Pontifícia Universidade Católica do Rio de Janeiro
Departamento de Informática

Projeto final de programação

Modelo para Teoria Social Cognitiva

Wallace Albertini

Orientador: Marcos Kalinowski

Rio de Janeiro

Dezembro de 2024

Sumário

1 Breve descrição

Este programa foi desenvolvido como um sistema para análise, classificação e avaliação de dados comportamentais com base em modelos psicológicos, como o TSC (Teoria Social Cognitiva). Ele oferece uma solução integrada para pesquisadores e profissionais interessados em realizar análises contextuais com apoio de aprendizado de máquina e recuperação de informações.

1.1 Usuários Primários

Este programa foi projetado principalmente para:

- Pesquisadores e profissionais das áreas de psicologia, educação e análise de comportamento.
- Professores e estudantes interessados em métodos de análise qualitativa e quantitativa, especialmente em níveis de graduação e pós-graduação.

1.2 Natureza do Programa

- Prova de conceito: Este é um sistema funcional que demonstra a integração entre técnicas de machine learning, modelos de linguagem natural e ferramentas de recuperação de informações.
- Ferramenta utilitária: Atualmente, encontra-se em fase inicial de desenvolvimento e pode ser estendido para se tornar uma solução mais robusta.
- Ressalvas:
  - O sistema depende de conexões externas para serviços como a API do OpenAI, o que pode limitar seu uso em ambientes offline.
  - A classificação automática pode apresentar variações dependendo da qualidade dos dados de entrada e do treinamento dos modelos subjacentes.

2 Visão do Projeto

2.1 Cenário Positivo 1:

Julia é uma pesquisadora em psicologia comportamental interessada em categorizar transcrições de entrevistas qualitativas para uma análise temática. Ela utiliza o programa para

carregar um arquivo Excel contendo várias transcrições, pré-processa os dados e inicia o sistema de classificação. Com base na recuperação de informações relevantes e na interação com o modelo GPT, Julia recebe as classificações acompanhadas de justificativas claras. Ela usa essas informações para validar suas hipóteses e, ao final, utiliza as métricas de avaliação para verificar a precisão e consistência das classificações geradas. Julia conclui o estudo com dados bem organizados e insights valiosos para seu artigo científico.

2.2 Cenário Positivo 2:

Marcos, um professor universitário em educação, precisa ensinar aos alunos de graduação como analisar comportamentos observados em contextos educacionais. Ele usa o programa para apresentar exemplos reais de entrevistas e transcrições, permitindo que os alunos entendam como o modelo TSC pode ser aplicado em cenários práticos. Após classificar exemplos, Marcos revisa as respostas com os alunos, destacando erros e acertos do modelo. Ele também utiliza a função de recuperação de contextos relacionados para promover discussões sobre como diferentes fatores afetam o comportamento. Isso enriquece a aula e melhora a compreensão dos alunos sobre o tema.

2.3 Cenário Negativo 1:

Marina é uma estudante de psicologia que tenta usar o programa para analisar dados qualitativos, mas encontra problemas ao carregar o arquivo Excel. O programa retorna uma mensagem de erro indicando que o formato das colunas ou a nomenclatura do arquivo não está correta. Marina tenta corrigir o problema, mas como não possui experiência com manipulação de dados, fica frustrada e desiste temporariamente de usar o programa, optando por buscar ajuda com seu orientador ou um colega.

2.4 Cenário Negativo 2:

Mauro, um aluno de mestrado, utiliza o programa para categorizar exemplos de comportamento observados em uma empresa. Ele espera resultados rápidos e precisos, mas percebe que a classificação automática, em alguns casos, não reflete o contexto esperado, devido à natureza limitada do treinamento do modelo ou à falta de exemplos similares no banco de dados. Apesar de tentar refinar os resultados utilizando a função de recuperação de contextos, Mauro conclui que a ferramenta precisa de uma maior personalização para atender adequadamente às especificidades do seu campo de pesquisa.

3 Especificação de Requisitos Funcionais e Não-Funcionais do Software

3.1 Requisitos Funcionais:

1. Carregamento de Dados: O sistema deve ser capaz de carregar dados de planilhas Excel, com múltiplas abas, contendo citações e categorias.

2. Pré-processamento de Dados: Após carregar os dados, o sistema realiza um pré-processamento, incluindo a extração de citações e a limpeza dos dados.

3. Análise de Texto: O sistema deve ser capaz de classificar as citações de acordo com categorias predefinidas da Teoria Social Cognitiva (TSC): Reciprocal Determinism, Observational Learning (Modeling), Reinforcement, Self-efficacy, Outcome Expectations, Behavioral Capability, e Environmental Factors.

4. Interface de Usuário: Deve ser fornecida uma interface simples e interativa utilizando o Streamlit, permitindo que o usuário insira uma consulta e receba a classificação da citação.

5. Consulta ao Modelo: O sistema deve permitir que o usuário insira uma questão (citação) e receba uma resposta do modelo de linguagem (GPT-4) com base nos dados pré-processados.

6. Armazenamento de Embeddings: O sistema deve ser capaz de armazenar os embeddings dos dados utilizando o ChromaDB para consulta e recuperação de informações relevantes.

7. Exibição de Resultados: Após a análise, a resposta do modelo deve ser exibida na interface com a classificação e justificativa em negrito.

3.2 Requisitos Não-Funcionais:

1. Desempenho: O sistema deve ser capaz de realizar a análise em tempo razoável para entradas de tamanho moderado.

2. Escalabilidade: O sistema deve ser projetado para suportar aumento de dados, com capacidade de carregar grandes arquivos Excel e processar múltiplas consultas simultâneas.

3. Usabilidade: A interface de usuário deve ser simples, com feedback claro durante o processo de análise.

4. Segurança: A chave de API para o modelo de linguagem (OpenAI) deve ser gerida de forma segura para evitar exposições indevidas.

5. Compatibilidade: O sistema deve ser compatível com ambientes Python, utilizando bibliotecas como streamlit, pandas, chromadb, openai, entre outras.

4 Descrição ou Modelo de Arquitetura, Dados, Semântica ou Outra Dimensão Relevante do Software

4.1 Arquitetura

O sistema é dividido em várias camadas e módulos, cada um com uma responsabilidade específica:

1. Módulo de Carregamento de Dados (DataLoader):
   - Responsável por carregar dados de arquivos Excel.
   - Extrai citações e categorias para posterior processamento.
2. Módulo de Pré-processamento (PreProcessor):
   - Responsável por realizar o pré-processamento das citações, incluindo a conversão de texto em embeddings (vetores).
   - Utiliza modelos como o all-MiniLM-L6-v2 para gerar embeddings a partir das citações.
3. Módulo de Recuperação de Dados (Retriever):
   - Utiliza o ChromaDB para armazenar e recuperar os embeddings das citações, realizando consultas para encontrar os dados mais relevantes com base nas entradas do usuário.
4. Módulo de Modelo de Linguagem (OpenAiModel):
   - Faz a consulta ao modelo GPT-4 da OpenAI para gerar respostas baseadas nos dados recuperados.
5. Módulo de Prompter:
   - Cria as mensagens de prompt que são enviadas ao modelo GPT-4 para garantir que ele produza uma resposta relevante e formatada corretamente.

6. Interface de Usuário (View):
    - Fornece a interface com o usuário utilizando o Streamlit, permitindo a interação do usuário com o sistema de forma simples e intuitiva.

4.2 Dados

- Entrada: Dados no formato de planilhas Excel, contendo citações (quotes) e suas respectivas categorias.
- Saída: A classificação das citações em uma das sete categorias da Teoria Social Cognitiva, com uma justificativa fornecida pelo modelo.

4.3 Descrição ou Modelo Funcional do Software:

O software segue um modelo funcional baseado no processo de consulta a um sistema de recuperação de dados e geração de resposta:

1. Carregamento de Dados: O Controller carrega os dados de um arquivo Excel, iniciando a base de dados para processamento.
2. Pré-processamento: As citações são processadas e convertidas em embeddings que serão usados para consultas de similaridade.
3. Recuperação de Dados: Quando o usuário faz uma consulta, o sistema recupera os dados mais relevantes com base nos embeddings armazenados.
4. Classificação e Resposta: O modelo GPT-4 gera uma resposta classificada com base nos dados recuperados, e a interface exibe a resposta ao usuário.

4.4 Sobre o Código:

- Linguagem: Python 3.x.
- Bibliotecas:
    - Streamlit: Utilizada para criar a interface de usuário interativa.
    - Pandas: Para manipulação de dados em formato DataFrame, especialmente no carregamento e processamento das planilhas.
    - ChromaDB: Para armazenamento e consulta de embeddings em um banco de dados vetorial.

- OpenAI: Para integração com o modelo GPT-4, utilizado para gerar respostas às consultas.
- Langchain: Utilizado para criação e manipulação de mensagens de prompt para o modelo de linguagem.

4.5 Estratégia de Comentários em Linha:

- Os comentários são utilizados de forma a descrever a função de cada bloco de código.
- Cada classe e método possuem descrições breves sobre sua responsabilidade no fluxo do programa.

4.6 Diretivas de Compilação:

- O código não requer compilação específica, sendo executado diretamente no ambiente Python com as bibliotecas necessárias instaladas.

5 Manual de Utilização para Usuários Contemplados

5.1 Tarefa 1: Carregar Dados de Planilhas Excel

5.1.1 Guia de Instruções:

Para carregar os dados de uma planilha Excel, siga os seguintes passos:

1. Abra o sistema e insira o arquivo Excel com as planilhas desejadas.
2. Certifique-se de que o arquivo Excel contém as colunas com as citações e as categorias adequadas.
3. No campo apropriado, insira o nome do arquivo Excel (por exemplo, Análise Temática de Conteúdo.xlsx) e as planilhas a serem processadas.
4. O sistema irá automaticamente processar as planilhas e extrair as citações e categorias.

Alternativa: Caso o sistema não reconheça corretamente as planilhas ou colunas, verifique se os nomes estão corretamente especificados e se o formato das colunas corresponde ao esperado.

5.1.2 Exceções ou Potenciais Problemas:

- Se o arquivo Excel não estiver no formato adequado (colunas de citações e categorias ausentes ou mal formatadas):
  - O sistema exibirá um erro informando que a leitura da planilha falhou. Verifique o formato das colunas e reenvie o arquivo.
  - É porque: O arquivo Excel não está estruturado corretamente, conforme esperado pelo sistema. As colunas devem seguir o padrão de 'Quote' e 'Categoria'.

5.2 Tarefa 2: Inserir uma Citação para Análise

5.2.1 Guia de Instruções:

Para inserir uma citação e analisar sua classificação, siga os seguintes passos:

1. No painel principal do sistema, localize o campo de entrada onde você pode digitar uma nova citação.
2. Insira a citação desejada no campo de texto.
3. Clique no botão Analisar para iniciar o processo de classificação.
4. O sistema exibirá a classificação da citação na forma de uma categoria (por exemplo, Self-efficacy) e uma justificativa.

Alternativa: Caso você deseje inserir mais de uma citação para análise, o sistema permite o processamento de várias citações de uma vez. Basta repetir os passos para cada citação.

5.2.2 Exceções ou Potenciais Problemas:

- Se a citação inserida for muito curta ou incompleta:
  - O sistema pode não conseguir gerar uma análise precisa. Insira uma citação mais completa ou detalhada para melhores resultados.
  - É porque: O modelo de linguagem precisa de contexto suficiente para realizar a classificação corretamente.
- Se o sistema não conseguir classificar a citação:
  - O sistema pode informar que a citação não se encaixa nas categorias predefinidas. Tente reformular ou fornecer mais contexto para a citação.

– É porque: O modelo de linguagem pode não ter informações suficientes para identificar a categoria com precisão.

6 Conclusão

Este trabalho apresentou o desenvolvimento de um sistema baseado na Teoria Social Cognitiva (TSC) com o objetivo de classificar exemplos comportamentais e contextuais de maneira automatizada. O projeto envolveu a integração de diversas tecnologias e metodologias modernas, como o uso de modelos de linguagem natural avançados, vetorização de textos com embeddings, armazenamento e recuperação eficiente de dados utilizando ChromaDB, e a construção de uma interface interativa para facilitar a utilização por usuários finais.

Durante o desenvolvimento, destacaram-se os seguintes aspectos:

1. Automatização e Precisão: O uso de modelos de linguagem como o GPT-4 e ferramentas como LangChain permitiu automatizar a análise textual com elevado nível de precisão, demonstrando a viabilidade da abordagem para cenários reais.

2. Modularidade e Escalabilidade: A arquitetura modular adotada facilita a reutilização e a expansão do sistema para novas categorias de análise ou diferentes domínios.

3. Interatividade e Acessibilidade: A criação de uma interface amigável utilizando Streamlit possibilitou que usuários com diferentes níveis de conhecimento técnico utilizassem a ferramenta de forma intuitiva.

4. Documentação e Usabilidade: Foi produzida uma documentação completa, incluindo manuais técnicos e de usuário, garantindo que o sistema seja facilmente compreendido e adaptado por futuros desenvolvedores e usuários.

O projeto demonstrou a eficácia da aplicação de técnicas de processamento de linguagem natural e inteligência artificial na resolução de problemas complexos de classificação e análise de dados qualitativos. Além disso, a ferramenta desenvolvida tem o potencial de contribuir significativamente para a análise comportamental, servindo como suporte em pesquisas acadêmicas, treinamentos corporativos e outras aplicações práticas.

Como trabalhos futuros, recomenda-se a inclusão de mais categorias na análise, o treinamento de modelos específicos com dados do domínio e a implementação de feedback contínuo dos usuários para refinar as respostas geradas. Acredita-se que este sistema pode ser uma base sólida para novas aplicações que integrem inteligência artificial e teorias comportamentais.