

Por fim, a abordagem *two-stage fine-tuning* se mantém com os melhores resultados para ambos os modelos, sendo o BERTimbau$_{\text{Large}}$ superior ao BERTimbau$_{\text{Base}}$ por aproximadamente 3% em ambas as métricas. Além disso, o BERTimbau$_{\text{Large}}$ demonstrou um aumento de 17% na pontuação F1 e 34% de EM em relação ao modelo treinado apenas nos dados e domínio aberto. A estratégia two-stage fine-tuning se destaca entre as abordagens de treinamento exploradas neste trabalho como a mais eficiente para treinar modelos de QA especializados em domínios específicos, especialmente em cenários com dados de treinamento limitados no domínio alvo.

Embora o modelo BERTimbau$_{\text{Large}}$ tenha apresentado os melhores resultados de forma geral, apenas ficando atrás nos resultados do modelo treinado exclusivamente na base OrgAcadQA, é crucial destacar a diferença no número de parâmetros entre os dois modelos. O BERTimbau$_{\text{Large}}$ possui 340 milhões de parâmetros treináveis, enquanto o BERTimbau$_{\text{Base}}$ conta com 110 milhões. Isso implica que os recursos computacionais empregados no treinamento e inferência do BERTimbau$_{\text{Large}}$ são substancialmente superiores em comparação com os utilizados pelo BERTimbau$_{\text{Base}}$. Dessa forma, torna-se necessário conduzir um estudo mais aprofundado para analisar se os ganhos obtidos pelo modelo BERTimbau$_{\text{Large}}$ são significativos o suficiente para justificar sua implementação em um determinado contexto.

A Figura 8 demostra a aplicação prática do chatbot no esclarecimento de dúvidas relacionadas aos tópicos abordados na Organização Acadêmica. O usuário inicia o processo selecionando um tópico específico relacionado à sua pergunta e, em seguida, introduz a questão desejada. Posteriormente, o chatbot realiza a comunicação com a aplicação servidor através de uma API. Por fim, a API retorna a resposta predita pelo modelo BERTimbau para o usuário.

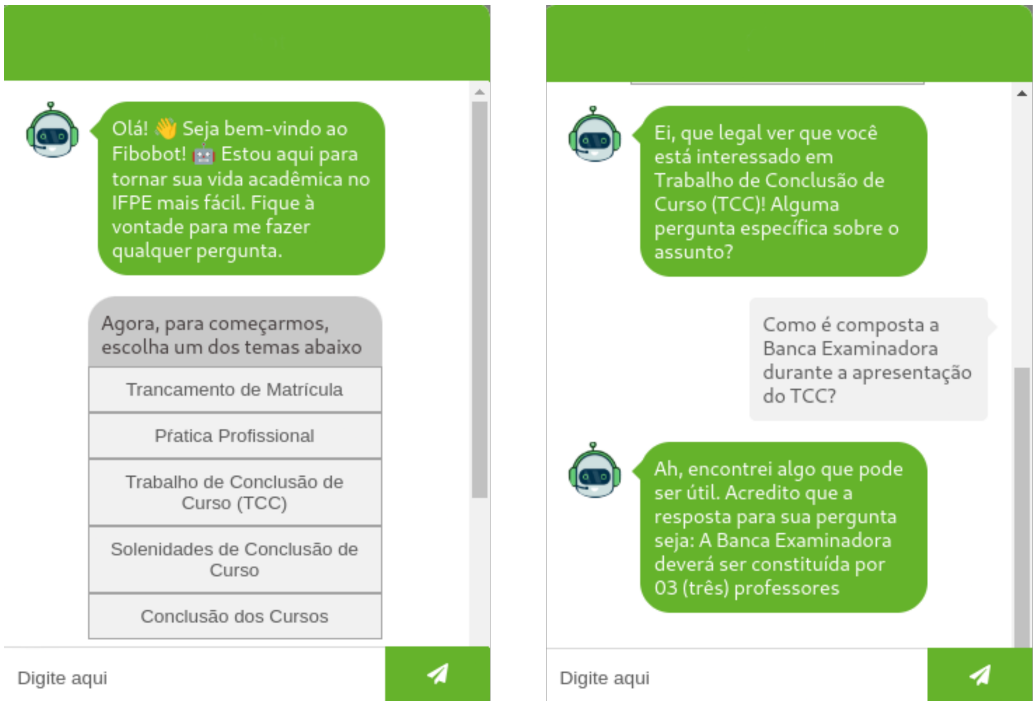
5. Conclusão

Este trabalho propôs o desenvolvimento de um sistema de chatbot inteligente dedicado a responder perguntas relacionadas aos tópicos da Organização Acadêmica, simplificando assim o acesso às informações presentes neste documento. Para alcançar esse objetivo, foram utilizados modelos BERTimbau treinados para a tarefa de QA utilizando a abordagem de MRC.

A aplicação dos modelos BERTimbau evidenciou resultados promissores na execução da tarefa de QA ao serem treinados na base SQuAD v1.1-PT-BR. Mesmo quando avaliados em um conjunto de dados de domínio específico, com características distintas, os modelos demonstraram desempenhos consistentes. Adicionalmente, a adoção da estratégia *two-staging fine-tuning* impulsionou significativamente o desempenho desses modelos ao serem treinados e avaliados em bases de domínio especifico, como é o caso do FAQUAD. Os resultados alcançados com essa estratégia superaram tanto os obtidos ao utilizar um modelo previamente treinado em uma base ampla de domínio aberto quanto os modelos treinados exclusivamente na base de domínio específico.

Para o treinamento e validação dos modelos no contexto da Organização Acadêmica, foi desenvolvida a base OrgAcadQA. Esta base segue o mesmo formato do SQuAD v1.1, sendo composta por 100 perguntas e respostas anotadas manualmente, provenientes de 5 capítulos da organização acadêmica. Como evidenciado nos experimentos conduzidos com o FAQUAD, a estratégia de *two-staging fine-tuning* destacou-se como a mais eficaz, proporcionando melhorias de até 17% em F1 e

Figura 8. Interação entre chatbot e usuário na resolução de dúvida a respeito de um dos capítulos da Organização Acadêmica



34% em EM na base OrgAcadQA, quando comparada ao modelo treinado na base de domínio aberto. Esses resultados fortalecem a ideia de que essa abordagem é benéfica para o treinamento de modelos em domínios específicos, especialmente quando há um número limitado de amostras, como ocorre na base OrgAcadQA, que devido ao seu tamanho reduzido, torna-se insuficiente para o treinamento de modelos robustos, como os BERTimbau.

Embora o estudo tenha alcançado com êxito os objetivos propostos inicialmente, apresentando resultados promissores no desenvolvimento de um chatbot inteligente no contexto da Organização Acadêmica, é importante destacar algumas limitações identificadas no sistema desenvolvido. O sistema atual é incapaz de identificar o contexto das perguntas dos usuários, exigindo que estes selecionem o tema ao qual suas perguntas se relacionam. Adicionalmente, outra limitação refere-se à incapacidade do modelo de discernir se a pergunta pode ou não ser respondida, resultando no modelo sempre fornecer uma resposta, mesmo em casos em que o contexto escolhido não contém a informação necessária para responder à pergunta realizada.

Como parte dos trabalhos futuros, propõe-se a ampliação da base OrgAcadQA através da inclusão dos dados referentes aos demais capítulos da Organização Acadêmica, visando a criação de uma base mais completa e robusta para o treinamento de modelos de QA no contexto da Organização Acadêmica. Adicionalmente, planeja-se explorar a aplicação de algoritmos de Recuperação de Informação na identificação de parágrafos que possam conter as respostas para as perguntas realizadas pelos usuários. Dessa forma, retirando do usuário a responsabilidade de escolha de um tópico para sua pergunta, simplificando o processo e tornando-o mais prático e simples para o usuário.

Referências

AKIBA, T. *et al*. Optuna: A next-generation hyperparameter optimization framework. In:

Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.: s.n.], 2019. 14

ALLAM, A. M. N.; HAGGAG, M. H. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, v. 2, n. 3, 2012. 4

ATHOTA, L. *et al*. Chatbot for healthcare system using artificial intelligence. In: *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. [S.l.: s.n.], 2020. p. 619–622. 2

Calijorne Soares, M. A.; PARREIRAS, F. S. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University - Computer and Information Sciences*, v. 32, n. 6, p. 635–646, 2020. ISSN 1319-1578. Disponível em: https://www.sciencedirect.com/science/article/pii/S131915781830082X. Acesso em: 05 set. 2024. 4

CHEN, J. *et al*. An Empirical Survey of Data Augmentation for Limited Data Learning in NLP. *Transactions of the Association for Computational Linguistics*, v. 11, p. 191–211, 03 2023. ISSN 2307-387X. Disponível em: https://doi.org/10.1162/tacl_a_00542. Acesso em: 05 set. 2024. 3

CHEN, Y.; ZULKERNINE, F. Bird-qa: A bert-based information retrieval approach to domain specific question answering. In: . [S.l.: s.n.], 2021. p. 3503–3510. 6, 7

CHOWDHARY, K. R. Natural language processing. In: ____. *Fundamentals of Artificial Intelligence*. New Delhi: Springer India, 2020. p. 603–649. ISBN 978-81-322-3972-7. Disponível em: https://doi.org/10.1007/978-81-322-3972-7_19. Acesso em: 05 set. 2024. 2

CLARIZIA, F. *et al*. Chatbot: An education support system for student. In: CASTIGLIONE, A. *et al*. (Ed.). *Cyberspace Safety and Security*. Cham: Springer International Publishing, 2018. p. 291–302. 2

COLLARANA, D. *et al*. A question answering system on regulatory documents. In: *International Conference on Legal Knowledge and Information Systems*. [s.n.], 2018. Disponível em: https://api.semanticscholar.org/CorpusID:55702047. Acesso em: 05 set. 2024. 5

CSAKY, R. Deep learning based chatbot models. *ArXiv*, abs/1908.08835, 2019. 2

DEVLIN, J. *et al*. Bert: Pre-training of deep bidirectional transformers for language understanding. In: *North American Chapter of the Association for Computational Linguistics*. [s.n.], 2019. Disponível em: https://api.semanticscholar.org/CorpusID:52967399. Acesso em: 05 set. 2024. 3, 12, 16

EDUCAçãO, C. e. T. d. P. Instituto Federal de. *ORGANIZAÇÃO ACADÊMICA INSTITUCIONAL*. [S.l.], 2015. 2

HOWARD, J.; RUDER, S. Universal language model fine-tuning for text classification. In: GUREVYCH, I.; MIYAO, Y. (Ed.). *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 328–339. Disponível em: https://aclanthology.org/P18-1031. Acesso em: 05 set. 2024. 3

KHURANA, D. *et al*. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, v. 82, n. 3, p. 3713–3744, Jan 2023. ISSN 1573-7721. Disponível em: https://doi.org/10.1007/s11042-022-13428-4. Acesso em: 05 set. 2024. 3

LATHKAR, M. *High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python*. [S.l.]: Springer, 2023. 9

LI, B.; RUDZICZ, F. TorontoCL at CMCL 2021 shared task: RoBERTa with multi-stage fine-tuning for eye-tracking prediction. In: CHERSONI, E. *et al*. (Ed.). *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*. Online: Association for Computational Linguistics, 2021. p. 85–89. Disponível em: https://aclanthology.org/2021.cmcl-1.9. Acesso em: 05 set. 2024. 15

LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019. 14

MELLO, G. L. de *et al*. *PeLLE: Encoder-based language models for Brazilian Portuguese based on open data*. 2024. Disponível em: https://arxiv.org/abs/2402.19204. Acesso em: 05 set. 2024. 3

NETO, J. R. *et al*. Chatbot to support frequently asked questions from students in higher education institutions. In: *Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional*. Porto Alegre, RS, Brasil: SBC, 2022. p. 591–601. ISSN 2763-9061. Disponível em: https://sol.sbc.org.br/index.php/eniac/article/view/22815. Acesso em: 05 set. 2024. 5

PEDREGOSA, F. *et al*. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. 15

PRECHELT, L. Early stopping-but when? In: *Neural Networks: Tricks of the trade*. [S.l.]: Springer, 2002. p. 55–69. 14

RAJPURKAR, P. *et al*. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. 2016. 5

SAYAMA, H. F.; ARAUJO, A. V.; FERNANDES, E. R. Faquad: Reading comprehension dataset in the domain of brazilian higher education. In: *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2019. p. 443–448. 7, 17, 18

SHARMA, V.; TIWARI, A. K. A study on user interface and user experience designs and its tools. *World Journal of Research and Review (WJRR)*, v. 12, n. 6, p. 41–45, 2021. 8

SILVA, E. H. M. D.; LATERZA, J.; FALEIROS, T. de P. New state-of-the-art for question answering on portuguese squad v1.1. *Anais do X Symposium on Knowledge Discovery, Mining and Learning (KDMiLe 2022)*, 2022. Disponível em: https://api.semanticscholar.org/CorpusID:259755828. Acesso em: 05 set. 2024. 3

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: Pretrained bert models for brazilian portuguese. In: CERRI, R.; PRATI, R. C. (Ed.). *Intelligent Systems*. Cham: Springer International Publishing, 2020. p. 403–417. ISBN 978-3-030-61377-8. 3

VASWANI, A. *et al*. Attention is all you need. In: GUYON, I. *et al*. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. Acesso em: 05 set. 2024. 3, 4

WAGNER, J. *et al*. The brwac corpus: A new open resource for brazilian portuguese. In: . [S.l.: s.n.], 2018. 3

WANG, H. *et al*. Pre-trained language models and their applications. *Engineering*, v. 25, p. 51–65, 2023. ISSN 2095-8099. Disponível em: https://www.sciencedirect.com/science/article/pii/S2095809922006324. Acesso em: 05 set. 2024. 3

WOLF, T. *et al*. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2020. 9

WU, Y. *et al*. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144, 2016. Disponível em: https://api.semanticscholar.org/CorpusID:3603249. Acesso em: 05 set. 2024. 13

WUBE, H. D. *et al*. Text-based chatbot in financial sector: a systematic literature review. *Data Sci. Financ. Econ*, v. 2, n. 3, p. 232–259, 2022. 2

ZENG, C. *et al*. A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Applied Sciences*, v. 10, n. 21, 2020. ISSN 2076-3417. Disponível em: https://www.mdpi.com/2076-3417/10/21/7640. Acesso em: 05 set. 2024. 4