FERRAMENTA DE APOIO A RESPOSTAS DE E-MAILS UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL

Trabalho de Conclusão de Curso

Engenharia da Computação

Felipe Pernambuco Maciel

Orientador: Prof. Dr. Bruno José Torres Fernandes

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

FELIPE PERNAMBUCO MACIEL

FERRAMENTA DE APOIO A RESPOSTAS DE E-MAILS UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Orientador: Prof. Dr. Bruno José Torres Fernandes

Recife, Outubro de 2022.

Maciel, Felipe Pernambuco

Ferramenta de apoio a respostas de e-mails utilizando Processamento de Linguagem Natural / Felipe Pernambuco Maciel. – Recife - PE, 2022.

xv, 39 f. : il. ; 29 cm.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) Universidade de Pernambuco, Escola Politécnica de Pernambuco, Recife, 2022.

Orientador: Profª. Dr. Bruno José Torres Fernandes.

Inclui referências.

1. Classificação textual. 2. Similaridade entre textos. 3. Processamento de Linguagem Natural. I. Título. II. Fernandes, Bruno José Torres. III. Universidade de Pernambuco.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 19/10/2022, às 10h00min, reuniu-se para deliberar sobre a defesa da monografia de conclusão de curso do(a) discente **FELIPE PERNAMBUCO MACIEL**, orientado(a) pelo(a) professor(a) **BRUNO JOSÉ TORRES FERNANDES**, sob título FERRAMENTA DE APOIO A RESPOSTAS DE E-MAILS UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL, a banca composta pelos professores:

ALEXANDRE MAGNO ANDRADE MACIEL (PRESIDENTE)
BRUNO JOSÉ TORRES FERNANDES (ORIENTADOR)

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

☒ Aprovada ☐ Aprovada com Restrições* ☐ Reprovada

e foi-lhe atribuída nota: 9,0 (nove)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O(A) discente terá oito dias para entrega da versão final da monografia a contar da data deste documento.

AVALIADOR 1: Prof (a) **ALEXANDRE MAGNO ANDRADE MACIEL**

AVALIADOR 2: Prof (a) **BRUNO JOSÉ TORRES FERNANDES**

AVALIADOR 3: Prof (a)

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

ESCOLA POLITÉCNICA
DE PERNAMBUCO

Autorização de publicação de PFC

Eu, **Felipe Pernambuco Maciel** autor(a) do projeto de final de curso intitulado: **FERRAMENTA DE APOIO A RESPOSTAS DE E-MAILS UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL;** autorizo a publicação de seu conteúdo na internet nos portais da Escola Politécnica de Pernambuco e Universidade de Pernambuco.

O conteúdo do projeto de final de curso é de responsabilidade do autor.

Felipe P. Maciel
Felipe Pernambuco Maciel

Orientador(a): **Bruno José Torres Fernandes**

Coorientador(a):

Prof, de TCC: **Daniel Augusto Ribeiro Chaves**

Data: 19/10/2022

Resumo

Nas últimas décadas o e-mail assumiu um importante papel na comunicação envolvendo pessoas e instituições. No cenário do setor administrativo e de secretariado de uma universidade isso não é diferente. Gerir o relacionamento com discentes, docentes, potenciais ingressantes e egressos é uma atividade de grande importância para esses setores. Além disso, diminuir o tempo da janela de resposta, a fim de manter a comunicação fluida e permitir celeridade no andamento dos processos é sempre uma meta para aumentar a satisfação dos usuários envolvidos na comunicação e dar vazão ao fluxo das solicitações. Diante deste cenário, o projeto apresentado neste trabalho descreve a construção e introduz uma ferramenta construída na linguagem de programação Python capaz de indicar uma sugestão de resposta a um e-mail recebido após comparar seu conteúdo de texto e obter a similaridade em relação a exemplos de uma base de dados previamente construída utilizando técnicas de Processamento de Linguagem Natural. O projeto se provou capaz de apoiar um usuário na tarefa de resposta de e-mails com temática frequente. Medidas com possibilidade de aplicação imediata como melhoria das instâncias de exemplo utilizados na planilha de apoio, bem como trabalhos futuros de melhoramento da análise de similaridade textual, tem a possibilidade de aumentar a precisão da ferramenta.

Palavras-chave: E-mail. PLN. Similaridade. Python. TF-IDF.

Abstract

In the last decades, e-mail has assumed an important role in communication involving people and institutions. In the scenario of the administrative and secretarial sector of a university, this is no different. Managing the relationship with students, professors, potential newcomers and graduates is an activity of great importance for these sectors and reducing the time of the response window as much as possible in order to maintain fluid communication and allow speed in the progress of processes is always a goal to increase the satisfaction of the users involved in the communication and to give flow to the flow of requests. Given this scenario, the project presented in this work describes the construction and introduces a tool built in the Python programming language capable of indicating a suggested response to an received email after comparing its text content and obtaining similarity in relation to examples of a database previously built using Natural Language Processing techniques. The project proved capable of supporting an user in the task of responding to frequently themed emails. Measures with the possibility of immediate application, such as improvement of the example instances used in the support spreadsheet, as well as future work to improve the analysis of textual similarity, have the possibility of increasing the accuracy of the tool.

Keywords: Email. NLP. Similarity. Python. TF-IDF.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e Siglas

BOW	*Bag of Words*
CWI	*Centrum Wiskunde & Informatica*
HTML	*HyperText Markup Language*
IEEE	*Institute of Electrical and Electronics Engineers*
IMAP	*Internet Message Access Protocol*
MIME	*Multipurpose Internet Mail Extensions*
PLN	Processamento de Linguagem Natural
PPGEC	Programa de Pós-graduação em Engenharia da Computação
SMTP	*Simple Mail Transport Protocol*
TF-IDF	*Term Frequency - Inverse Document Frequency*
UPE	Universidade de Pernambuco
VSM	*Vector Space Model*
XML	*Extensible Markup Language*

Sumário

1. Introdução

Desde sua origem em 1971 e após a sua popularização junto à expansão da Internet a partir da década de 1990, o e-mail assumiu um importante papel na comunicação envolvendo pessoas e instituições. No âmbito das universidades isso não é diferente. Gerir o relacionamento com discentes, docentes, potenciais ingressantes e egressos em uma universidade é uma atividade de grande importância para o setor administrativo e de secretariado. Neste cenário, o e-mail assumiu um importante papel na comunicação entre os envolvidos pelas características principais que possui que é de ser uma comunicação assíncrona, gerar registros das interações realizadas, permitir a fácil visualização do histórico de uma troca de mensagens, permitir a incorporação de anexos, hyperlinks e fotos [1]. Mesmo com a ascensão das redes sociais nas duas últimas décadas proporcionando outras formas de comunicação via Internet, as estatísticas globais do uso de e-mails seguem em crescimento. Segundo pesquisa do *The Radicati Group*, uma empresa de pesquisa de mercado de tecnologia, o número de usuários de e-mail chegou a 4,147 bilhões em 2021 e cresce a uma taxa anual de 3% novos usuários com um tráfego diário que foi de 319,6 bilhões de e-mails e tem projeção estimada de 333,2 bilhões para o ano de 2022, o que indica um crescimento de 4,3% [2].

Esse papel e importância assumidos pelo e-mail ter se tornado responsável por uma grande fatia nas comunicações traz ambos desafios e oportunidades. Por um lado, para o caso da universidade, é gerado um alto volume de contatos e solicitações por essa via, tendo picos em certas épocas do ano, como os períodos que antecedem a matrícula. Em paralelo a isso, apesar do e-mail ser uma forma de comunicação assíncrona, é necessário diminuir ao máximo o tempo da janela de resposta a fim de manter a comunicação fluida e ágil, permitir celeridade no andamento dos processos, aumentar a satisfação dos usuários envolvidos na comunicação e dar vazão ao fluxo das solicitações. Por um outro lado nota-se também oportunidades. Muitos desses contatos envolvem temas que se repetem e que muitas vezes possuem, também, respostas similares. Em alguns locais de trabalho, a utilização do e-mail para comunicação se tornou popular a ponto de

empregadores limitarem o tempo dedicado à leitura e resposta de e-mails de forma a permitir os empregados a realizarem outras tarefas necessárias [3].

A sobrecarga de funcionários com tarefas com características de serem repetitivas e com necessidade de reconhecimento de padrões é um problema que pode ser enxergado sob a ótica de oportunidade, visto que essas características se sobrepõem às soluções que o campo da Inteligência Artificial é capaz de gerar [4].

Google e Microsoft, empresas que possuem dois dos mais populares provedores de e-mail atualmente, notaram nas características do problema uma oportunidade de melhoria com foco na agilidade de respostas e criaram os recursos de Resposta Inteligente e Respostas Sugeridas, respectivamente. Esses recursos têm como alvo e-mails que podem ser respondidos com uma resposta curta. Quando um e-mail com esse perfil chega na caixa de entrada do usuário, um algoritmo que faz uso de tecnologias de Processamento de Linguagem Natural (PLN) e aprendizado de máquina sugere algumas respostas curtas que podem ser facilmente acionadas para respondê-lo. Em uma outra vertente do mercado, a Amazon oferece o *Amazon Comprehend*, serviço de PLN que também utiliza aprendizado de máquina para extrair significado e percepções em texto. O serviço, que é capaz de identificar características como a linguagem do texto, extrair palavras e frases chave, locais, tema e uma série de outros tópicos, pode ser combinado com outros serviços oferecidos pela Amazon como o *Amazon Simple Email Service* para, por exemplo, construir soluções poderosas de respostas a clientes dentro de um fluxo de suporte ao cliente [5].

Tabela 1 - Comparação entre soluções existentes

Solução	Formato	Custo	Customização	Público Alvo	Foco da Solução
Resposta Inteligente	Embutido no Gmail	Gratuito (incluso no Gmail)	Nenhuma	Usuário doméstico e corporativo	E-mails que possam ser respondidos com uma resposta curta
Respostas Sugeridas	Embutido no Outlook	Gratuito (incluso no Outlook)	Nenhuma	Usuário doméstico e corporativo	E-mails que possam ser respondidos com uma resposta curta
Amazon Comprehend	API	Pago, cobrança por unidades de 100 caracteres	Alta, com possibilidade de integração com outras soluções	Empresas	Soluções complexas de atendimento ao cliente e análise textual

Fonte: Próprio autor

Exemplos como esses mostram que agregar à rotina dos usuários ferramentas de apoio que tenham capacidade de reduzir o tempo de resposta tem potencial de ajudar a manter a fluidez na comunicação e com isso proporcionar uma maior agilidade no andamento dos processos e aumentar a satisfação dos usuários envolvidos.

A possibilidade de utilizar o conteúdo textual dos e-mails recebidos de forma a desenvolver uma solução que facilitasse a rotina dos usuários, fosse mais customizável e adaptável que as soluções embutidas no Gmail e Outlook, e que pudesse ser utilizada de forma gratuita, ao contrário da solução da Amazon, foi o que motivou este projeto, cujo objetivo é desenvolver uma ferramenta de apoio a respostas de e-mail usando como caso particular e contexto do mundo real a Secretaria do Programa de Pós-graduação em Engenharia da Computação (PPGEC) para estudo de caso. Essa ferramenta atuará fazendo a análise do conteúdo das mensagens utilizando conceitos de Processamento de Linguagem Natural, uma vertente da Inteligência Artificial que permite máquinas a analisar, interpretar e processar linguagem humana, e a linguagem de programação Python, que possui alta capacidade de automação de tarefas [6].

1.1. Objetivos

Este trabalho tem como objetivo principal desenvolver uma ferramenta de apoio a resposta de e-mails utilizando como estudo de caso a secretaria do Programa de Pós-graduação em Engenharia da Computação (PPGEC) da Universidade de Pernambuco (UPE) utilizando a linguagem de programação Python e fazendo uso de ferramentas e conceitos de Processamento de Linguagem Natural (PLN) para propor sugestões de respostas após análise de similaridade do texto presente no corpo do e-mail de entrada com propostas de respostas pré-redigidas pelo usuário da solução. Diante do objetivo principal proposto, metas com maior especificidade também foram definidas. São elas:

- Diminuir a janela de tempo entre recebimento e resposta de e-mails;
- Diminuir o tempo total diário gasto com resposta de e-mails;
- Reduzir a quantidade de respostas manualmente digitadas.

1.2. Estrutura do Documento

Este trabalho está dividido em 4 capítulos, estando incluído nesta contagem o capítulo atual que promove uma introdução ao tema e objetivos do projeto. Em seguida, o Capítulo 2 traz um estudo a respeito do Processamento de Linguagem Natural e similaridade entre textos. O Capítulo 3 traz as etapas do desenvolvimento que resultou na ferramenta proposta. Já o Capítulo 4 encerra o trabalho com uma reflexão sobre o que foi desenvolvido e possibilidades de melhorias ao projeto

2. Fundamentação Teórica

Processamento de Linguagem Natural (PLN) é um subcampo dentro da ciência da computação e inteligência artificial que faz uso de conceitos de linguística e estatística para estudar o processamento da linguagem humana por computadores. PLN é uma área de estudo motivada por teorias que utiliza uma variação de técnicas para a análise e representação automática da linguagem humana [7], além de ser uma área de pesquisa que envolve diversas disciplinas do conhecimento humano e por tratar de linguagem, por si só, ganha complexidade extra devido a profundidade natural do assunto e característica de ser um tema fluido, já que a linguagem sofre modificações ao longo do tempo [8].

O PLN busca proporcionar maior integração entre máquinas e humanos ao possibilitar que computadores entendam, interpretem e manipulem a linguagem humana. O seu estudo possibilita a humanidade de estar mais próxima do conceito de Computação Ubíqua [9], segundo o qual a computação estará presente no dia a dia dos seres humanos de forma integrada, imersiva e onipresente.

Com foco em análise de similaridade, este capítulo busca apresentar alguns conceitos fundamentais a respeito do processo de extração de características entre textos para posterior comparação, bem como a métrica utilizada para estabelecer a similaridade entre eles.

2.1. Conceitos de Similaridade Entre Textos

Similaridade entre textos é a base das tarefas de Processamento de Linguagem Natural [10]. Palavras podem ser similares de forma léxica e de forma semântica. Elas são similares lexicalmente se tiverem uma sequência de caracteres similar. Já a similaridade semântica é baseada no significado, contexto e forma como são utilizadas.

A similaridade léxica é obtida por meio de algoritmos baseados em cadeias de caracteres [11] ou, termo mais popularmente conhecido na área de desenvolvimento de software, *strings.* Já a similaridade semântica utiliza técnicas de análise do corpo

de documentos. O cálculo da similaridade textual pode ser dividido em: Distância Textual e Representação Textual. A Distância Textual utiliza métricas que conseguem determinar a distância entre representações numéricas das características de um texto. Essa representação, por sua vez, possui diferentes formas de ser obtida. Já a Representação Textual utiliza abordagens de análises baseadas em *strings*, corpo de documentos e utilização de grafos para estabelecer similaridades semânticas [11].

2.2. Extração de Características e Vetorização

Um importante passo dentro do processo que obtém a similaridade entre textos é a representação numérica das características de um texto. Esse tipo de representação é chamada de *Vector Space Model* (VSM) [12] e o resultado dessa representação será um vetor multidimensional a partir do qual será possível escolher e aplicar uma métrica apropriada para medir a similaridade entre dois termos por meio de alguma distância, definida pela métrica, entre eles.

Essa etapa utiliza técnicas estatísticas de contagem e cálculo para extrair informações do corpo de um documento num processo cuja saída da extração será um vetor, que quando combinado, forma uma matriz que representa um documento. Para essa parte, algumas são as abordagens possíveis para fazer a análise:

- ***Bag-of-Words*** (BOW): é a forma de representar um documento como uma combinação de palavras, sem considerar a ordem de aparição delas no documento [10]. O método de *CountVectorization* é a essência do modelo BOW, no qual é realizada a contagem da ocorrência de palavras em um documento.
- ***Term Frequency - Inverse Document Frequency*** (TF-IDF): conta a quantidade de aparições de uma palavra num documento para identificar palavras que aparecem com mais frequência [10]. Palavras que aparecem com mais frequência num documento possuem uma menor importância para o significado semântico do documento.

No BOW puro, as palavras presentes em um texto possuem pesos iguais, sendo levado em consideração apenas a contagem de ocorrências de cada uma no

texto. Por outro lado, para o TF-IDF, utilizando como exemplo a temática de redação de textos de e-mails, a ocorrência de palavras comumente utilizadas como saudações: “bom”, “dia”, “boa”, “tarde”, “noite”, “olá”, introduções como: “eu”, “gostaria”, “de”, “saber”, “quando”, “qual” ou finalizações como: “obrigado”, “grato”, “desde”, “já”, “atenciosamente”, tendem a ganhar uma menor importância no método pois o assunto e conteúdo de um e-mail não se torna mais identificável ou diferenciável pela presença delas.

O valor da frequência de cada termo do documento é obtido como em

$$tf_{t,d} = log_{10}(count(t,d) + 1)\,,$$

onde $tf_{t,d}$ é a frequência do termo t no documento d. Em seguida é obtida a relação da frequência inversa idf_t do documento utilizando a equação

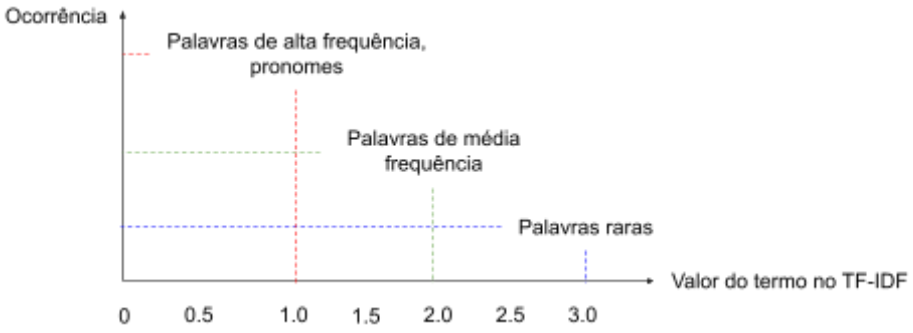
$$idf_t = log_{10}(\frac{N}{df_t}),$$

onde N é o número total de documentos e df_t é o número de documentos em que um termo aparece, onde termos que aparecem menos ganham maior valor e diferenciam documentos [13]. Após isso é calculado o TF-IDF em um produto escalar do valor do TF com o IDF

$$w_{t,d} = tf_{t,d} \times idf_t\,,$$

onde um maior valor significa que o termo possui maior relevância para o documento. A relação pode ser vista na Figura 1.

Figura 1 - Frequência de Ocorrência de Palavra X Valor do Termo no TF-IDF



Fonte: Próprio autor

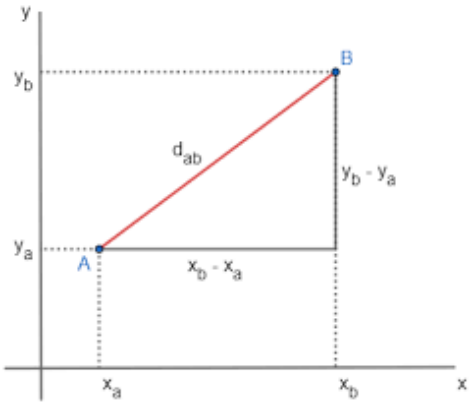
O resultado final da aplicação do TF-IDF para cada documento e termos presentes neles é uma matriz formada pelos vetores com pesos determinados pelo TF-IDF.

2.3. Métricas de Obtenção de Similaridade

Dentre as métricas para obter a similaridade entre termos, algumas surgem com maior destaque. Acompanhadas de uma breve explicação, são elas:

- **Distância Euclidiana:** é medido calculando a raiz quadrada da soma do quadrado diferenças entre os dois elementos vetoriais.

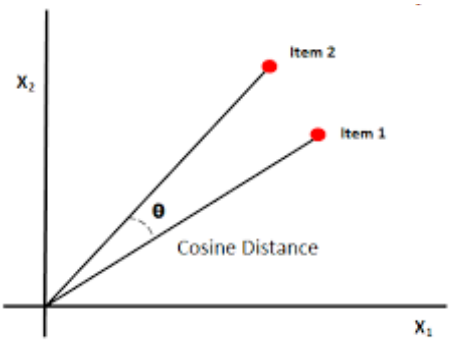
Figura 2 - Ilustração da Distância Euclidiana



Fonte: [14]

- **Similaridade por Cosseno:** é uma medida de semelhança entre dois vetores projetados num espaço multidimensional medida pelo cosseno do ângulo formado entre eles.

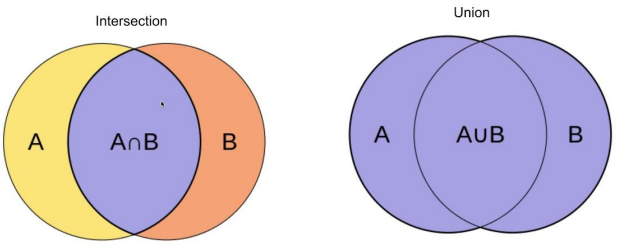
Figura 3 - Ilustração da Similaridade por Cosseno



Fonte: [15]

- **Similaridade de Jaccard:** utiliza teoria dos conjuntos para calcular o número de termos compartilhados dividido pelo número de todos os termos exclusivos presentes em ambas as *strings* dos textos comparados.

Figura 4 - Ilustração da Similaridade de Jaccard

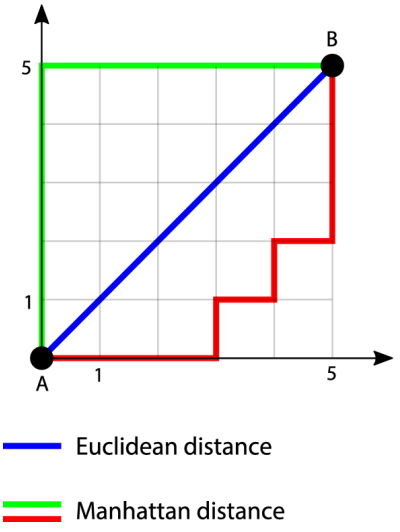


$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Fonte: [16]

- **Distância de Manhattan:** essa é uma métrica utilizada para computar a distância do caminho entre dois pontos, mas esses pontos devem estar dispostos numa malha do tipo grade. A Distância de Manhattan entre os pontos é a soma das diferenças entre suas componentes.

Figura 5 - Ilustração da Distância de Manhattan



Fonte: [17]

3. Desenvolvimento da Ferramenta

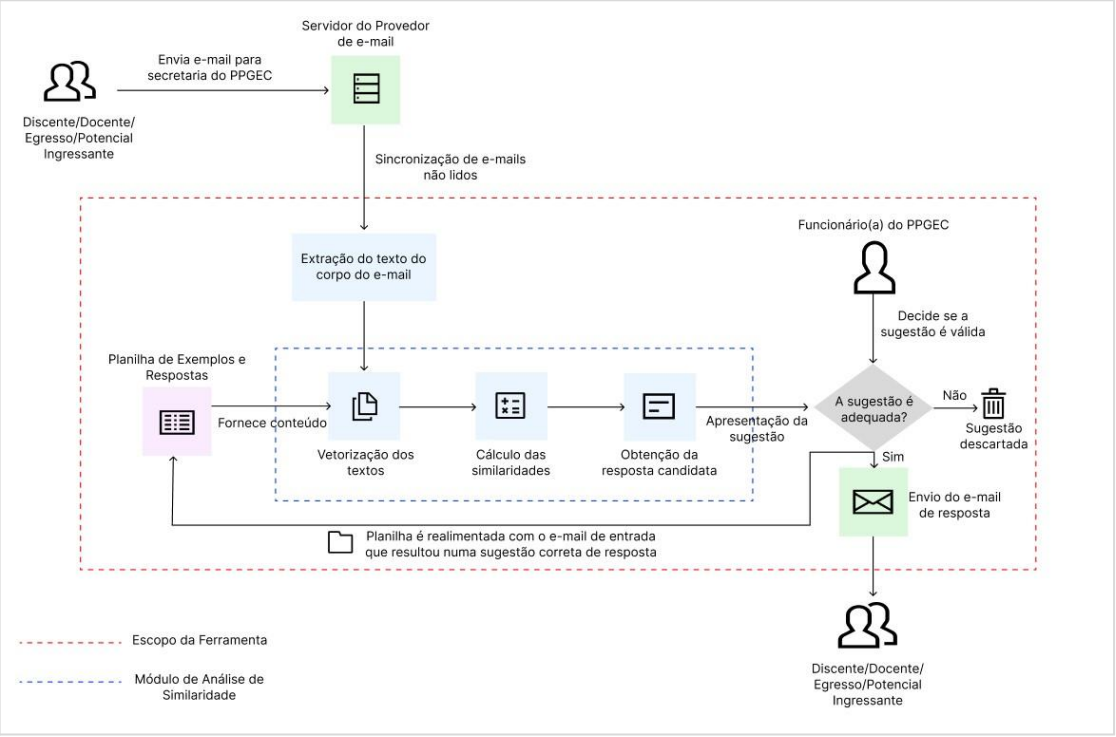
Este capítulo visa descrever com mais detalhes o processo de desenvolvimento da ferramenta de apoio a resposta de e-mails que utiliza Processamento de Linguagem Natural e técnicas de obtenção de similaridade entre textos para fazer a leitura do corpo de um e-mail da caixa de entrada e sugerir a melhor resposta para ele. Isso é feito ao comparar a similaridade do corpo desse e-mail de entrada a um exemplo presente numa planilha pré construída.

Serão descritas as principais ferramentas utilizadas bem como decisões importantes que foram tomadas no processo de desenvolvimento.

3.1. Fluxograma de Processos da Solução

Para melhor visualização de como funciona a ferramenta esta seção irá discorrer sobre o fluxo de execução da solução ilustrada pela Figura 6.

Figura 6 - Fluxograma de Processos da Solução



Fonte: Próprio autor

Primeiramente, a ferramenta possui dois modos de operação de forma a espelhar a rotina da secretaria do PPGEC. O modo Varredura, caracterizado por ser um modo de execução única, realiza a sincronização da caixa de entrada de e-mail e obtém todos os e-mails com sinalizador de 'não-lido'. O modo *Listener* é um modo de execução recorrente, ele realiza em loop a verificação da caixa de entrada de acordo com um período configurado. Por ser de execução única, o modo Varredura pode ser executado para verificar uma a caixa uma única vez, por exemplo, na segunda-feira no início do expediente após o fim de semana sem verificação dos e-mails. Já o modo *Listener* pode ser executado, por exemplo, ao longo de um dia de jornada de trabalho de forma a ficar analisando a caixa para a chegada novos e-mails em tempo real.

Após a escolha do modo, é feito o login via conexão IMAP com as credenciais do usuário e escolha do provedor, para então obter os e-mails não lidos. Cada e-mail segue para análise de conteúdo individualmente.

Na análise dos e-mails é feita uma sequência de passos para realizar a limpeza dos dados trazidos no pacote da mensagem para posterior extração do texto do corpo do e-mail, que pode vir em formato de texto simples ou em formato HTML. Após esse processo, extraído apenas o texto do corpo do e-mail, esse texto segue para o módulo de análise de similaridade.

No módulo de análise de similaridade é feita uma conexão com a planilha de exemplos, de onde são obtidos os textos de exemplo de entrada que serão comparados com o texto do corpo do e-mail extraído no passo anterior. Uma instância da classe *TfidfVectorizer* da biblioteca Scikit-learn fica a cargo da extração de características e vetorização dos textos através da técnica TF-IDF. Isso é feito para o novo texto de entrada e todos os outros da coluna de exemplos de entrada da planilha. Com as mensagens agora em forma de vetor, entra em cena a métrica de obtenção de similaridade escolhida, a Similaridade por Cosseno. O vetor do texto de entrada é comparado com todos os outros exemplos de entrada e é gerada uma lista de valores de similaridade. O texto correspondente ao índice da maior similaridade dessa lista na coluna de respostas é retornado, gerando assim a sugestão de resposta ao usuário.

Por fim, uma janela contendo essa sugestão de resposta e informações do destinatário surge, dando ao usuário da ferramenta a possibilidade de editar a sugestão, descartá-la ou enviá-la como resposta. Quando uma sugestão de resposta é aceita, o texto do e-mail que deu origem a essa sugestão é inserido na planilha como forma de realimentar o sistema. Alimentar a planilha com mais de uma instância de e-mails de entrada como exemplo pode ser capaz de aumentar a precisão, visto que aumentaria a possibilidade de textos que tratassem o mesmo tema, mas com estilos de escrita diferentes, se mostrarem mais similares com alguma das instâncias de exemplo desse tema presente na planilha.

3.2. Obtenção de Similaridade Entre Textos

Para extração de características e vetorização, a característica do TF-IDF de identificar em palavras que aparecem menos uma maior importância no que se refere ao sentido do texto foi a principal diferença que o fez ser escolhido como método para extração de características e vetorização na construção da ferramenta. É ele quem vai garantir que palavras comuns presentes em corpos de e-mails como saudações, introduções, despedidas e agradecimentos não se destaquem na obtenção de similaridade do novo texto de entrada em relação aos textos existentes na planilha, deixando a cargo de palavras menos comuns o maior peso na definição do sentido e significado do texto.

Já em relação às métricas de obtenção de similaridade apresentadas, para ser utilizada na ferramenta desenvolvida, foi escolhida a Similaridade por Cosseno como métrica de obtenção de distância entre os pontos que representam os textos após as suas vetorizações. A Distância de Manhattan foi descartada logo de início pois os vetores obtidos das características de textos não tendem a se comportar como forma de grade, principal uso dessa métrica. Inicialmente a Similaridade de Jaccard foi considerada, mas a sua característica de considerar apenas o conjunto de palavras únicas para cada texto a torna pouco propícia para a obtenção de similaridade textual pois descarta a quantidade de aparições de uma palavra num texto.

Entre a Distância Euclidiana e a Similaridade por Cosseno, a segunda foi a escolhida devido a uma característica importante: ela não é afetada pela magnitude dos vetores gerados pela extração de características dos textos. A métrica da Distância Euclidiana, por outro lado, é. Trazendo para similaridade entre textos isso significa que para a Distância Euclidiana dois textos podem ser considerados pouco similares apenas por terem tamanhos diferentes. Um exemplo que pode ser dado seria a ocorrência de uma mesma palavra 20 vezes em um texto e cinco vezes em outro. Semanticamente esses textos podem possuir significados semelhantes, mas sob a ótica da Distância Euclidiana eles poderiam ser considerados pouco similares por terem magnitudes diferentes. Já sob a ótica da Similaridade por Cosseno, quando os vetores desses textos fossem plotados num espaço multidimensional eles poderiam ter um maior grau de similaridade pois o que importaria seria o ângulo entre as projeções, que sendo menor, indicaria uma maior similaridade. Matematicamente, ela mede o cosseno do ângulo entre dois vetores projetados

$$similaridade = cos(\Theta) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

sob o ponto de vista de significado semântico dos textos isso é relevante pois reduz a importância do tamanho dos textos na obtenção de similaridade.

3.3. Python 3

A linguagem de programação Python, criada por Guido van Rossum em 1991 no Centrum Wiskunde & Informatica (CWI) - Instituto de Pesquisa Nacional para Matemática e Ciência da Computação - localizado na Holanda, é a linguagem de programação mais popular da atualidade segundo o último ranking do Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) divulgado em 23 de agosto de 2022 [18].

Python é uma linguagem de programação orientada a objetos, interpretada, dinamicamente tipada, com suporte a construção de interfaces gráficas, curva de aprendizado relativamente pouco íngreme e de fácil leitura e escrita. Foram essas características que proporcionaram a linguagem a construir uma forte e ativa comunidade de usuários [19].

A partir do crescimento dessa comunidade, vieram as principais características que levaram essa linguagem a ser escolhida para ser usada na construção da ferramenta desse projeto. A linguagem Python possui uma extensa coleção de bibliotecas e módulos, sobretudo na área de computação científica, ciência de dados e inteligência artificial. Outro ramo forte da linguagem é a capacidade de automação de tarefas.

A combinação dessas características com os requisitos do projeto a tornaram a linguagem ideal para o desenvolvimento da ferramenta. A versão atual, 3.10, está em *bugfix* segundo o site oficial da linguagem, ou seja, em status de manutenção e correção ativo para lançamento de novas funcionalidades [20]. A versão na qual a ferramenta foi construída foi a 3.8.2, versão com status *security*, que indica estabilidade da versão, apenas abrindo para a possibilidade de modificação em caso de correção de problemas de segurança exploráveis por atacantes maliciosos.

3.4. Bibliotecas Utilizadas

Como visto, a linguagem Python tem nas bibliotecas disponíveis para desenvolvimento grande parte do seu potencial e flexibilidade. Abaixo serão descritas as bibliotecas utilizadas na construção da ferramenta proposta neste trabalho acompanhadas de uma breve descrição do seu funcionamento e o uso no projeto:

- **IMAPclient:** a biblioteca promove e facilita o uso de um cliente do protocolo Internet Message Access Protocol (IMAP) [21]. O protocolo IMAP sincroniza os e-mails do servidor do provedor com o dispositivo em uso. Dentro do projeto é utilizada para fazer login no servidor do provedor utilizando as credenciais do usuário, sincronizar os e-mails do servidor (no projeto definidos como os não-lidos) e realizar a marcação de leitura dos e-mails após a análise.
- **Pyzmail:** biblioteca bem documentada que provê funções e classes que ajudam a analisar, compor e enviar e-mails, principalmente ao diminuir as dificuldades de manipulação da estrutura Multipurpose Internet Mail Extensions (MIME) [22]. Essa estrutura é um padrão de formato de mensagem para transmissão através da internet que foi adicionado ao Simple

Mail Transport Protocol (SMTP) para permitir aumentar as possibilidades de transmissão de conteúdos no e-mail para outros formatos além de texto. No desenvolvimento da ferramenta a Pyzmail foi utilizada para capturar o corpo em texto do e-mail recebido para posterior análise.

- **BeautifulSoup:** biblioteca para navegação, busca e extração de dados nos formatos HTML e XML [23]. Na construção do programa é utilizada na extração de texto da porção HTML do e-mail recebido.
- **Email:** biblioteca nativa da linguagem Python para realizar o gerenciamento e manipulação de e-mails, mas não envio. Dentro do código da ferramenta possui função na construção da estrutura do e-mail dado como resposta na saída do processamento.
- **SMTPlib:** biblioteca utilizada para para realizar a comunicação com servidores de e-mail e enviar e-mails [24]. No código da ferramenta desenvolvida uma instância da classe principal da biblioteca é utilizada para realizar o disparo do e-mail de resposta após feita a análise e processamento do e-mail de entrada.
- **Pandas:** popular ferramenta open source de análise e manipulação de dados. No desenvolvimento da ferramenta foi utilizada para importação e interações com a planilha de exemplos previamente construída.
- **Scikit-learn:** popular, simples e eficiente biblioteca para realizar manipulação, análise, pré-processamento de dados e desenvolvimento de soluções em aprendizado de máquina [25]. No desenvolvimento da ferramenta proposta é utilizada para realizar o pré-processamento do texto do e-mail de entrada e cálculo da similaridade entre os exemplos da planilha.
- **Tkinter:** biblioteca padrão de criação de interfaces para Python. Utilizada para realizar a construção da interface da ferramenta proposta.

3.5. Planilha

Uma planilha com exemplos de entrada de e-mails que mapeia essas entradas a saídas pré-construídas foi construída para ser acessada pelo código da ferramenta de apoio a resposta de e-mails. O formato escolhido para ela foi o ‘xlsx’, formato atual padrão do software Excel da Microsoft para planilhas sem macro.

Esse foi o formato escolhido pela necessidade dupla de acessibilidade, primeiramente do código à planilha e segundamente do usuário da ferramenta à planilha para manutenção dos exemplos e saídas. Além do Excel ser um dos aplicativos de manipulação de planilhas mais populares mundialmente, o formato ‘xlsx’ é acessível por meio de outros softwares populares como o open source LibreOffice e o crescente em popularidade manipulador de planilhas Google Sheets.

Na planilha, a coluna ‘EXEMPLO_ENTRADA’ traz exemplos de possíveis e-mails recebidos na caixa de entrada da secretaria do PPGEC. A coluna ‘RESPOSTA’ traz uma resposta criada para ser utilizada como resposta a esse texto de entrada de exemplo. A Tabela 1 mostra um exemplo da formatação e algumas linhas da planilha utilizada em testes da ferramenta.

Tabela 2 - Entradas da planilha inicial utilizada para testes

EXEMPLO_ENTRADA	RESPOSTA
Olá, eu gostaria de saber quando se inicia a matrícula para a pós de Engenharia de Dados e quais são os documentos necessários para efetuá-la. Agradeço desde já!	Prezado(a), a matrícula se inicia no dia 10/10/2022 e para realizá-la é necessário que traga como documentos: - Diploma ou certificado de conclusão - Documento original com foto - CPF
Eu tenho uma dúvida sobre a entrega dos pré-projetos de qualificação de mestrado. Qual a data de entrega e qual a data de apresentação? Qual o formato do envio?	Prezado(a), o prazo limite para entrega dos pré-projetos de qualificação de mestrado é 30/09/2022, se iniciando em 26/09/2022. O documento deve ser enviado contendo no máximo 15 páginas ao todo, deverá ser enviado ao e-mail da secretaria com o orientador como cópia. Como título, o e-mail deve ter "Documento de pré-projeto: Seu Nome" As datas de apresentação se iniciam em 10/10/2022 e vão até 14/10/2022.

Boa noite, eu gostaria de saber mais informações sobre o edital de seleção de pesquisadores para estágio pós-doutoral.	O Programa de Pós-graduação em Engenharia de Computação (PPGEC) da Universidade de Pernambuco anuncia o Edital para Seleção e Ingresso Regular de Pesquisadores de Pós-Doutorado ao Programa de Pós-Graduação em Engenharia da Computação (PPGEC) no segundo semestre letivo do ano de 2022. Mais informações serão divulgadas nos próximos dias.
Oi, bom dia, há um tempo atrás vi no site da pós sobre o lançamento de um edital para regime especial, tem previsão de lançamento?	O Programa de Pós-graduação em Engenharia de Computação (PPGEC) da Universidade de Pernambuco anuncia a abertura do processo de seleção para alunos especiais (turma 2022.2). Datas importantes: - 18 de Julho de 2022 a 01 de Agosto de 2022 - Inscrição Online - 02 de Agosto de 2022 a 04 de Agosto de 2022. - Análise da documentação dos candidatos - 04 de Agosto de 2022 - Divulgação dos resultados - 05 a 07 de Agosto de 2022 - Prazo para recursos - 08 de Agosto de 2022 - Divulgação do resultado final após recursos - 09 a 10 de Agosto de 2022 - Matrícula vínculo

Fonte: Próprio autor

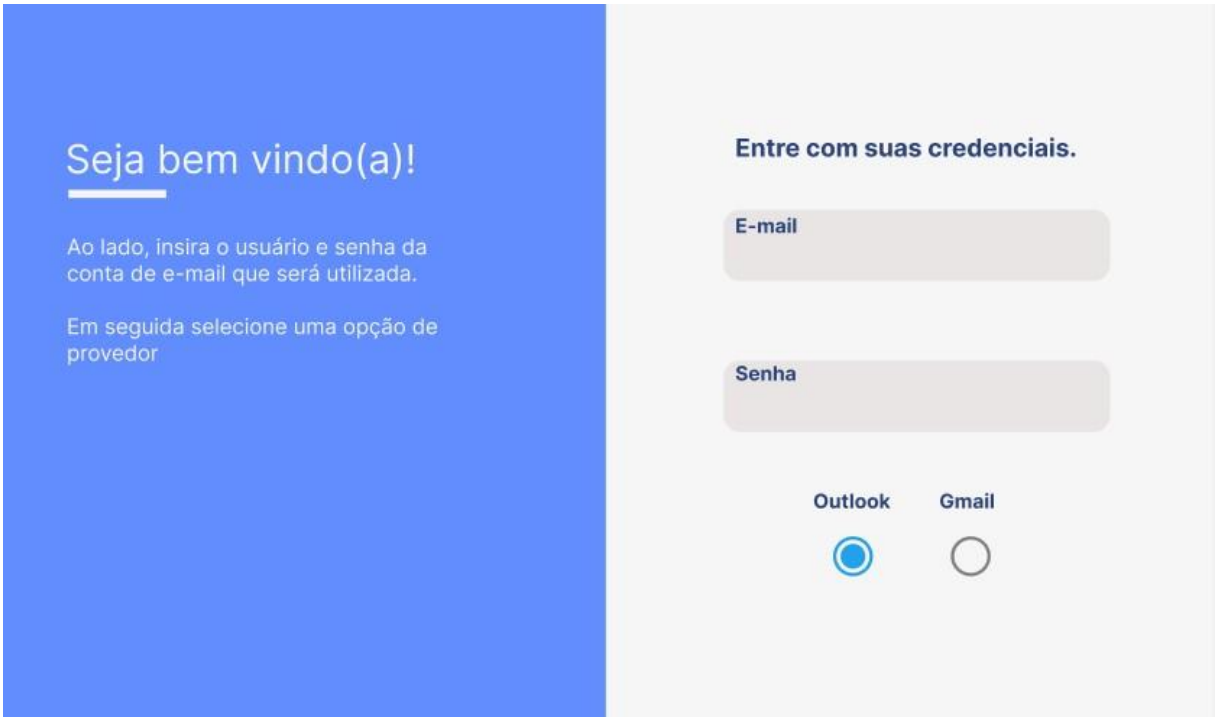
A ideia é que essa planilha seja dinamicamente construída, modificada para adaptar novos exemplos de e-mails de entrada e receber realimentação de exemplos após uma sugestão aceita a fim de aprimorar a obtenção de similaridade

entre o texto de um novo corpo de e-mail de entrada com os itens da coluna 1, além disso, esse caráter dinâmico proposto à planilha deve ser capaz de espelhar as variações das necessidades de respostas ao longo dos meses.

3.6. Interface Gráfica

Com interface gráfica simples, a ferramenta deve possuir apenas 3 telas: entrada de credenciais, seleção de modo de operação e edição de e-mail sugerido como resposta. A primeira tela, ilustrada pela Figura 7, capta as credenciais do usuário para permitir a conexão IMAP ao servidor do provedor do e-mail. Ela também coleta a opção de provedor do usuário, visto que o endereço do servidor é diferente de acordo com o provedor.

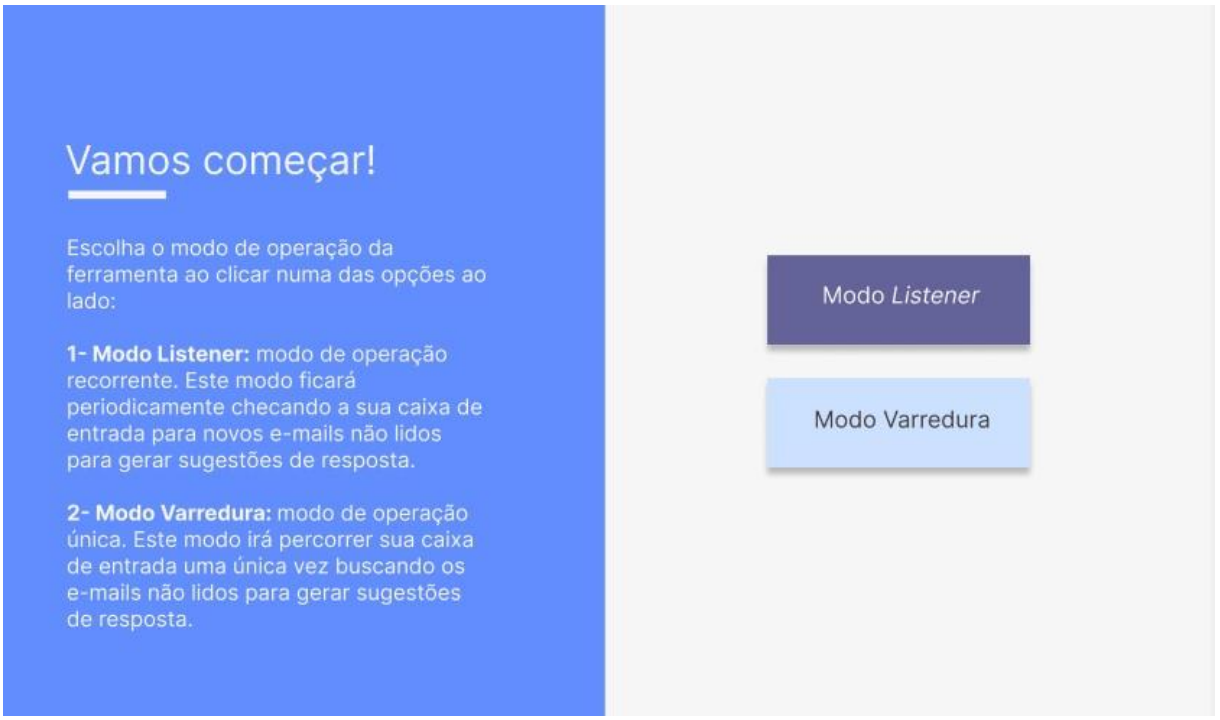
Figura 7- Tela de entrada de credenciais



Fonte: Próprio autor

Na tela de escolha de modo, mostrada na Figura 8, o usuário vai poder escolher o modo de operação da ferramenta entre *Listener* e Varredura. Após escolher uma das opções a ferramenta iniciará a execução do modo.

Figura 8 - Tela de escolha de modo



Fonte: Próprio autor

Por fim, a Figura 9 ilustra a tela de disparo de e-mail sugerido. Ela contém, um campo de “Para”, no qual o endereço de e-mail do usuário que originalmente enviou a mensagem à secretaria já vai estar preenchido e um campo de “Assunto”, no qual o assunto da mensagem original estará preenchido com o acréscimo de “Re:” antes dele, indicando uma resposta. Mais abaixo na imagem, à esquerda fica o campo que projeta o texto do e-mail de entrada que gerou a sugestão de resposta, e à direita a sugestão de resposta gerada pela ferramenta em si, apresentada em campo editável para permitir o acréscimo eventual de informações ao texto. Dois botões finalizam os componentes da tela, um que descarta a resposta sugerida e outro que aprova e realiza o disparo da resposta.

Figura 9 - Tela de edição de e-mail sugerido

Fonte: Próprio autor

4. Resultados e Testes

Nesta seção serão apresentados resultados práticos de funcionamento da ferramenta no que diz respeito à obtenção de similaridade entre o texto de um novo e-mail de entrada e os textos de exemplo presentes na planilha pré construída.

Inicialmente foram definidos 6 temas para popular a sugestão inicial da planilha:

- Questionamento sobre data de início da matrícula e documentos necessários para realizar uma pós em Engenharia de Dados no PPGEC;
- Questionamento sobre entrega dos pré-projetos de qualificação de mestrado;
- Questionamento sobre o calendário de defesas de mestrado;
- Questionamento sobre o edital de seleção de pesquisadores para estágio pós-doutoral;
- Questionamento sobre o Regime Especial;
- Questionamento sobre o formato em que as aulas seriam ministradas, presenciais ou remotamente.

Após a definição dos temas foi realizado um povoamento inicial da planilha com um exemplo de entrada e uma saída para cada tema de forma a realizar os testes iniciais da ferramenta.

A métrica escolhida para avaliar os resultados foi a de Precisão, definida pela divisão entre as classificações corretas e a soma das classificações corretas com as classificações incorretas $\frac{CC}{CC + CI}$. Os resultados iniciais obtidos foram bastante satisfatórios do ponto de vista de precisão, como mostra a Tabela 2.

Tabela 3 - Testes iniciais da ferramenta

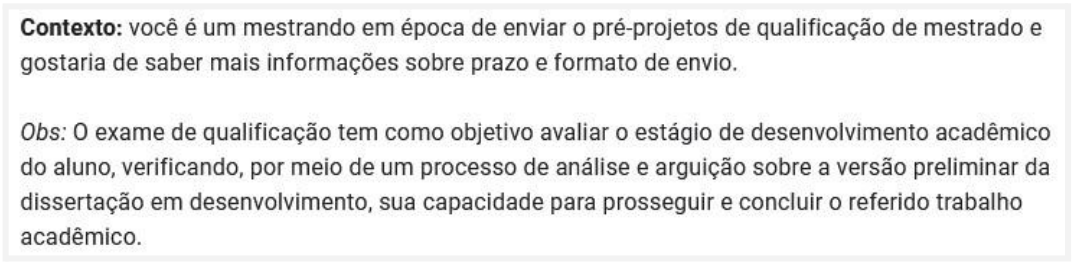
	Tema 1	Tema 2	Tema 3	Tema 4	Tema 5	Tema 6
Classificações certas	5	4	5	5	3	4
Classificações erradas	0	1	0	0	2	1
Tentativas	5	5	5	5	5	5
Precisão	100%	80%	100%	100%	60%	80%
Média da precisão	86,67%					

Fonte: Próprio autor

Essa primeira rodada de testes, porém, sofria de um alto viés por a mesma pessoa que inseriu os exemplos iniciais na planilha, ter sido a pessoa quem formulou os textos de teste.

Foi criado e distribuído, então, um formulário como tentativa de diminuição de viés. O formulário possuía instruções sobre como deveria ser preenchido e 6 questões a serem respondidas em formato de texto longo pelos usuários, uma para cada tema inicial proposto para os testes. Em cada uma das questões era dado um contexto no qual o usuário deveria se imaginar para redigir um e-mail que seria direcionado a secretaria do PPGEC em busca da informação descrita no contexto dado como nos exemplos mostrados nas Figuras 10 e 11.

Figura 10 - Exemplo de contexto provido no formulário



Fonte: Próprio autor

Figura 11 - Exemplo de contexto provido no formulário

Contexto: o cenário é de pandemia e entre incertezas, você quer saber se as aulas da pós graduação que você irá iniciar serão ministradas de forma presencial ou remota

Fonte: Próprio autor

Foram recebidas como respostas 50 instâncias de texto distribuídas entre os 6 temas. A primeira medida foi pegar uma instância aleatória de resposta sobre cada tema e realimentar a planilha de exemplos. Em seguida foram realizados novos testes com o restante das instâncias de respostas recebidas no formulário. A Figura 12 mostra um exemplo dos testes realizados.

Figura 12 - Exemplo de saída de teste

Novo e-mail de entrada: Olá, boa tarde. Sou aluno do mestrado e a data da qualificação está se aproximando. Dito isso, queria saber informações mais detalhadas à respeito do formato do arquivo a ser enviado e do prazo final para entrega do material. Atenciosamente, Bruno Luiz.

Similaridade: 0.23748808076756153

Linha correspondente na planilha: 11

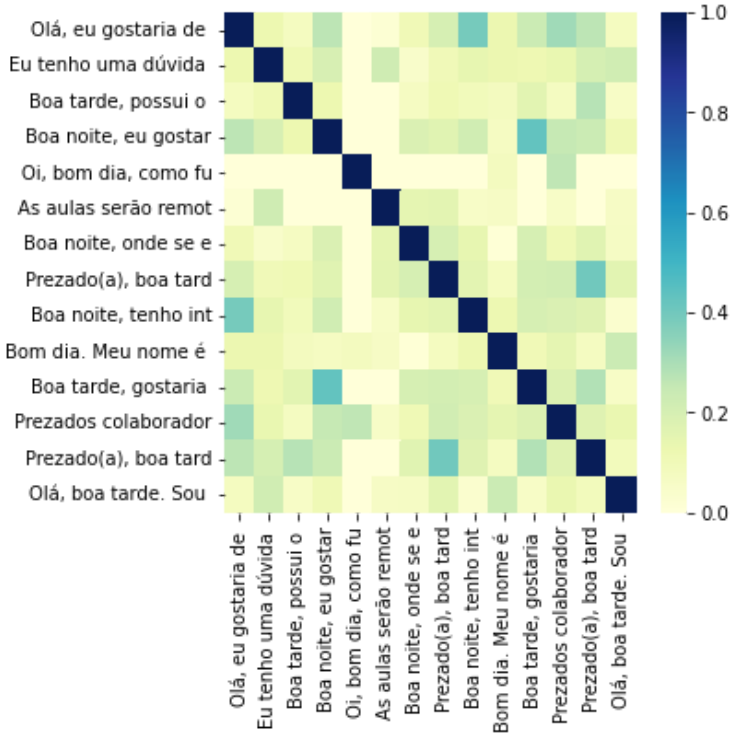
Exemplo mais similar: Bom dia. Meu nome é Luiz Cordeiro e participo do mestrado na própria POLI-UPE. Queria saber as datas certinhas para o envio do pré-projeto de qualificação, pois queria me organizar melhor em relação a isso para conciliar com o trabalho e entregar certinho todos os documentos. Att,

Resposta: Prezado(a), o prazo limite para entrega dos pré-projetos de qualificação de mestrado é 30/09/2022, se iniciando em 26/09/2022. O documento deve ser enviado contendo no máximo 15 páginas ao todo, deverá ser enviado ao e-mail da secretaria com o orientador como cópia. Como título, o e-mail deve ter "Documento de pré-projeto: Seu Nome". As datas de apresentação se iniciam em 10/10/2022 e vão até 14/10/2022.

Fonte: Próprio autor

E na Figura 13 mapa de calor dos valores das similaridades para o exemplo da Figura 12.

Figura 13 - Mapa de calor das similaridades de um exemplo de teste



Fonte: Próprio autor

Os resultados obtidos com o restante das instâncias de respostas recebidas no formulário são apresentados na Tabela 3.

Tabela 4 - Testes da ferramenta com dados do formulário

	Tema 1	**Tema 2**	**Tema 3**	**Tema 4**	**Tema 5**	**Tema 6**
Classificações certas	7	4	4	5	5	6
Classificações erradas	1	3	3	2	2	2
Tentativas	8	7	7	7	7	8
Precisão	87,50%	57,15%	57,15%	71,43%	71,43%	75,00%
Média da precisão	69,95%					

Fonte: Próprio autor

5. Conclusão e Trabalhos Futuros

Neste trabalho foi descrito o processo de desenvolvimento de uma ferramenta de apoio a resposta de e-mails por meio da comparação do texto de um novo e-mail de entrada com exemplos contidos numa planilha. Essa comparação para obtenção da similaridade entre os textos é feita utilizando técnicas de Processamento de Linguagem Natural e Estatística.

Durante a realização dos testes, inicialmente foi obtida uma precisão média de 86,67%. Numa segunda rodada de testes, dessa vez realizados utilizando dados coletados via formulário, foi obtida uma precisão média de 69,95%. Explorar técnicas mais robustas que tenham potencial de aprimorar a obtenção de similaridade textual e classificação dos textos. Uma possibilidade para isso é fazer uso de conceitos de Redes Complexas para classificação de textos [26] e representação textual [27]. Esse tipo de abordagem realiza a modelagem dos textos como redes, após retirada de *stopwords,* e representa o texto como uma matriz ponderada para obter índices de similaridade baseados em características tanto semânticas quanto topológicas da rede, levando em consideração os termos vizinhos. A modelagem de textos como redes de adjacência de palavras tem se mostrado útil não apenas para entender mecanismos fundamentais da linguagem, mas também para aperfeiçoar aplicações reais quando combinada com métodos tradicionais de processamento de texto [28].

Em se tratando da ferramenta em si, uma outra possibilidade de melhoria seria permitir ao usuário a inclusão de uma lista de endereços de e-mails que não precisam ter uma nova mensagem que chega à caixa de entrada checada pela ferramenta. Isso diminuiria a quantidade de descartes a sugestões feitas pela ferramenta, devido a quantidade de e-mails de spam, empresas, anúncios, que são recebidos no dia a dia dos usuários de e-mail. Uma outra proposta é possibilitar a inclusão de anexos na etapa de edição da sugestão de resposta, antes do envio ou descarte do e-mail.

Em 2017 a gigante Google abriu para os desenvolvedores a possibilidade de criar *add-ons* para o Gmail, o que posteriormente evoluiu para o desenvolvimento de *add-ons* para o Google Workspace em geral por meio da API oficial ‘Google

Workspace Add-ons API'. Portanto, uma possibilidade futura seria desenvolver a ferramenta em formato de add-on integrado ao Gmail, visto que uma das possibilidades que a API oficial oferece é a de executar tarefas em segundo plano usando o serviço de hospedagem dos *add-ons* [29].

Nos testes realizados durante o desenvolvimento, o projeto indicou ter potencial de apoiar um usuário na tarefa de resposta de e-mails com temática frequente, porém um teste real com o usuário final do estudo de caso proposto, que seria um funcionário da secretaria do Programa de Pós Graduação em Engenharia da Computação, fica como trabalho futuro como forma de observar a ferramenta em funcionamento em ambiente de produção.

Referências

[1] KUROSE, JAMES F., ROSS, KEITH W. **Redes de computadores e a Internet: uma abordagem top-down.** 5. ed. São Paulo: Addison Wesley, 2010.

[2] The Radicati Group. **Email Statistics Report, 2019-2023**. Disponível em: https://www.radicati.com/wp/wp-content/uploads/2018/12/Email-Statistics-Report-2019-2023-Executive-Summary.pdf. Acesso em: 17 out. 2022

[3] DÜRSCHEID, C., FREHNER, C. **Email communication.** Zurich. Pragmatics of Computer-Mediated Communication. 2013.

[4] Bhbosale, S., Pujari, V., Multani, Z. **Advantages And Disadvantages Of Artificial Intelligence.** Khed, Índia. Aayushi International Interdisciplinary Research Journal. 2020.

[5] Balkan, M. **Forwarding emails automatically based on content with Amazon Simple Email Service.** AWS Messaging & Targeting Blog. Disponível em: https://aws.amazon.com/blogs/messaging-and-targeting/forwarding-emails-automatically-based-on-content-with-amazon-simple-email-service/. Acesso em: 17 out. 2022

[6] SWEIGART, A. **Automate the boring stuff with python: practical programming for total beginners.** 2. ed. San Francisco: No Starch Press. 2015

[7] Cambria, E., White, B. **Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article].** Califórnia, EUA. IEEE Computational Intelligence Magazine. 2014.

[8] Corballis, M. **Language Evolution: A Changing Perspective.** EUA**.** Trends in Cognitive Sciences. 2017.

[9] MÜHLHÄUSER, M., GUREVYCH, I. **Introduction to Ubiquitous Computing.** EUA. Information Science Reference. 2008.

[10] WANG, J., DONG, Y. **Measurement of Text Similarity: A Survey**. Ningbo, China. Information, 2020.

[11] GOMAA, W., FAHMY, A. **A Survey of Text Similarity Approaches.** Cairo, Egito. International Journal of Computer Applications. 2013.

[12] Harish, B., Guru, D., Manjunath, S. **Representation and Classification of Text Documents: A Brief Review**. Mysore, Índia. International Journal of Computer Applications. 2010.

[13] Cleopatra, D. **Finding Word Similarity using TF-IDF and Cosine in a Term-Context Matrix from Scratch in Python.** Medium. Disponível em: https://towardsdatascience.com/finding-word-similarity-using-tf-idf-in-a-term-context-matrix-from-scratch-in-python-e423533a407. Acesso em: 26 set. 2022

[14] LUIZ, R. **Distância entre dois pontos**. Brasil Escola. Disponível em: https://brasilescola.uol.com.br/matematica/distancia-entre-dois-pontos.htm. Acesso em 25 set. 2022.

[15] Paiva, I. **Similaridade Manual de Observações**. Medium. Disponível em: https://medium.com/@iuricpaiva_3212/similaridade-manual-de-observa%C3%A7%C3%B5es-9e18897fe7b5. Acesso em: 26 set. 2022

[16] Goyal, S. **Recommendation System**. California State University. Disponível em: http://csweb01.csueastbay.edu/~uv8356/tutorial/collaborative.html. Acesso em: 26 set. 2022

[17] Luu, V., Forestier, G., Weber, J., Bourgeois, P., Djelil, F., Muller, P. **A review of alignment based similarity measures for web usage mining**. Mulhouse, França. Artificial Intelligence Review. 2020.

[18] CASS, S. **Top Programming Languages 2022**. Spectrum IEEE. Disponível em: https://spectrum.ieee.org/top-programming-languages-2022. Acesso em: 30 set. 2022

[19] Simplilearn. **14 Most Important Python Features and How to Use them.** Simplilearn. Disponível em: https://www.simplilearn.com/python-features-article. Acesso em: 30 set. 2022

[20] Python Software Foundation. **Status of Python branches**. Disponível em: https://devguide.python.org/#status-of-python-branches. Acesso em: 01 out 2022

[21] Menno, F. **IMAPClient 2.3.1 Documentation**. Read the Docs. Disponível em: https://imapclient.readthedocs.io/en/master/. Acesso em: 01 out 2022

[22] Spineux, A. **Pyzmail 1.0.3 Documentation.** Pypi. Disponível em: https://pypi.org/project/pyzmail/. Acesso em: 01 out 2022

[23] Richardson, L. **Beautifulsoup4 4.11.1 Documentation.** Pypi. Disponível em: https://pypi.org/project/beautifulsoup4/. Acesso em: 01 out 2022

[24] PyMOTW. **SMTPlib Documentation**. PyMOTW. Disponível em: http://pymotw.com/2/smtplib/. Acesso em: 01 out 2022

[25] Scikit-Learn Project. **Scikit-Learn scikit-learn 1.1.2 Documentation**. Scikit-Learn. Disponível em: https://scikit-learn.org/stable/index.html. Acesso em: 01 out 2022

[26] Amancio, D., Oliveira, O., Costa, L. **Structure–semantics interplay in complex networks and its effects on the predictability of similarity in texts**. São Carlos, Brasil. Physica A: Statistical Mechanics and its Applications. 2012.

[27] AMANCIO, D. **Classificação de textos com redes complexas**. São Carlos, Brasil. Instituto de Física de São Carlos. 2013.

[28] Google Workspace for Developers. **Google Workspace Add-ons.** Disponível em: https://developers.google.com/workspace/add-ons/overview. Acesso em: 25 out 2022.