PUC

ISSN 0103-9741

Monografias em Ciência da Computação

nº 08/2023

Módulo de Processamento e Análise de Dados com Gerenciamento em Chunks para Redes Sociais

Alexandre Heine

Sérgio Lifschitz

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 08/2023 ISSN: 0103-9741

Módulo de Processamento e Análise de Dados com Gerenciamento em Chunks para Redes Sociais

Alexandre Heine, Sérgio Lifschitz

xandeaph@gmail.com.br, sergio@inf.puc-rio.br

Resumo. Este projeto aborda uma questão de Ciências de Dados e Engenharia de Software, pois os seus principais desafios são gerenciar o uso de espaço em memória RAM, enquanto são realizadas análises em dados de redes sociais, muito volumosos; e desenvolver um módulo que seja extensível para se adequar ao uso em outras análises além da análise de sentimentos. Também, foi objetivo satisfazer condições tanto em que os dados estavam em arquivo ou armazenados em banco de dados, permitindo que ele se comporte tanto como um *stand-alone* ou interagindo com um banco de dados. Como *stand-alone*, o programa se destina a outros desenvolvedores, que podem reutilizar a maioria dos componentes, apenas instanciando para alguma nova análise e alterando as configurações. Como produto integrado ao sistema EPARS, ele pode atender a demandas de diversos outros usuários que apenas precisariam interagir com a interface do sistema, mas como ainda está em desenvolvimento, não está disponível para uso, por enquanto. Assim, espera-se que este projeto seja reaproveitado tanto para usos pessoais quanto para outros sistemas e, por isso, seja utilizado por bastante tempo, principalmente por estudiosos da área de redes sociais, um nicho de pesquisa atualmente em alta.

Palavras-chave: Framework; Processamento de Linguagem Natural; Análise de Dados; Redes Sociais; Grandes Volumes de Dados

In charge of publications:
PUC-Rio Departamento de Informática - Publicações
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: publicar@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

Table of Contents

1 Descrição e Objetivos Gerais do Software

Este capítulo apresenta a motivação do projeto e sua finalidade, assim como os objetivos que visou alcançar durante seu desenvolvimento.

1.1 Motivação

Atualmente, redes sociais *online* envolvem um movimento de grandes quantidades de dados de diversos tipos muito rapidamente, conceito conhecido como Big Data. A partir disso, vários desafios podem ser encontrados, desde como armazenar esses grandes volumes a como analisar esses dados, visto que, para pesquisas, não é factível utilizar toda a base de dados em memória ao mesmo tempo.

Dentro desse contexto, o laboratório BioBD do Departamento de Informática da PUC-Rio desenvolveu uma ferramenta para coleta de dados do Twitter chamada eTC[2] (ePOCS Twitter Crawler) para pesquisas em mídias sociais e opinião pública em parceria com o Departamento de Comunicação, em 2015.

A ferramenta atendia às demandas de coleta e análises estatísticas, mas conforme a quantidade de dados cresceu, o processamento de grandes volumes, em torno de alguns milhões de tweets, se tornou cada vez mais difícil de gerenciar pela ferramenta e o modo de implementação das análises dificultava que fosse estendido.

Desse modo, como meio de aproveitar o conhecimento adquirido no desenvolvimento da ferramenta eTC, foi discutido o desenvolvimento da ferramenta EPARS (Ambiente de Extração Processamento e Análise de Redes Sociais), para reestruturação do projeto como um todo, integrando novas redes sociais e aumentando a qualidade do produto, com uma documentação realizada desde o início, testes automatizados e criação de uma infraestrutura flexível a mudanças e novas análises.

Assim, o projeto deste documento, se propõe a resolver alguns dos desafios levantados para desenvolvimento desde novo sistema, mas com ênfase no módulo de análise de dados: como lidar com esses volumes de modo que seja possível analisá-los sem ocorrer problemas de memória; e como tornar as análises reprodutíveis para outras bases de dados, sem que seja necessário recriar todo o processo desde o gerenciamento de dados até a análise.

1.2 Escopo

O software tem como objetivo o desenvolvimento de um módulo, em Python, para processamento e análise de dados, o qual seja capaz de lidar com grandes volumes de dados e permita reusabilidade de seus módulos para outras tarefas de análise de dados. Para tal, envolve tanto um desafio na área de Engenharia de Software no desenvolvimento de um módulo com características de framework, utilizando de *cold spots* e *hot spots* para determinar submódulos que não têm necessidade de serem alterados e podem ser reutilizados como estão para outras tarefas; e outros que, para diferentes análises, tem necessidade de serem modificados para melhor comportá-los.

Como também foram utilizadas classes abstratas e o conceito de interfaces, isso permite que os *hot spots* sejam classes que, principalmente, implementam uma dessas classes ou interfaces genéricas, havendo menos necessidade de alterações e tornando o código menos específico para a análise implementada no momento (análise de sentimentos).

Além disso, também foi necessária a implementação, como dito anteriormente, de uma análise para verificação do funcionamento do software, em que optou-se por uma análise de sentimentos, utilizando o modelo de linguagem BERTimbau[3], uma variação do modelo de linguagem BERT[1], treinado para o português brasileiro, utilizando de *datasets* disponíveis no Corpus Carolina[1], um repositório da USP para bases de dados em português brasileiro.

Assim, o escopo do projeto também está na área de Ciências de Dados, já que foi necessário implementar e pensar no funcionamento das análises para sua estruturação, além de ter sido desenvolvida uma API para manter o modelo ativo (não ser necessário colocá-lo e retirá-lo da memória sempre que fosse processada uma análise nova), para comunicação com o módulo de gerenciamento dos dados e para o pré-processamento dos dados antes de submetê-los ao modelo.

1.3 Público Alvo e Durabilidade

O projeto é um módulo com o objetivo de refatorar e aprimorar funcionamento do processamento de análises do sistema eTC. Tendo isso em vista, o público alvo do sistema original também fará uso, eventualmente, do que foi desenvolvido neste projeto e, sabendo que ano passado o eTC foi utilizado por universidades de mais de 12 estados diferentes, quando o EPARS atingir a fase de produção, será ainda mais atrativo ao público pela diversidade de redes sociais que ele propõe fazer coleta e análise de dados.

Assim, este projeto, terá um público que o usará indiretamente pela interface do sistema. Ao mesmo tempo, desenvolvedores ou pessoas com conhecimento em programação, podem reutilizar o código para tarefas próprias.

Em suma, o público alvo desse projeto são tanto pesquisadores em dados de redes sociais quanto pessoas com conhecimento de programação para uso próprio e a vida média do projeto está vinculada ao interesse em do público em análises em redes sociais.

2 Especificação de Requisitos

2.1 Requisitos Funcionais

[RF1] O sistema deve ser capaz de se comunicar tanto com arquivos quanto com bancos de dados.

Descrição: A ferramenta deve permitir que o usuário configure as conexões de entrada e saída de dados da análise a um arquivo ou banco de dados. Assim, o uso de dados para análises e posterior armazenamento de resultados deve ser flexível, de acordo com as configurações feitas.

[RF2] O sistema deve ser capaz de ler e salvar os dados nos padrões de arquivo CSV, XLSX, JSON, HTML e XML.

Descrição: Dentre os arquivos que a ferramenta deve poder ler ou salvar, devem estar esses formatos, pois são comumente utilizados na estruturação de dados de várias fontes, inclusive redes sociais.

[RF3] O sistema deve ser capaz de realizar o processamento de dados em *chunks*.

[1] https://sites.usp.br/corpuscarolina/repositorios-2023/

Descrição: Com o objetivo de garantir que o sistema não vai sobrecarregar os servidores durante uma análise, deve ser capaz de tratar dados em partes menores (*chunks*), de tal modo que não prejudique outras operações dos demais sistemas que compartilham do servidor.

[RF4] O sistema deve ser capaz de processar os dados para uma determinada análise.

Descrição: Os dados carregados no sistema devem ser processados para estar em conformidade com a análise desejada. Como, no momento, o sistema está integrado apenas a uma análise de sentimentos, o processamento (redução de colunas, filtragem dos dados) deve ser realizado ao menos para essa análise.

[RF5] O sistema deve prover um resultado ao final de uma análise e atualizar o estado da análise.

Descrição: Ao terminar uma análise, deve ser salvo o resultado em arquivo ou banco de dados e a análise deve passar do estado de "espera" para o estado de "finalizada".

2.2 Requisitos Não Funcionais

2.2.1 Robustez

[RNF1] O sistema não deve ultrapassar os limites de memória RAM disponibilizados pelo servidor.

Descrição: Dado que o sistema gerencia a quantidade de dados utilizada por meio de *chunks*, ele deve resolver questões de memória RAM no limite, conseguindo viabilizar o seu uso junto a outros sistemas sem que prejudique aos outros dessa forma.

[RNF2] O sistema deve ser estável.

Descriçao: O sistema deve ser estável, de tal modo que não ocorram travamentos ou pare de funcionar inesperadamente.

2.3 Escalabilidade

[RNF3] Grandes volumes de dados não devem afetar a eficácia do sistema.

Descrição: Devido à solução com *chunks*, o sistema deve ser capaz de lidar com grandes volumes de dados, os dividindo em partes menores.

2.4 Manutenibilidade

[RNF4] O sistema deve ser flexível para adaptação a novas análises.

Descrição: A partir do uso de interfaces e classes abstratas, deve ser possível adaptar o sistema a novas análises sem a necessidade de mudanças das demais classes do sistema.

2.5 Interoperabilidade

[RNF5] O sistema deve permitir sua integração a outros softwares.

Descrição: Como tem-se por objetivo integrá-lo ao EPARS, o sistema deve ser integrável a esse sistema por meio de banco de dados, por exemplo.

3 Modelo de Arquitetura

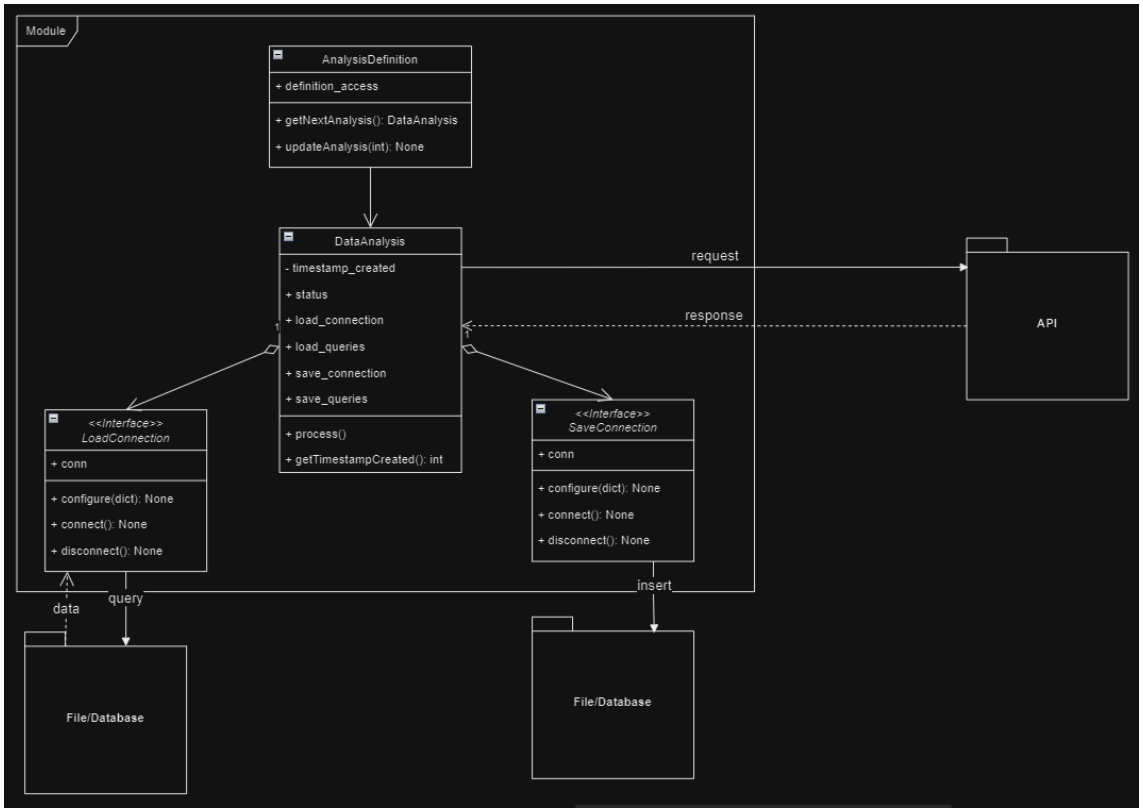

Figure 1: Arquitetura do Software

Assim como na imagem 1, o funcionamento do software ocorre da seguinte forma:

- 1 - O usuário define as configurações de conexão à definição da análise.
- 2 - Dada a definição da análise, são obtidas as informações de conexão para carregamento de dados na aplicação e informações de conexão para salvamento dos dados resultantes da análise.
- 3 - O módulo de análise faz o carregamento dos dados, obtendo somente chunks, pouco a pouco, como na imagem 2.
- 4 - A cada obtenção de chunks, é enviada uma requisição à API com os dados do chunk.
- 5 - A API faz o pré-processamento próprio dos dados e retorna o resultado.
- 6 - Dado o resultado parcial, o módulo de análise solicita que eles sejam salvos num arquivo ou num banco de dados.

```
def process(self):
    # Initializes the query connection
    self.load_connection.initializeQueryConnection(
        query=self.load_queries.sentimentAnalysisQuery()
    )

    # Gets the next data chunk
    data_chunk = self.load_connection.getNextDataChunk()

    while data_chunk is not None:
        # Analyzes the data chunk
        results = self.classify(data_chunk)

        self.save_connection.saveResults(
            results=results,
            query=self.save_queries.sentimentAnalysisInsert()
        )

        # Gets the next data chunk
        data_chunk = self.load_connection.getNextDataChunk()
```

Figure 2: Tabela de *Tweets* Tratada

References

[1] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805* (2018).

[2] Heine, A. A., Coutinho, B., Barreto, M., Xavier, N., Villas, M. V., Ituassu, A., and Lifschitz, S. Análise de dados para comunicação política a partir de um sistema de coleta de tweets. In *Anais Estendidos do XXXVI Simpósio Brasileiro de Bancos de Dados* (2021), SBC, pp. 49–55.

[3] Souza, F., Nogueira, R., and Lotufo, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)* (2020).