Análise de viabilidade para a automação da correção de respostas discursivas com Inteligência Artificial

Elias Iuri Sobieski Dalvite

[1] Instituto Federal Sul-Rio-Grandense - Câmpus Passo Fundo
Bacharelado em Ciência da Computação
Passo Fundo
2024

Abstract. *This project presents an automated system for the correction of essay-type questions, using Natural Language Processing models based on the Transformer architecture. The system is capable of interpreting and evaluating students' responses, comparing them to a model answer prepared by the teacher, thus ensuring fair and accurate assessments. In addition to significantly reducing the time required for grading and providing almost instant feedback to students, the system also relieves the teachers' workload, allowing them to focus on pedagogical activities and individualized support.*

Resumo. *Este trabalho apresenta um sistema automatizado para a correção de questões discursivas, utilizando modelos de Processamento de Linguagem Natural baseados na arquitetura Transformer. O sistema é capaz de interpretar e avaliar as respostas dos alunos, comparando-as com uma resposta modelo elaborada pelo professor, garantindo assim avaliações justas e precisas. Além de reduzir significativamente o tempo necessário para a correção e fornecer feedback quase instantâneo aos estudantes, o sistema também alivia a carga de trabalho dos professores, permitindo que eles se concentrem em atividades pedagógicas e no suporte individualizado.*

1. Introdução

O uso de questões de múltipla escolha é amplamente difundido na comunidade acadêmica, constituindo uma das principais escolhas em avaliações somativas. Provas objetivas de múltipla escolha são utilizadas, sobretudo, em exames como vestibulares, concursos e provas finais de cursos de graduação. Essa ampla difusão se justifica pelo fato de os exames compostos por esse tipo de questão preencherem mais completamente os requisitos de validade e fidedignidade, além de apresentarem vantagens quanto à praticidade em provas com grande número de candidatos [Bollela et al. 2018]. O Exame Nacional do Ensino Médio (ENEM), por exemplo, é uma prova totalmente objetiva. Contudo, para diminuir o número de ”chutes”, o ENEM utiliza o modelo Teoria de Resposta ao Item (TRI), que se trata de um conjunto de modelos matemáticos que representa a relação entre a probabilidade de o participante responder corretamente a uma questão, seu conhecimento na área de avaliação e as características dos itens [Porto 2023].

Utilizar questões de múltipla escolha também facilita o processo de correção das avaliações, tornando-o prático, rápido e objetivo, o que permite que o estudante tenha acesso ao resultado de suas avaliações em um curto espaço de tempo. Por outro lado,

avaliações que utilizam questões discursivas criam um espaço de reflexão para o estudante, no qual ele pode dissertar sobre determinado tema. No que diz respeito ao processo de correção, observa-se a disponibilidade de uma grande quantidade de ferramentas computacionais para esse fim. No entanto, para a automatização do processo de correção de avaliações com questões discursivas, ainda se observa um número mais restrito de estudos nessa área. Nesse sentido, o presente trabalho desenvolveu uma ferramenta de correção automática de questões discursivas, utilizando técnicas de Processamento de Linguagem Natural (PLN).

2. Revisão Bibliográfica

Os instrumentos de avaliação são essenciais para avaliar o processo de aprendizado dos discentes uma vez que expressam o seu estágio de aprendizado. Ou seja, por meio desses instrumentos é possível mapear o que o aluno aprendeu, deixou de aprender ou ainda precisa aprender. A prova é o instrumento de avaliação mais comumente utilizado na escola. É comum escolas terem grande parte de seu processo avaliativo centrado em provas, visto que possibilita fidedignidade na aprovação do aluno e na devolução dos resultados à comunidade escolar.

O pensamento criativo, especialmente quando medido por meio de tarefas como dissertações e projetos de pesquisa originais, tem um poder preditivo significativo sobre o desempenho acadêmico, superando até mesmo traços de personalidade como a Conscienciosidade. Além disso, os alunos criativos mostram uma clara preferência por métodos de avaliação interativos e criativos, sugerindo que avaliações que incentivam a expressão de soluções originais e inovadoras podem ser mais eficazes na promoção de um desempenho acadêmico de excelência [Chamorro-Premuzic 2006].

O estudo dirigido por [Pepple et al. 2010] revelou que, sem a inclusão de questões discursivas longas, alguns alunos que não passaram poderiam ter sido aprovados se fossem avaliados apenas com as questões de múltipla escolha. Além disso, o estudo sugere que o formato de avaliação pode ter uma influência significativa nas notas finais, com a adição das questões de múltipla escolha resultando em um desempenho superior de maneira geral.

No que diz respeito a elaboração, a prova discursiva é mais fácil de ser elaborada, é possui menos questões, implica em menos tempo para elaboração e é apresentada com questões mais gerais e com respostas amplas. A prova objetiva é mais difícil de ser elaborada, requer mais tempo do elaborador, possui mais questões, sendo essas mais específicas de respostas breves. No que diz respeito a correção, a prova discursiva é considerada mais difícil, exige mais tempo, retarda a possibilidade de retorno dos resultados e a distribuição das notas é controlada pelo professor. Na prova objetiva, a correção é mais fácil, exige menos tempo, oferta possibilidade de retorno dos resultados imediato e a distribuição das notas é determinada pela própria prova [Rampazzo 2010]. Nesse contexto o presente trabalho endereça a questão da correção de provas discursivas ao propor o desenvolvimento de um sistema que permite automatizar o processo de correção de avaliações utilizando técnicas de processo amento de texto baseadas em Inteligencia Artificial.

2.1. Inteligencia Artificial

A Inteligência Artificial (IA) é uma tecnologia avançada que capacita as máquinas a realizar trabalhos racionais e tomadas de decisões, sendo capaz de dirigir carros, diagnos-

ticar doenças, reconhecer padrões em imagens e fornecer recomendações personalizadas. Desta forma, é possível identificar que as IAs têm impulsionado uma revolução em várias áreas, aproveitando técnicas como Redes Neurais Artificiais e Aprendizado de Máquina para treinar os sistemas a reconhecerem padrões, assim alcançando a habilidade de aprender a partir de dados e solucionar problemas de maneiras eficientes [Ludermir 2021].

2.2. Processamento de Linguagem Natural (PLN)

O processamento da linguagem natural (PLN) trata computacionalmente os diversos aspectos comunicação humana, como sons, palavras, sentenças e discursos, considerando formatos e referências, estruturas e significados, contextos e usos. Em outras palavras o PLN permite ao computador se comunicar em linguagem humana, o que vem de encontro com a necessidade do projeto. Esse processo de comunicação pode ocorrer nos seguintes níveis [Gonzalez and Lima 2003]:

- **Fonético:** referente a relação entre as palavras e os sons que elas emitem;
- **Morfológico:** referente aa construção das palavras a partir unidades de significado primitivas e de como classificá-las em categorias morfológicas;
- **Sintático:** refererente ao relacionamento das palavras entre si, cada uma assumindo seu papel estrutural nas frases;
- **Semântico:** referente ao relacionamento das palavras com seus significados e de como eles são combinados para formar os significados das sentenças;
- **Pragmático:** do uso de frases e sentenças em diferentes contextos, afetando o significado.

O Processamento de Linguagem Natural também é considerada uma área do aprendizado de máquina que permite que um computador entenda, analise, manipule e, potencialmente, gere linguagem humana. A técnica de PLN envolve pré-processamento de dados e incorporação de palavras. Com o uso de técnicas de aprendizado profundo, o PLN teve avanços significativos nos últimos anos. A linguagem natural deve ser transformada em uma estrutura matemática para que as máquinas compreendam o texto [Mridha et al. 2021].

2.3. Transformers

Introduzido por [Vaswani 2017], o modelo Transformer revolucionou o campo do Processamento de Linguagem Natural. Substituindo as redes neurais recorrentes (RNNs), que enfrentavam dificuldades em capturar dependências de longo prazo, os Transformers proporcionaram melhorias significativas na qualidade e precisão de tarefas como tradução multilíngue, oferecendo resultados mais precisos e uma melhor capacidade de lidar com contextos complexos [Pimentel 2023].

Com a chegada deste modelo, a área de tradução automática, por exemplo, passou a apresentar resultados significativamente melhores, possibilitando a criação de modelos com maior capacidade de lidar com contextos complexos e produzir traduções mais precisas, iniciando uma nova era nesta área da IA, com modelos cada vez mais sotisficados e precisos [Iosifova et al. 2020].

O modelo se destaca por sua capacidade de escalar eficientemente com o volume de dados e o tamanho do modelo, facilitar o treinamento paralelo e capturar características

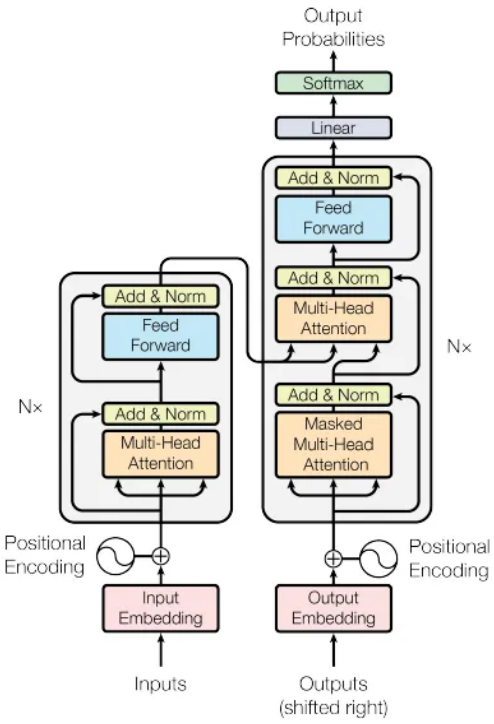


Figura 1. Arquitetura dos transformers

de longas sequências. Além disso, supera redes neurais convolucionais em desempenho, oferecendo resultados superiores em diversas tarefas. [Wolf et al. 2019]. Seu modelo de arquitetura pode ser visualizado na Figura 1.

Em conclusão, a utilização de transformers demonstrou ser uma tecnologia revolucionária no processamento de linguagem natural. Ao permitir que todas as palavras em uma frase sejam processadas simultaneamente, os transformers não apenas aceleram o tempo de treinamento dos modelos, mas também garantem uma compreensão mais profunda de frases complexas. Essa abordagem evita a perda de informações cruciais, tornando-os especialmente eficazes na interpretação de contextos longos e intricados.

Entre os modelos que utilizam a arquitetura Transformer e têm revolucionado o Processamento de Linguagem Natural, destacam-se o BERT (Bidirectional Encoder Representations from Transformers), que foca na compreensão bidirecional do contexto, o T5 (Text-to-Text Transfer Transformer), que converte todas as tarefas de PLN em problemas de transformação de texto, e o GPT (Generative Pre-trained Transformer), conhecido por sua habilidade de geração de texto fluente e contextual.

O GPT-3, por exemplo, com seus 175 bilhões de parâmetros, demonstrou capacidades impressionantes em tarefas como tradução, geração de código, e compreensão de linguagem, consolidando a eficácia da abordagem baseada em Transformers.

Atualmente, o GPT-4, a versão mais avançada da família GPT, o qual foi o escolhido para ser testado inicialmente nesta aplicação.

2.4. Métrica Kappa de Cohen

Proposto por Jacob Cohen em 1960, o coeficiente Kappa é um método estatístico utilizado para medir o nível de concordância ou consistência entre dois conjuntos de dados. Ele é comumente empregado para avaliar a concordância entre avaliadores que estão analisando o mesmo objeto ou situação. O coeficiente de Kappa é geralmente considerado uma medida mais confiável do que o cálculo simples da taxa de concordância percentual, pois leva em conta a possibilidade de os avaliadores concordarem por acaso [Silva 2023].

Assim como os coeficientes de correlação, o coeficiente Kappa de Cohen pode variar de 0,00 a 1,00, sendo que 0 indica o nível de concordância que seria esperado apenas por acaso, enquanto 1,00 representa uma concordância perfeita entre os avaliadores.

O Kappa ponderado é uma variação do Kappa de Cohen que possibilita o uso de esquemas de ponderação para considerar a proximidade entre as categorias de concordância. Ele é especialmente útil em situações com variáveis ordinais ou classificadas. O objetivo desse coeficiente é diferenciar uma discordância significativa (por exemplo, quando um avaliador classifica uma questão como 1 e outro como 3) de uma discordância mais leve (como quando um avaliador classifica uma questão como 1 e outro como 2) [Silva 2023].

2.5. Tecnologias utilizadas

2.5.1. Python

Python foi a linguagem de programação utilizada para desenvolver o sistema. O Python se destaca por sua simplicidade e orientação para o desenvolvimento ágil, sendo aplicável tanto no ambiente empresarial quanto em setores específicos, como desenvolvimento científico, geoprocessamento e aplicativos móveis. Além disso, Python não possui uma finalidade específica, permitindo sua aplicação conforme o propósito do usuário, diferentemente de linguagens como PHP, direcionado para desenvolvimento web, ou Java SE, voltado para aplicações de desktop [Amorim et al. 2023].

Nos últimos 5-10 anos, a inteligência artificial tem utilizado quase exclusivamente a linguagem de programação Python, que surgiu no século passado. Isso ocorreu devido à necessidade de processar grandes volumes de dados para o avanço da civilização, ao fato de que o Python adquiriu todas as ferramentas necessárias para resolver problemas na área de Inteligência Artificial, e à facilidade de aprendizado e uso da linguagem, que a tornou popular entre os programadores e criou uma grande comunidade de profissionais na área. Esses fatores fizeram do Python e das tecnologias de inteligência artificial uma combinação quase inseparável [Zulunov and Soliev 2023].

Em resumo, a versatilidade e a robustez do Python têm permitido sua aplicação em uma ampla variedade de contextos e setores, reforçando sua posição como uma ferramenta essencial tanto para o desenvolvimento ágil de software quanto para as complexas demandas da inteligência artificial. Essa adaptabilidade contínua de Python assegura seu papel vital na inovação tecnológica e no avanço científico.

2.5.2. Flutter

Flutter é um framework multiplataforma desenvolvido pelo Google, lançado em 2016, para criar aplicativos móveis de alto desempenho em Android, iOS e Fuchsia. Diferente de outros frameworks que dependem de visualizações web (web views), o Flutter utiliza seu próprio motor de renderização de alto desempenho e widgets nativos, permitindo uma performance próxima à de aplicativos nativos. Ele compila o código Dart para código nativo utilizando o NDK do Android e o LLVM para iOS. Uma funcionalidade chave é o "Stateful Hot Reload", que acelera o desenvolvimento ao atualizar o código do aplicativo em tempo real sem alterar seu estado ou estrutura, preservando transições e ações. O Flutter também é central para a estratégia do Google em sistemas operacionais de próxima geração [Tashildar et al. 2020] .

O Flutter será utilizado para desenvolver a interface gráfica do sistema devido à sua capacidade de criar aplicativos multiplataforma com alta performance e aparência nativa. Com sua arquitetura baseada em widgets personalizados e um motor de renderização poderoso, o Flutter permitirá construir uma interface gráfica que seja fluida, responsiva e esteticamente agradável, atendendo às necessidades específicas do sistema.

2.5.3. FastAPI

O FastAPI foi a tecnologia escolhida para o desenvolvimento do backend do sistema por ser um framework de backend intuitivo para a construção de APIs com Python 3.8+. Baseado em type hints padrão do Python e compatível com documentação automatizada via OpenAPI, ele simplifica o desenvolvimento com validação automática de dados, minimização de código redundante e design intuitivo baseado em type hints. Suas capacidades assíncronas garantem escalabilidade e eficiência, tornando-o ideal para projetos modernos. A ampla comunidade e recursos do Python também contribuem para sua popularidade em projetos de backend de diferentes tamanhos e complexidades [Appareddy et al. 2023].

2.5.4. ChatGPT

O ChatGPT é um modelo de linguagem desenvolvido pela OpenAI que utiliza técnicas avançadas de inteligência artificial para gerar respostas em linguagem natural a partir de comandos ou entradas fornecidas. Este modelo é projetado para compreender e responder de forma contextual, oferecendo respostas precisas e fluídas, que se assemelham à comunicação humana. Sua aplicação abrange diversas áreas, incluindo processamento de linguagem natural, atendimento ao cliente, criação de conteúdo, cibersegurança, educação e desenvolvimento de software. O impacto do ChatGPT é amplamente reconhecido, pois ele representa um avanço significativo na interação humano-máquina, contribuindo para a evolução de soluções tecnológicas baseadas em inteligência artificial. A capacidade de adaptação e a eficiência do ChatGPT consolidam sua relevância como uma ferramenta essencial para o desenvolvimento de sistemas automatizados modernos [Kalla et al. 2023].

Neste artigo, será utilizado o modelo GPT-4, uma tecnologia central que exemplifica como modelos de linguagem avançados podem gerar respostas em linguagem natural

com base em entradas específicas. Sua aplicação será analisada no contexto de interação humano-máquina e no desenvolvimento de sistemas automatizados, destacando seu impacto e utilidade em áreas como atendimento ao cliente, geração de conteúdo e processamento de linguagem natural. Especificamente, o artigo abordará o uso do ChatGPT no contexto da correção de questões acadêmicas, explorando seu funcionamento, benefícios e limitações. Essa análise reforça sua relevância como uma ferramenta inovadora no campo da inteligência artificial, com potencial para transformar processos educacionais e tecnológicos.

2.6. Trabalhos relacionados

O trabalho realizado por [Silva 2023] propôs um método computacional para a correção automática de questões discursivas, baseado na aplicação de técnicas de Extração da Informação (EI) e no uso de padrões sintáticos. Neste trabalho foi desenvolvido um modelo de Inteligência Artificial que faz toda a análise do texto, através de etapas como a tokenização, normalização, análise léxica e análise sintática a fim de reconhecer padrões para poder avaliar a proximidade da resposta informada pelo aluno em relação à resposta esperada informada pelo docente.

O trabalho desenvolvido por [Oliveira et al. 2020] apresenta um sistema que também requer a resposta do aluno e a resposta esperada, que ao tirar medidas da similaridade entre as respostas, retorna o a nota para o aluno, que parte de 0 até 100, tudo isso através de um pré-processamento do texto realizando a tokenização, a normalização e a remoção de stopwords dos textos de entrada.

[Galhardi et al. 2018] destaca a importância da correção automática de respostas curtas (ASAG) para melhorar as avaliações estudantis. No entanto, há uma escassez de pesquisas utilizando dados em português, especialmente com conjuntos de dados amplos e adequados para abordagens de aprendizado de máquina. Em seu trabalho, Galhardi criou e disponibilizou publicamente um novo conjunto de dados ASAG em português, coletado com a participação de 659 alunos, 14 estudantes de graduação e 13 professores.

Por fim, é importante destacar que plataformas estudadas e desenvolvidas por [Singh et al. 2017] e [Neto et al. 2022] têm em comum o estudo de sistemas que não só avaliam a resposta dada pelo aluno em comparação com a resposta aceita pelo professor, como também propõem um maior foco no feedback dessa resposta para o aluno, contribuindo na etapa da correção dos dados de forma rápida, ao se focar também na elaboração e formalização de um feedback conciso para o aluno avaliado.

Assim, é possível destacar como a área da correção automática de respostas curtas vem recebendo atenção ao longo dos anos, e ainda tem muitos desafios pela frente.

Desta forma, o trabalho atual busca contribuir para este campo, avaliando e testando o modelo de Transformer GPT-4, avaliando quais as facilidades que este modelo de IA traz para esta área de estudo, visto o quão importante é explorar e aprimorar as técnicas de ASAG disponíveis, buscando criar sistemas mais eficientes e precisos, capazes de apoiar educadores e instituições educacionais na tarefa essencial de avaliar o aprendizado dos estudantes.

3. Metodologia

3.1. Estrutura da Aplicação

O sistema foi projetado para ser de fácil utilização e acessível para todos, tendo suporte para estar disponível na web e como aplicativo para smartphone. A parte visual do sistema foi organizada de forma a listar em ordem todos os dados que serão utilizados pelo sistema para realizar a avaliação (Ver Figura 2).

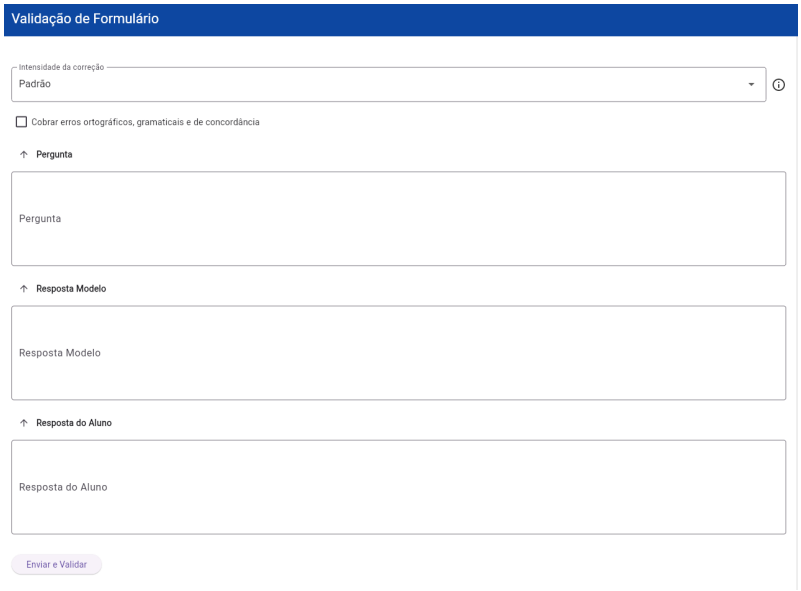


Figura 2. Campos necessários para a correção

A pergunta e a resposta do aluno devem ser informadas da mesma forma que se encontram listadas na avaliação (com erros de concordância/ortográficos aparentes, caso existam).

A resposta modelo deve ser uma resposta muito bem detalhada elaborada pelo professor, baseado nos pontos que ele avaliou ou iria avaliar na questão sendo analisada, exemplo pode ser visualizado na Figura 3.

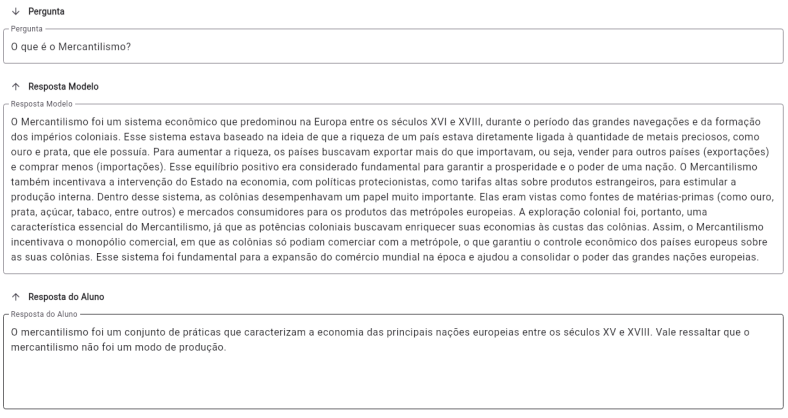


Figura 3. Exemplo de dados de entrada

Após preencher todos os 3 campos, é só pressionar o botão ”Enviar e validar”e aguardar alguns instantes para visualizar a resposta recomendada pela IA (Ver Figura 4).

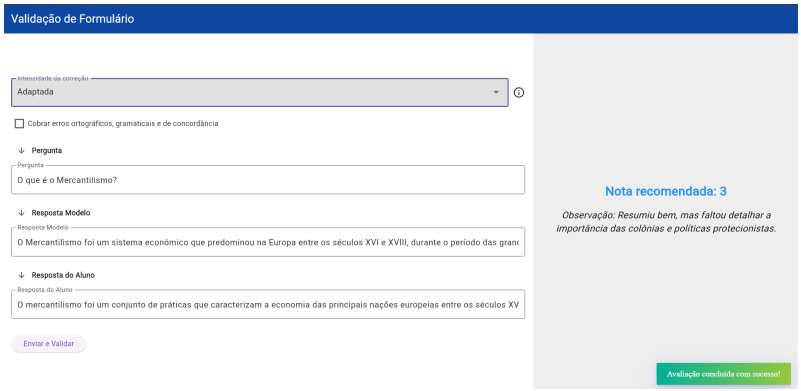


Figura 4. Retorno da avaliação

A nota ”3”representa, neste caso, o valor de 75% (ou 0,75 pontos) atribuídos à questão.

3.2. Fluxo Geral de Funcionamento

O funcionamento do sistema pode ser dividido em etapas (Ver figura 5), descritas a seguir:

- **Entrada de Dados no Frontend:** O usuário insere as informações referentes à avaliação por meio de campos textuais e opções configuráveis. Para melhorar a usabilidade, são utilizados campos expansíveis para textos longos e menus suspensos para a escolha da intensidade da correção.
- **Validação dos Dados:** Antes de enviar os dados ao backend, o formulário realiza validações para garantir que nenhum campo obrigatório esteja vazio ou inválido. Caso ocorra algum erro, mensagens claras e objetivas são exibidas na interface.
- **Envio ao Backend:** Após a validação, os dados são enviados ao backend por meio de uma requisição HTTP POST. A API do backend utiliza um modelo de dados baseado na biblioteca Pydantic para assegurar que as informações estejam no formato correto.
- **Geração de Prompt e Integração com GPT-4:** O backend gera um prompt detalhado com base nos dados fornecidos. Esse prompt é personalizado de acordo com a intensidade de correção selecionada:

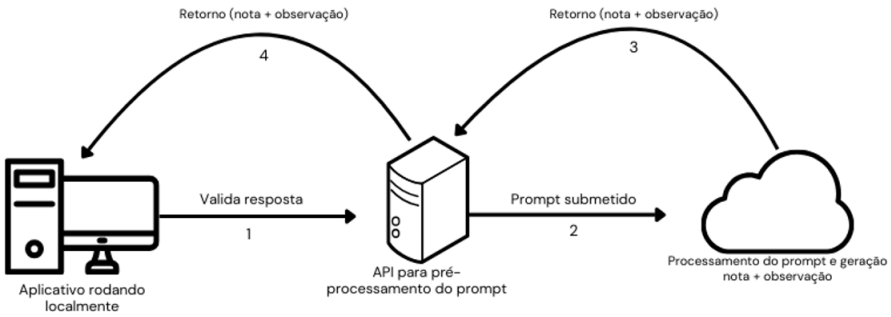


Figura 5. Fluxograma do sistema

Além disso, o prompt pode incluir a orientação de considerar ou ignorar erros ortográficos e gramaticais, dependendo da configuração escolhida pelo usuário. O backend então, realiza a configuração do prompt que será enviado à API da OpenAI, solicitando uma avaliação.

- **Processamento da Resposta da OpenAI:** A API OpenAI retorna uma nota e, uma observação justificando a pontuação. O backend valida a resposta para garantir que esteja no formato esperado e realiza ajustes, se necessário.
- **Exibição dos Resultados:** O backend envia a nota e a observação processadas ao frontend, onde são exibidas em destaque para o usuário. Caso ocorra algum erro durante o processamento, mensagens informativas são apresentadas para orientar o usuário.

3.3. Diferenciação por Intensidade de Correção

O sistema foi projetado para adaptar o nível de rigor da avaliação com base na intensidade selecionada pelo usuário:

1. **Correção Adaptada:** Correção mais aberta, ideal para casos que exigem uma abordagem mais direta e clara.
2. **Correção Padrão:** Correção equilibrada, para casos comuns de correção.
3. **Correção Exigente:** Correção rigorosa, adequada para situações em que o nível de apoio é mais profundo.

Essa configuração permite que o sistema seja utilizado em diferentes contextos educacionais, atendendo às necessidades específicas de cada avaliação. Também existe a opção de cobrança de **erros ortográficos**, que influencia na nota e também na observação realizada pelo aplicativo (Ver figura 6).

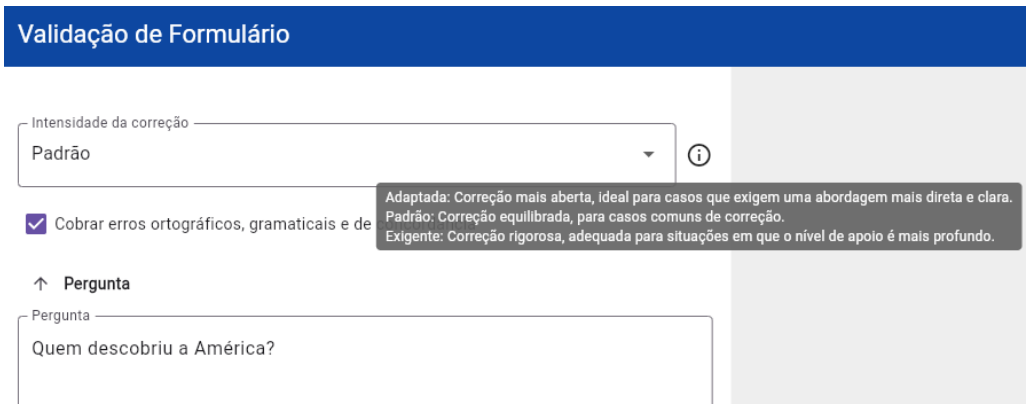


Figura 6. Filtros personalizados de correção

Tais métricas foram adicionadas a pedido do professor, onde diversos casos pedem diferentes formas de correção, tais como acesso ao material, prova em dupla, provas de português (alta cobrança de erros ortográficos, gramaticais e de concordância). O sistema também visa funcionar com avaliações de alto nível (por exemplo, Ensino Médio e Ensino Superior), o que torna a necessidade de uma opção para fazer a correção de forma mais rigorosa.

3.4. Engenharia de prompt

A engenharia de prompt desempenha um papel crucial no desempenho e na precisão da correção automática realizada pelo sistema. O treinamento da IA foi realizado com um conjunto de dados de respostas de alunos e respostas modelo dos professores. A primeira etapa do treinamento consistiu em fornecer ao sistema uma amostra representativa de dados, com uma variação significativa nas respostas dos alunos, para ajustar o prompt que é enviado ao modelo, de modo que o retorno se aproxime da avaliação do professor.

Durante os testes, os prompts foram ajustados em várias versões intermediárias até a versão final, com o objetivo de refinar a capacidade da IA de interpretar e corrigir as respostas. Isso incluiu o desenvolvimento de um prompt específico para todos os diferentes tipos de perguntas e respostas, garantindo uma avaliação generalizada. Embora o prompt gerado geralmente siga um padrão, ele se altera conforme os filtros personalizados selecionados já vistos na figura 6.

A intensidade da correção ortográfica e intensidade da correção se encontram em variáveis distintas calculadas em tempo de execução (Figura 7 e Figura 8).

```
intensidade_correcao = {
    "Fraca":
        "É muito importante citar que a resposta modelo do professor mostra TODOS os pontos que PODEM ser citados, e não que DEVEM ser citados.\n"
        "Você está corrigindo uma prova definida como 'Adaptada; Correção mais aberta, ideal para casos que exigem uma abordagem mais direta e clara', "
        "ou seja, as respostas podem faltar com diversos dados e o importante é considerar O CONHECIMENTO DO ALUNO EM RELAÇÃO AO CONTEÚDO, e não a PROXIMIDADE "
        "da resposta dele em relação à resposta modelo."
        "Baseado na análise, atribua uma nota de 0 a 4 para a resposta do aluno em relação à resposta modelo.\n",
    "Média":
        "É muito importante citar que a resposta modelo do professor mostra TODOS os pontos que PODEM ser citados, e não que DEVEM ser citados.\n"
        "Você está corrigindo uma prova definida como 'Padrão: Correção equilibrada, para casos comuns de correção.', "
        "ou seja, as respostas podem faltar com diversos dados e o importante é considerar O CONHECIMENTO DO ALUNO EM RELAÇÃO AO CONTEÚDO, e não a PROXIMIDADE "
        "da resposta dele em relação à resposta modelo."
        "Baseado na análise, atribua uma nota de 0 a 4 para a resposta do aluno em relação à resposta modelo.\n",
    "Forte":
        "É muito importante citar que a resposta modelo do professor mostra TODOS os pontos que PODEM ser citados, e não que DEVEM ser citados.\n"
        "Você está corrigindo uma prova definida como 'Exigente: Correção rigorosa, adequada para situações em que o nível de apoio é mais profundo.', "
        "ou seja, as respostas podem faltar com alguns dados e o importante é considerar O CONHECIMENTO DO ALUNO EM RELAÇÃO AO CONTEÚDO, e não a PROXIMIDADE "
        "da resposta dele em relação à resposta modelo."
        "Baseado na análise, atribua uma nota de 0 a 4 para a resposta do aluno em relação à resposta modelo.\n",
}.get(data.intensidadeCorrecao, "Avalie normalmente.")  # Padrão
```

Figura 7. Intensidade correção geral

A intensidade geral é levada em consideração como variável ”intensidade_correcao”, a qual influencia diretamente na intensidade da avaliação dos erros ortográficos (caso habilitada), retratada como ”correcao_erros_ortograficos”, na Figura 8.

```
intensidade_ortografia = {
    "Fraca":
        " devem ser avaliados de forma mais FLEXÍVEL e menos rigorosa",
    "Média":
        " devem ser avaliados de forma EQUILIBRADA e criteriosa",
    "Forte":
        " devem ser avaliados de forma RIGOROSA e detalhada",
}.get(data.intensidadeCorrecao, " devem ser avaliados")  # Padrão

correcao_erros_ortograficos = (
    f"Erros ortogrxáficos, gramaticais e de concordância {intensidade_ortografia}."
    if data.errosOrtograficos
    else "IGNORE erros ortográficos, gramaticais e de concordância na avaliação."
)
```

Figura 8. Intensidade correção ortografia

A definição das variáveis culmina no prompt final, visto na Figura 9.

```
prompt = (
    f"Pergunta (questão da prova): {data.pergunta}\n"
    f"Resposta modelo (resposta completa definida pelo professor): {data.respostaModelo}\n"
    f"Resposta do Aluno: {data.respostaAluno}\n"
    "Seu trabalho é avaliar a resposta do aluno com base na resposta modelo fornecida."
    f"{correcao_erros_ortograficos}\n"
    f"{intensidade_correcao}\n"
    f"Se a nota dada for entre 0 e 3, pode fazer comentários como: "
    "'poderia ter citado X' ou 'seria interessante explorar mais sobre Y', etc\n"
    "Responda SEMPRE apenas com a nota dada e uma observação do porque da nota, com no máximo 100 caracteres.\n"
    "Separador de sentenças = #/#\n"
    "Formato: nota#/#observacao\n"
    "Não utilize mais nenhum caractere especial além dos usados no separador de sentenças"
)
```

Figura 9. Prompt final

A aplicação foi desenvolvida para corrigir uma única resposta por interação, retornando o resultado e a observação para a mostragem na interface do professor. Isso não se torna um problema, pois todos os componentes do sistema — incluindo a interface do usuário, o servidor e os demais módulos — são isolados de forma que cada parte possa ser testada independentemente. Esse isolamento permite que cada componente seja validado de maneira independente, sem dependências entre eles. O ambiente modularizado facilita a realização de testes unitários e de integração, garantindo que alterações em um módulo não afetem diretamente os outros. Portanto, o processo de testes é mais controlado e seguro, permitindo verificar o comportamento correto de cada unidade sem interferência de outros componentes, o que torna a validação e a manutenção do sistema mais eficientes e seguras.

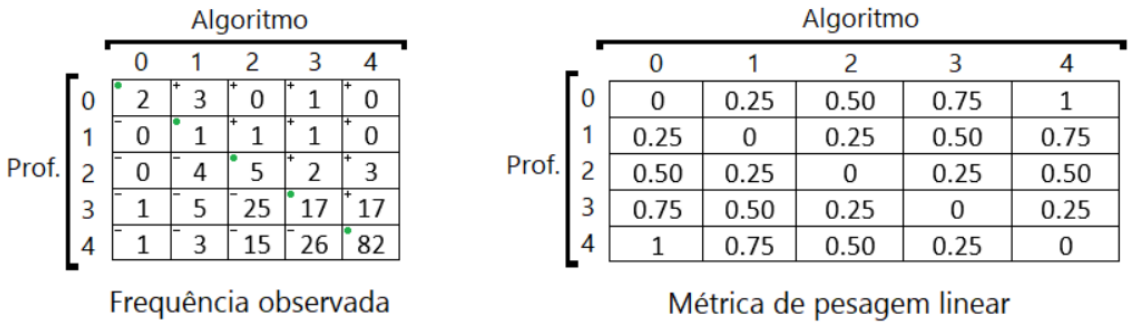
4. Resultados

A aplicação foi testada com duas provas do 7o ano do Ensino Fundamental II de uma escola pública. Foram coletados os dados dos alunos que aceitaram participar. No total foram recolhidas e mapeadas 215 respostas de alunos, 16 questões de prova e 16 respostas modelo com o professor.

As provas fornecidas precisaram ser digitalizadas para poderem ser utilizadas pelo sistema. Foi buscado manter o máximo da essência da resposta do aluno, mantendo a concordância e a ortografia da forma que se encontram nas provas. Para treinamento inicial foi utilizado cerca de 20% dos dados dos alunos de forma a aproximar ao máximo a nota dada pelo professor do resultado dado pela aplicação.

Após regulagem do sistema com pequena parte dos dados, foram realizados testes gerais do aplicativo com toda a base de dados coletada. As respostas das questões avaliadas têm um padrão de comportamento, que vai de 0 a 4, onde nota 4 significa que a questão está 100% correta, nota 3 = 3/4 da questão correta, 2 = 2/4, 1 = 1/4 e 0 = questão avaliada como errada (nenhum décimo de ponto considerado). Assim, foram metrificados alguns dados:

- Nota dada pela IA igual à nota dada pelo professor: 107 (49.76%);
- Nota dada tem um ponto de diferença (pra mais ou pra menos): 78 (36.27%);
- Nota dada tem dois pontos de diferença (pra mais ou pra menos): 24 (11.16%);
- Nota dada tem três pontos de diferença (pra mais ou pra menos): 5 (2.32%);
- Nota dada tem quatro pontos de diferença (pra mais ou pra menos): 1 (0.46%).



Prof. \ Algoritmo	0	1	2	3	4
0	2	3	0	1	0
1	0	1	1	1	0
2	0	4	5	2	3
3	1	5	25	17	17
4	1	3	15	26	82

Frequência observada

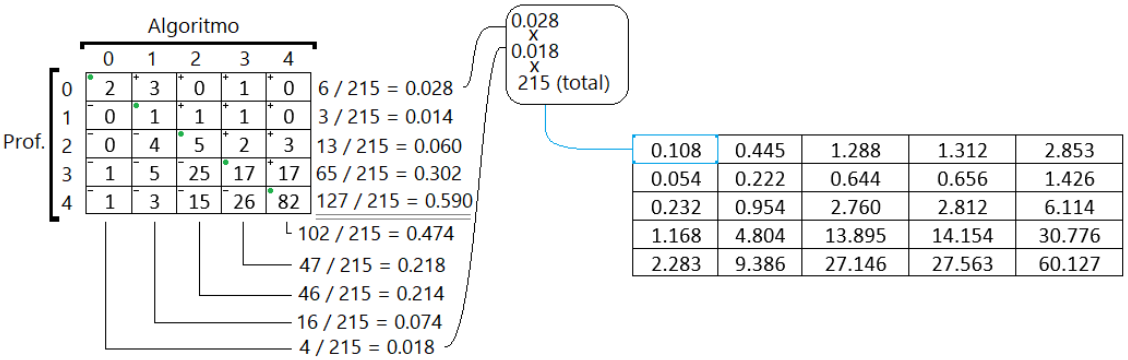
Prof. \ Algoritmo	0	1	2	3	4
0	0	0.25	0.50	0.75	1
1	0.25	0	0.25	0.50	0.75
2	0.50	0.25	0	0.25	0.50
3	0.75	0.50	0.25	0	0.25
4	1	0.75	0.50	0.25	0

Métrica de pesagem linear

Figura 10. Frequência observada + Métrica de pesagem linear

A métrica de Kappa Ponderado foi utilizada para avaliação. O mesmo avalia apenas a concordância entre dois ou mais avaliadores.

A frequência observada (Figura 10) se baseia em mapear como as distâncias das respostas dos dois avaliadores está distribuída. Já a Métrica de pesagem linear mostra o peso dessa distância, onde pode ser observado que, quanto mais próximas estiverem as respostas menor é o peso somado no cálculo final do Kappa, indo de 0,0 (ambos avaliadores deram a mesma resposta) até 1 (respostas totalmente opostas). Ambas as tabelas são utilizadas para calcular a tabela de frequência esperada (Figura 11), que será utilizada na fórmula final do Kappa ponderado.



0.108	0.445	1.288	1.312	2.853
0.054	0.222	0.644	0.656	1.426
0.232	0.954	2.760	2.812	6.114
1.168	4.804	13.895	14.154	30.776
2.283	9.386	27.146	27.563	60.127

Figura 11. Fórmula para tabela frequências esperadas

A fórmula final baseia-se no somatório da Métrica de pesagem pela Frequência observada, dividida pelo somatório da Métrica de pesagem pela Frequência esperada.

$$\kappa_w = 1 - \frac{\sum w_{ij} \cdot p_{ij}}{\sum w_{ij} \cdot e_{ij}} = 1 - \frac{36.25}{54.509} = 0.665$$

Figura 12. Resultado Kappa ponderado

O resultado do coeficiente Kappa é interpretado de forma com que, caso seja igual a 0,00, representa concordância “ruim”, de 0,00 até 0,20 como concordância “leve”, de 0,21 até 0,40 como concordância “razoável”, de 0,41 até 0,60 como concordância “mo-

derada", de 0,61 até 0,80 como concordância "substancial" e de 0,81 até 0,99 como concordância "quase perfeita". Um coeficiente kappa de 1 representa concordância perfeita.

O resultado calculado para o primeiro teste definitivo retornou um coeficiente de 0.665, o qual entra na categoria de concordância "substancial"entre os dois examinadores (neste caso, o professor e a IA). Em contrapartida, o coeficiente Kappa indica que os testes e regulagens realizados tornaram a forma de corrigir da IA aceitável apenas para as provas do professor em questão, não sendo possível avaliar o sistema sendo utilizado em outros contextos.

O resultado calculado para o primeiro teste definitivo indicou um coeficiente de 0,665, o que caracteriza uma concordância "substancial"entre os dois examinadores (neste caso, o professor e a IA). No entanto, o coeficiente Kappa sugere que os testes e ajustes realizados tornaram a forma de correção da IA aceitável apenas para as provas do professor em questão, não permitindo a avaliação do sistema em outros contextos. O trabalho de [Silva 2023], que utilizou respostas de 0 a 3, obteve um coeficiente médio de 0,674, evidenciando uma concordância muito semelhante à do trabalho atual.

5. Conclusão

Conclui-se, através da análise dos resultados obtidos nos testes, que a aplicação desenvolvida demonstrou ser capaz de se aproximar do método de correção utilizado pelo professor que aplicou e corrigiu as provas inicialmente. Isso evidencia o potencial do sistema para auxiliar no processo de avaliação de respostas discursivas, especialmente ao reduzir o tempo gasto na correção e proporcionar maior consistência nas avaliações.

No entanto, é importante compreender que o resultado obtido através do modelo de IA é livre de qualquer subjetividade. Portanto, as métricas analisadas não validam se a aplicação é eficaz em corrigir respostas, já que, muitas vezes, a subjetividade do professor é o que determina a nota atribuída à questão.

Assim, é importante entender que os testes e resultados obtidos nesta pesquisa visam prever se a aplicação é capaz de corrigir questões se aproximando ao máximo do raciocínio do avaliador em questão.

Portanto, alguns aspectos precisam ser abordados em trabalhos futuros para ampliar a viabilidade e a aplicabilidade da solução. Primeiramente, é essencial testar a aplicação com uma base de dados mais ampla e diversificada, incluindo provas de séries mais avançadas, como do Ensino Médio e do Ensino Superior, isso permitirá avaliar a capacidade do sistema de lidar com diferentes níveis de complexidade nas respostas e diferentes estilos de correção por parte dos professores.

Outro ponto importante é a implementação de métodos mais eficientes para o ingresso de dados na aplicação. Atualmente, o processo de digitação manual das provas, respostas modelo e respostas dos alunos toma muito tempo, tornando o sistema inviável para aplicações em larga escala, visto que seu principal objetivo é justamente economizar tempo.

Desta forma, as próximas etapas de desenvolvimento e trabalhos futuros devem focar em testar o sistema com bases de dados mais amplas e variadas, abrangendo diferentes disciplinas, níveis educacionais e estilos de correção. Expandir a análise do sistema em contextos mais desafiadores, como avaliações em disciplinas técnicas e científicas,

que exigem maior precisão na correção. Automatizar a realização da digitalização de provas manuscritas para diminuir consideravelmente o tempo necessário de validação de avaliações em larga escala. Implementação de opção para importar arquivos de texto/csv/excel para facilitar a experiência do usuário. Melhorar a visualização das observações retornadas pela IA, podendo destacar trechos da resposta e adicionar comentários com sugestões para trechos que podem ser melhorados.

Esses avanços serão fundamentais para consolidar a solução como uma ferramenta prática, eficiente e amplamente utilizável no ambiente educacional.

Referências

Amorim, T. H. d. et al. (2023). Desenvolvimento de um sistema web para controle de pedidos de balões personalizados de uma empresa de sombrio, santa catarina.

Appareddy, R., Kedhari, R., Edla, K., et al. (2023). Phishing url detection using machine learning with knn algorithm and fast api. In *2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, pages 1–4. IEEE.

Bollela, V. R., Borges, M. d. C., and Troncon, L. E. d. A. (2018). Avaliação Somativa de Habilidades Cognitivas: Experiência Envolvendo Boas Práticas para a Elaboração de Testes de Múltipla Escolha e a Composição de Exames. *Revista Brasileira de Educação Médica*, 42:74 – 85.

Chamorro-Premuzic, T. (2006). Creativity versus conscientiousness: Which is a better predictor of student performance? *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 20(4):521–531.

Galhardi, L., Barbosa, C. R., de Souza, R. C. T., and Brancher, J. D. (2018). Portuguese automatic short answer grading. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 29, page 1373.

Gonzalez, M. and Lima, V. L. S. (2003). Recuperação de informação e processamento da linguagem natural. In *XXIII Congresso da Sociedade Brasileira de Computação*, volume 3, pages 347–395. sn.

Iosifova, O., Iosifov, I., Rolik, O., and Sokolov, V. (2020). Techniques comparison for natural language processing. *MoMLeT&DS*, 2631(I):57–67.

Kalla, D., Smith, N., Samaah, F., and Kuraku, S. (2023). Study and analysis of chat gpt and its impact on different fields of study. *International journal of innovative science and research technology*, 8(3).

Ludermir, T. B. (2021). Inteligência artificial e aprendizado de máquina: estado atual e tendências. *Estudos Avançados*, 35:85–94.

Mridha, M. F., Keya, A. J., Hamid, M. A., Monowar, M. M., and Rahman, M. S. (2021). A comprehensive review on fake news detection with deep learning. *IEEE access*, 9:156151–156170.

Neto, J. R., Falcao, T. P., Oliveira, V., Souza, S., Fiorentino, G., Galdino, J. V., Alves, G., and Mello, R. F. (2022). Tutoria: Plataforma para suporte à correção de atividades e envio de feedback personalizado. In *Anais do I Workshop de Aplicações Práticas de Learning Analytics em Instituições de Ensino no Brasil*, pages 21–29. SBC.

Oliveira, D., Pozzebon, E., and Santos, T. d. (2020). Aplicação das técnicas de processamento de linguagem natural cosine similarity e word movers distance para auxiliar na correção de questões discursivas em um tutor inteligente. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1243–1252. SBC.

Pepple, D. J., Young, L. E., and Carroll, R. G. (2010). A comparison of student performance in multiple-choice and long essay questions in the mbbs stage i physiology examination at the university of the west indies (mona campus). *Advances in physiology education*, 34(2):86–89.

Pimentel, C. H. M. (2023). Treinando um modelo de tradução automática baseado em transformers.

Porto, D. (2023). Enem 2023: entenda como a nota do exame é calculada. *CNN. Questões e redação são avaliadas utilizando métodos distintos*, page 1.

Rampazzo, S. R. d. R. (2010). Instrumentos de Avaliação: Reflexões e Possibilidades de Uso no processo de Ensino e Aprendizagem. *Professor PDE e os desafios da Escola Pública Paranaense*, 2:05 – 12.

Silva, A. C. (2023). Correção automática de questões discursivas de resposta curta: uma abordagem baseada em extração de informação.

Singh, A., Karayev, S., Gutowski, K., and Abbeel, P. (2017). Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. In *Proceedings of the fourth (2017) acm conference on learning@ scale*, pages 81–88.

Tashildar, A., Shah, N., Gala, R., Giri, T., and Chavhan, P. (2020). Application development using flutter. *International Research Journal of Modernization in Engineering Technology and Science*, 2(8):1262–1266.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zulunov, R. and Soliev, B. (2023). Importance of python language in development of artificial intelligence. -, 1(1):7–12.