UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ANDRÉ JORDÃO DO NASCIMENTO

INGESTÃO E PROCESSAMENTO DE DADOS TEXTUAIS DO REDDIT: UMA SOLUÇÃO DE QUALIDADE E DISPONIBILIDADE

CAMPINA GRANDE - PB

2023

ANDRÉ JORDÃO DO NASCIMENTO

INGESTÃO E PROCESSAMENTO DE DADOS TEXTUAIS DO REDDIT: UMA SOLUÇÃO DE QUALIDADE E DISPONIBILIDADE

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Fábio Jorge Almeida Morais

CAMPINA GRANDE - PB

2023

ANDRÉ JORDÃO DO NASCIMENTO

INGESTÃO E PROCESSAMENTO DE DADOS TEXTUAIS DO REDDIT: UMA SOLUÇÃO DE QUALIDADE E DISPONIBILIDADE

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Fábio Jorge Almeida Morais

Orientador – UASC/CEEI/UFCG

Reinaldo Cezar de Morais Gomes

Examinador – UASC/CEEI/UFCG

Melina Mongiovi Cunha Lima Sabino

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 17 de Novembro de 2023.

CAMPINA GRANDE - PB

RESUMO

Um dos maiores problemas encontrados em aplicações que estão envolvidas no ecossistema de Big Data está relacionado à disponibilidade e qualidade de dados para modelos de IA e outras análises direcionadas. Aplicações com esse foco necessitam de dados que disponham de alta qualidade, já que o resultado de seus serviços depende da integridade da informação usada no processo. Quando pensamos em dados textuais, devemos saber que a informação fornecida para aplicações que envolvem processamento de texto, devem ser as melhores possíveis. Desta forma, foi desenvolvido uma aplicação que trata da gerência da coleta e tratamento contínuo de dados textuais. O contexto da aplicação está fixo na coleta de dados textuais da rede social Reddit. Através da API fornecida pela rede, é feita a ingestão de dados de uma comunidade específica. Com base nos dados coletados, a ferramenta trata de fazer todo o orquestramento de tarefas que gerenciam a coleta, tratamento e disponibilização desses dados. Para teste da ferramenta, os dados disponíveis são passados para um modelo de PLN, que usa LDA para mapear tópicos com base nos textos extraídos do site. A aplicação se baseia nos conceitos de streaming de dados e processamento de texto, de forma contínua e automática, a fim de manter uma base de dados sólida e de qualidade para análises de texto.

INGESTING AND PROCESSING TEXTUAL DATA FROM REDDIT: A QUALITY AND AVAILABILITY SOLUTION

ABSTRACT

One of the biggest problems encountered in applications that are involved in the Big Data ecosystem is related to the availability and quality of data for AI models and other targeted analyses. Applications with this focus need high-quality data, since the results of their services depend on the integrity of the information used in the process. When we think of textual data, we should know that the information provided to applications that involve text processing should be the best possible. An application has therefore been developed to manage the collection and ongoing processing of textual data. The context of the application is fixed on the collection of textual data from the Reddit social network. Using the API provided by the network, data is ingested from a specific community. Based on the data collected, the tool orchestrates all the tasks that manage the collection, processing and availability of this data. To test the tool, the available data is passed to a PLN model, which uses LDA to map topics based on the texts extracted from the site. The application is based on the concepts of streaming data and text processing, continuously and automatically, in order to maintain a solid, quality database for text analysis.

Ingestão e processamento de dados textuais do Reddit: Uma solução de qualidade e disponibilidade

André Jordão do Nascimento
Universidade Federal de Campina Grande
Campina Grande, Paraíba
andre.nascimento@ccc.ufcg.edu.br

Fabio Jorge Almeida Morais
Universidade Federal de Campina Grande
Campina Grande, Paraíba
fabio@computacao.ufcg.edu.br

RESUMO

Um dos maiores problemas encontrados em aplicações que estão envolvidas no ecossistema de Big Data está relacionado à disponibilidade e qualidade de dados para modelos de IA e outras análises direcionadas. Aplicações com esse foco necessitam de dados que disponham de alta qualidade, já que o resultado de seus serviços depende da integridade da informação usada no processo. Quando pensamos em dados textuais, devemos saber que a informação fornecida para aplicações que envolvem processamento de texto, devem ser as melhores possíveis. Desta forma, foi desenvolvido uma aplicação que trata da gerência da coleta e tratamento contínuo de dados textuais. O contexto da aplicação está fixo na coleta de dados textuais da rede social Reddit. Através da API fornecida pela rede, é feita a ingestão de dados de uma comunidade específica. Com base nos dados coletados, a ferramenta trata de fazer todo o orquestramento de tarefas que gerenciam a coleta, tratamento e disponibilização desses dados. Para teste da ferramenta, os dados disponíveis são passados para um modelo de PLN, que usa LDA para mapear tópicos com base nos textos extraídos do site. A aplicação se baseia nos conceitos de streaming de dados e processamento de texto, de forma contínua e automática, a fim de manter uma base de dados sólida e de qualidade para análises de texto.

Palavras-chave

Big Data, Processamento de linguagem Natural, ETL, Reddit.

Repositório

https://github.com/RTT-app

1. INTRODUÇÃO

Com a chegada da internet e os avanços tecnológicos ao seu redor, surgiu um grande aumento no fluxo de informações. Esse aumento se deve aos diversos geradores de dados que surgiram com o passar do tempo. Entre esses geradores, se destacam dispositivos IOT, aparelhos móveis e ambientes virtuais como as Redes sociais.

Quando relacionamos geradores de dados e as Redes sociais, um tipo de informação que se destaca são os dados textuais. Dados desse tipo são gerados a todo momento, em chats, publicações e comentários. Todos esses quadros possuem informações valiosas sobre o usuário, porém os dados gerados não possuem uma estrutura definida. Através de técnicas de processamento de linguagem natural (PLN) podemos extrair conhecimento valioso sobre os usuários da plataforma, e entender melhor como se organizam as informações dentro dela.

Para que esses dados ofereçam valor para análises, é necessário que sejam pré-processados e apresentem alta qualidade. Dito isso, no campo de PLN (Processamento de linguagem natural) existem algumas estratégias ligadas a extração de valor desses dados. Entre elas estão a Tokenização e o stemming. Através do uso desses procedimentos conseguimos mapear melhor o texto analisado, e prover qualidade aos dados gerados pela plataforma.

Além da qualidade, prover disponibilidade também é um dos desafios encontrados na área. Para garantir essas características, é essencial contar com um processo automatizado que extraia, adeque e persista as informações de forma contínua. Resultando em uma base de dados alinhada com o objetivo desejado.

Atualmente, as redes sociais representam os ambientes virtuais com o maior tráfego de pessoas, gerando uma quantidade expressiva de informações. Um exemplo notável é o Reddit. Ele é uma plataforma de mídia social e agregador de conteúdo onde os usuários podem compartilhar links, textos, imagens e vídeos em uma variedade de tópicos e comunidades, chamadas de "subreddits". Suas comunidades concentram inúmeros participantes, promovendo o engajamento social em diversos grupos, abrangendo tópicos como política, música, tecnologia, jogos, economia, entre outros. Nessas comunidades, as opiniões de milhões de usuários se entrelaçam, proporcionando um rico repositório de informações valiosas.

Para aproveitar essa riqueza de dados, o Reddit oferece acesso gratuito à sua API, que disponibiliza uma variedade de informações provenientes de suas comunidades. Dentre os dados disponíveis, destacam-se os textos, presentes nos títulos, conteúdos principais e comentários dos posts. Os comentários, em particular, podem fornecer insights valiosos sobre as opiniões presentes na comunidade, permitindo a compreensão do viés dos usuários.

Com base nesse conhecimento e na falta de ferramental para essa tarefa, foi desenvolvida uma ferramenta que trata da extração e disponibilidade de dados textuais provenientes dos comentários dos posts nas comunidades do Reddit. A ferramenta opera por meio de um pipeline ETL (extração, transformação e carregamento de dados), automatizando a coleta e escalonando o processo para manter a base de dados atualizada com os posts e comentários mais recentes da comunidade.

Ao concluir o desenvolvimento da ferramenta, um modelo LDA foi utilizado para gerar tópicos com base nos dados coletados, permitindo a exibição e análise dos dados gerados pela ferramenta em um contexto de análise textual. Isso proporciona uma visão aprofundada das tendências e informações relevantes presentes nas comunidades do Reddit, demonstrando o potencial de explorar essas valiosas fontes de dados textuais.

2. FUNDAMENTAÇÃO TEÓRICA

A seção de fundamentação teórica deste trabalho é dedicada a estabelecer as bases conceituais essenciais para a compreensão de elementos fundamentais ligados à área de processamento de dados e processamento de linguagem natural. Esses conceitos desempenham papéis cruciais no cenário atual de análise de dados, fornecendo as ferramentas necessárias para lidar com grandes volumes de informações, extrair conhecimento significativo e transformar dados textuais não estruturados em insights valiosos. Nesta seção, exploraremos as definições e princípios-chave desses conceitos, fornecendo assim uma base sólida para a compreensão das estratégias subsequentes de processamento de dados.

2.1 Big data

O termo "Big Data" tem se tornado cada vez mais proeminente no cenário atual de tecnologia e negócios. Ele se refere a conjuntos de dados extremamente grandes e complexos que superam a capacidade das ferramentas tradicionais de processamento de dados. O surgimento e a proliferação de Big Data estão intrinsecamente ligados à era da informação digital, onde uma quantidade massiva de dados é gerada a cada segundo, proveniente de fontes variadas, como redes sociais, dispositivos IoT (Internet das Coisas), registros de transações financeiras, entre outros.

A importância do Big Data reside na capacidade de aproveitar esses dados para obter insights valiosos. Empresas, governos, organizações de pesquisa e muitos outros setores estão usando análises de Big Data para tomar decisões mais informadas, identificar tendências, prever eventos futuros, personalizar produtos e serviços e até mesmo melhorar a eficiência operacional. Para lidar com o Big Data, são necessárias ferramentas e tecnologias específicas, como sistemas de armazenamento distribuído, bancos de dados NoSQL e técnicas avançadas de análise de dados, como aprendizado de máquina e processamento de linguagem natural.

2.2 Pipeline de dados ETL

O conceito "ETL" se relaciona com um procedimento essencial no âmbito da administração e análise de dados, que consiste nas etapas de Extração, Transformação e Carga de Dados em um sistema ou armazenamento adequado. Entre as ferramentas mais utilizadas atualmente, destaca-se o Apache Airflow [6], aqui conseguimos definir um pipeline de execução. Um pipeline de execução refere-se a um conjunto sequencial de processos ou etapas que são executados de maneira ordenada, em que a saída de uma etapa se torna a entrada da próxima. O processo ETL, cuja sigla em inglês corresponde a "Extract, Transform, Load", constitui uma metodologia organizada e eficiente para a adequação e combinação de dados provenientes de múltiplas origens, tornando-os aptos para serem utilizados em análises e aplicações subsequentes.

Extração (Extract), a primeira etapa do processo ETL envolve a coleta de dados. Os dados coletados podem vir de sistemas como bancos de dados, sistemas de arquivos, serviços web, sensores IoT, entre outros. A extração visa recuperar os dados brutos de suas fontes e movê-los para um local centralizado para processamento subsequente. A extração pode ser um processo complexo, especialmente quando os dados estão em diferentes formatos e locais geográficos.

Transformação (Transform), é o passo que sucede a extração. Os dados brutos frequentemente precisam ser limpos, enriquecidos e transformados para serem úteis para análises ou para alimentar sistemas de negócios. Essa etapa envolve a aplicação de regras de negócios, filtros, agregações, conversões de formatos e outras operações para garantir que os dados estejam coerentes e relevantes. A transformação é uma parte crítica do processo, pois garante a qualidade e a consistência dos dados.

A Carga (Load) é a etapa final do processo ETL. Esse passo envolve a persistência dos dados transformados em um local de destino, como um data warehouse, banco de dados ou aplicação de análise. Essa carga é executada de forma eficiente para garantir que os dados estejam disponíveis para consulta e análise. Dependendo da aplicação, os dados podem ser carregados periodicamente (carga em lote) ou em tempo real (carga em tempo real) para fornecer informações atualizadas.

A importância do Pipeline ETL reside na sua capacidade de preparar dados para análises significativas e para o suporte a decisões informadas. Ele ajuda a eliminar inconsistências, duplicações e erros nos dados, o que é crucial para tomadas de decisões precisas. Além disso, permite a integração de dados de diferentes fontes, possibilitando análises abrangentes e uma visão unificada do negócio. O Pipeline ETL é especialmente relevante em cenários de Big Data, onde a complexidade e a quantidade de dados são altas.

2.3 DAG

Um DAG (Directed Acyclic Graph) [7] no Apache Airflow é uma representação visual e programática de um fluxo de trabalho ou pipeline de tarefas. Ele consiste em nós interligados por arestas em uma estrutura direcionada e acíclica, o que significa que as tarefas têm uma ordem de execução e não há ciclos no fluxo. Cada nó em um DAG representa uma ação a ser executada, e as arestas indicam a ordem em que essas ações devem ocorrer. O Airflow usa DAGs para definir, agendar e monitorar fluxos de trabalho, permitindo a automação e orquestração de processos complexos.

2.4 Processamento de linguagem natural

O Processamento de Linguagem Natural (PLN) é um campo interdisciplinar da ciência da computação, inteligência artificial e linguística que se concentra na interação entre seres humanos e computadores por meio da linguagem natural. O objetivo principal do PLN é permitir que as máquinas compreendam, interpretem e gerem texto ou fala da mesma forma que os seres humanos, o que envolve a análise semântica, a extração de informações, a tradução automática e a geração de texto.

A área de processamento de linguagem natural desempenha um papel fundamental em uma variedade de aplicações modernas, desde motores de busca e assistentes pessoais até análise de dados em empresas e instituições de pesquisa. Com o avanço das

técnicas de aprendizado de máquina, como redes neurais profundas, a área tem feito progressos significativos, permitindo análises mais precisas e interações mais naturais entre humanos e máquinas. Para que essas aplicações desempenhem de forma ótima, é muito importante que o consumo de dados seja feito a partir de uma boa fonte de dados.

Uma boa fonte de dados desempenha um papel central e crítico na área de Processamento de Linguagem Natural (PLN) por várias razões fundamentais. A qualidade dos dados é essencial para a modelagem e treinamento precisos dos modelos de PLN, que incluem redes neurais e algoritmos de aprendizado de máquina. Dados de alta qualidade asseguram que esses modelos possam capturar com precisão nuances da linguagem, regras gramaticais e contextos.

2.5 Tokenização e stemming

A tokenização e o stemming são dois conceitos essenciais no campo do Processamento de Linguagem Natural (PLN) que desempenham papéis cruciais na análise e no tratamento de texto.

A tokenização é o processo de dividir um texto em unidades menores, chamadas de tokens, que podem ser palavras, frases, sílabas ou caracteres, dependendo da granularidade desejada. Essa etapa é fundamental em tarefas de Processamento de Linguagem Natural (PLN), permitindo a análise e processamento do texto. A tokenização é crucial para a Contagem de Palavras, Análise de Frequência e criação de N-gramas. Ela é essencial para a análise estatística, modelagem de linguagem e aprendizado de máquina, proporcionando as bases para uma análise mais profunda e significativa do texto.

O stemming é um processo que envolve a redução de palavras à sua forma raiz, chamada de "stem". O objetivo do stemming é eliminar a variação morfológica das palavras, reduzindo-as a uma forma comum. Por exemplo, as palavras "correr", "correu" e "correndo" seriam reduzidas ao mesmo stem "corr". A etapa de stem pode ser muito importante para o processo, pois traz características ao conjunto de dados, como: redução de dimensionalidade, melhor recuperação de informações e agrupamento de palavras semelhantes. No entanto, é importante observar que o stemming não é perfeito e pode gerar resultados imprecisos em alguns casos. Por exemplo, a palavra "melhor" e "melhorar" teriam o mesmo stem "melhor," embora tenham significados diferentes.

3. METODOLOGIA

A aplicação que foi desenvolvida tem o objetivo de ser um coletor escalonado de dados textuais de subreddits, acompanhado por módulos de processamento e carga de dados, além de uma API com uma função de LDA (Latent Dirichlet Allocation) [15]. Nesta seção, será detalhado o caminho metodológico adotado desde o planejamento até a implementação prática dessa ferramenta ETL (Extração, Transformação e Carga). O principal objetivo desse processo é capturar, transformar e carregar os dados textuais provenientes dos subreddits de forma a torná-los acessíveis para análises subsequentes.

O primeiro passo do projeto foi o planejamento, onde foram definidas as etapas necessárias para atingir nosso objetivo central, que é criar uma ferramenta robusta e flexível, capaz de lidar com a complexidade dos dados textuais online. O planejamento envolveu a definição de fontes de dados, a escolha de técnicas de extração, transformação e carga. A ferramenta desenvolvida realiza a extração de dados dos subreddits selecionados de forma escalonada e programada. Aqui é utilizado o acesso a API do Reddit [17].

Através dessa API coletamos o texto de posts e informações relevantes da comunidade alvo. Isso nos permite obter um conjunto diversificado de dados textuais para análise.

Após a coleta, os dados brutos passam por um processo de transformação. Utilizamos técnicas de pré-processamento de texto, como tokenização, remoção de stopwords e stemming, para normalizar o conteúdo textual.

Uma vez que os dados foram transformados, eles são carregados em um repositório de dados. A escolha da tecnologia de armazenamento foi feita com base na escalabilidade da ferramenta. Por isso, o esquema de armazenamento dos dados segue o padrão NoSQL. Dessa forma os dados são organizados de maneira a facilitar o acesso e a consulta, permitindo análises posteriores.

Além da coleta, transformação e carga de dados, nossa aplicação disponibiliza uma API com uma função de LDA. Essa função utiliza o algoritmo Latent Dirichlet Allocation para identificar tópicos nos dados textuais. Os usuários podem enviar consultas à API para obter informações sobre os tópicos presentes nos subreddits analisados.

Em resumo, a metodologia seguida para a criação desta ferramenta ETL, abrangeu o planejamento detalhado, a extração de dados, a transformação para normalização e enriquecimento, a carga eficaz dos dados e a disponibilização de funcionalidades analíticas por meio da API. O resultado é uma ferramenta versátil e poderosa para lidar com dados textuais online e extrair informações valiosas a partir deles.

4. VISÃO GERAL DA FERRAMENTA

A ferramenta desenvolvida é responsável por executar todos os passos necessários para prover uma base de dados atualizada. Todos os passos são cumpridos de forma contínua, sendo executados a cada hora. A arquitetura da ferramenta é baseada em microserviços. Os serviços em questão se dividem entre API para auxílio na coleta e processamento dos dados, API de banco de dados, módulo responsável pelo escalonamento das tarefas e módulo de LDA. A seguir, na Figura 1, podemos observar o diagrama que representa o fluxo central de execução da ferramenta, extração, transformação e carga dos dados.

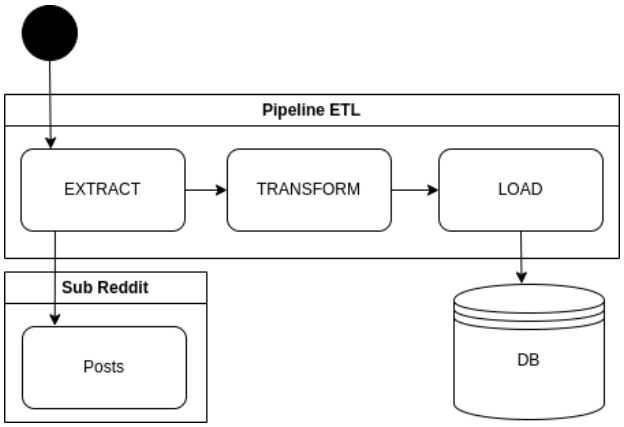

Figura 1. Fluxo de execução da ferramenta.

4.1 Escalonador de tasks

O Apache Airflow é uma plataforma de gerenciamento de fluxo de trabalho que desempenha um papel crucial como um escalonador em pipelines ETL (Extract, Transform, Load). Em um cenário onde precisamos coletar, adequar e persistir dados, o Airflow pode ser configurado para orquestrar esse processo através de um arquivo DAG.

A DAG no contexto descrito desempenha o papel de escalonar as seguintes tarefas do pipeline: Coleta de Dados, Adequação e Persistência. Todos os passos descritos funcionam com base em chamadas de APIs externas ao módulo do Airflow. A seguir observamos uma versão mais detalhada do diagrama de fluxo de execução, onde vemos a relação entre a DAG e as APIs envolvidas no processo.

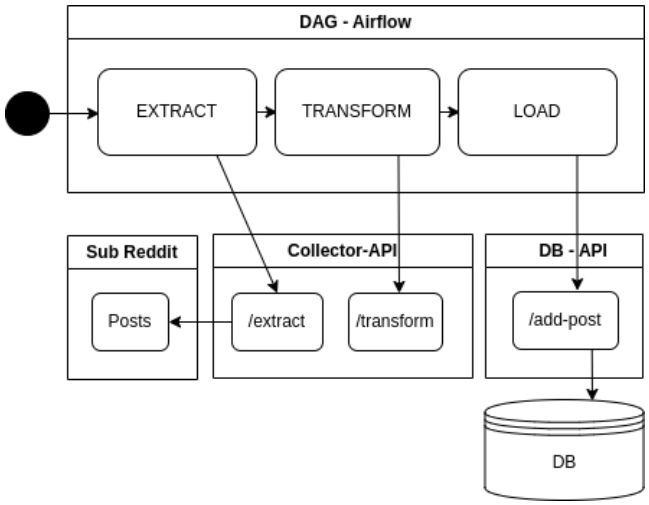

Figura 2. Fluxo de execução, detalhado a partir da DAG e as chamadas feitas a APIs externas.

O Airflow opera em um ambiente de container Docker [5] para garantir a portabilidade e facilidade de implantação. A DAG definida no Airflow especifica a ordem das tarefas a serem executadas, suas dependências e os parâmetros necessários para cada uma delas. No arquivo que define a DAG, também podemos encontrar as configurações relacionadas ao escalonamento do pipeline. Os passos do pipeline foram configurados para serem executados a cada hora. O código referente ao escalonador pode ser encontrado junto a organização no repositório de nome: collector-airflow [1].

4.2 Módulo de processamento

O módulo de processamento é a API responsável por desempenhar as funções de coleta e processamento dos dados textuais originados do Reddit. Aqui conseguimos extrair e tratar os dados da rede social, através de chamadas de API.

A arquitetura da API se baseia no padrão REST e foi toda desenvolvida utilizando Python, Flask [10] e Redis [9] como banco de dados. O Redis foi escolhido a fim de otimizar o processamento compartilhado entre as rotas da API. Já que o banco de dados possibilita acesso e carga mais eficientes, pois suas operações são salvas em memória. Para preparo do ambiente da aplicação, é feita a conteinerização da aplicação com o uso de Docker.

O módulo possui duas rotas principais, sendo elas "/extract" e "/transform". A seguir, podemos ver o diagrama que descreve o relacionamento entre as rotas e os recursos acessados no banco de dados.

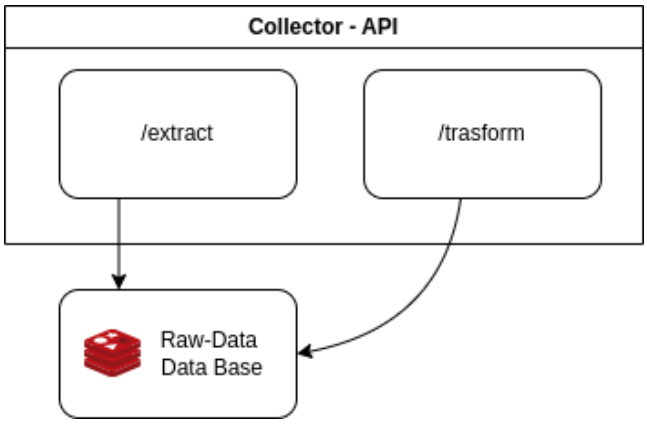

Figura 3. Diagrama que descreve o relacionamento entre as rotas e o banco de dados.

O código fonte referente ao módulo de processamento dos dados extraídos pode ser encontrado na organização que centraliza as aplicações da ferramenta. O repositório em questão se chama: collector-api [2].

4.2.1 Rota "/extract"

Na rota de extração ("/extract"), os comentários do Subreddit são consumidos através da API da plataforma. Após isso, o lote coletado é carregado na memória sem sofrer nenhum tipo de tratamento. Todos os conjuntos de dados carregados em banco, são identificados pelo momento em que foram extraídos.

Ao fim da extração, caso os dados sejam persistidos com sucesso, a rota nos retorna a chave identificadora para o lote de dados extraídos.

4.2.2 Rota "/transform"

A partir da extração dos dados, somos capazes de adequar qualquer lote carregado na memória. Para isso, devemos recorrer a rota "/transform". Aqui, acessamos o lote guardado em memória através de sua chave identificadora, a fim de adequá-los de forma direcionada.

Com os dados brutos em mãos, neles são aplicadas técnicas de processamento de texto. Nesse passo, são utilizados mecanismos como stemming e a remoção de stopwords (palavras com baixo índice de significância para o nosso conjunto), além de também ser feita a limpeza de caracteres especiais que podem ser encontrados nos comentários da plataforma. Aqui utilizamos o Porter Stemmer [14] e o conjunto de stopwords do inglês provido pela biblioteca NLTK [11]. Através do uso dessas técnicas conseguimos diminuir o espaço amostral do conjunto de dados e fornecer um montante renovado, com mais valor para análises posteriores.

Quando finalizado o processo de transformação desses dados, a rota nos retorna a representação dos dados em formato Json. Os elementos de cada post são representados em forma de lista, ou seja, o nésimo elemento das listas representa o nésimo post coletado do Reddit. Cada post possui, comentário, score

(pontuação da plataforma), corpo (self_text) e título relacionados. Um exemplo da estrutura pode ser visto na Figura abaixo.


```
"posts": {
    "comment": [
      "mean went knew time strong feel",
      "fetterman blind support war crime",
    ],
    "score": [
      1,
      1,
    ],
    "self_text": [
      "",
      ""
    ],
    "title": [
      "Listen: Billionaire Brags About Secret Information With Him",
      "Celeste Maloy is Ammon Bundy's cousin. Would that impact"
    ]
  }
```


Figura 4. Estrutura de retorno de dados Tratados pela rota "/transform".

4.3 Módulo de persistência

O módulo de persistência existe para atender à necessidade de comunicação com o banco de dados final. Aqui é onde temos acesso a operações básicas para a gestão dos nossos dados. A API oferece uma espécie de CRUD, permitindo o armazenamento de posts do Reddit como documentos em nosso banco de dados.

Toda a API foi desenvolvida utilizando Python, Flask e MongoDB [8]. Para a execução e preparação do ambiente da aplicação, foi feito o uso de container Docker, tanto para a API quanto para o MongoDB.

A escolha de um banco de dados NoSQL foi feita com base nos fatores mais relevantes relacionados à aplicação. Considerando que o contexto da ferramenta está associado ao Big Data, é ideal que tenhamos um banco de dados que ofereça altos níveis de desempenho, escalabilidade e flexibilidade. Levando em consideração todos esses pontos, optamos pelo MongoDB como nosso banco de dados final. O MongoDB oferece um uso simplificado, além de proporcionar a opção de utilização do Atlas (um produto da MongoDB que disponibiliza o banco de dados na nuvem).

Como mencionado anteriormente, a API fornece um conjunto de operações CRUD para interagir com o banco de dados. Entre todas as rotas disponíveis, a fundamental para o funcionamento do processo é a rota 'add-post'. Através dela, é possível efetuar o cadastro de um post em nosso banco de dados.

Junto à organização principal da ferramenta, podemos encontrar o código fonte do módulo (db-api [3]).

4.4 Módulo de análise com LDA

Com base na ferramenta descrita anteriormente, partimos para o módulo de código responsável por prover a análise com LDA (Latent Dirichlet allocation). No seu desenvolvimento, foi utilizado Python, NLTK, Scikit-learn [13], Flask e Docker.

A partir dessa aplicação conseguimos mapear tópicos de acordo com a base de dados fornecida pelo núcleo de coleta da ferramenta. Além disso, o módulo fornece uma forma de otimização básica, feita através do grid search [16]. A otimização em questão, trabalha com base no teste de combinações entre hiperparâmetros fornecidos para o modelo. No nosso caso, o modelo usado será o LDA e o hiperparâmetro escolhido é o número de tópicos usado. O conjunto de valores escolhidos para concorrer como hiperparâmetro pode ser definido no código fonte da aplicação, em forma de lista.

A rota principal do módulo recebe um json que define dois aspectos da execução do modelo. Um deles é uma flag que sinaliza o uso da otimização via grid search, o outro sinaliza a quantidade de tópicos usada no LDA, no caso de não usarmos a otimização.

Através do módulo, conseguimos extrair tópicos dos dados coletados anteriormente. Junto dos tópicos, também recebemos "n" comentários com maior afinidade ao tópico e "m" palavras acompanhadas de sua métrica de frequência dentro do tópico. A quantidade de comentários e palavras retornadas, pode ser escolhida dentro do código fonte do módulo. Com isso, temos uma saída que segue a seguinte estrutura.


```
{
  "topics": [
    {
      "id": 1,
      "top_documents": [
        [
          "law sb ban tran kid use school facil align gender ident offer children bounti report
          tran student use restroom match gender ident transphob legisl defin sex immut biolog
          characterist birth wholli ignor exist intersex children sex trait entir male femal
          legisl also state mere presenc tran student might inflict psycholog injuri cisgend
          peer increas likelihood sexual assault molest rape tran independ journalist erin reed
          point year transgend student abl use restroom consist gender school across idaho year
          without incid thi order allow inclus practic continu pursu challeng lambda legal staff
          attorney kell olson told windi citi time x b r pastorarrest beg differ x b assault
          bathroom republican tran peopl assault pastor tran peopl long shot x b transgend peopl
          like assault bathroom cannot use correct bathroom bathroom match gender ident x b
          scullion call christian nationalist delight inflict pain upon tran peopl like get
          jolli know tran peopl cisgend peopl also hate bigot polici x b republican would rather
          dead peopl tran peopl sad statement regard republican differ westboro baptist church",
          "Court blocks Idaho's deeply transphobic student bathroom bill. The law offers a
          bounty on trans bathroom users and paints them as would-be rapists.",
          "0.9961478975536096"
        ]
      ],
      "top_words": [
        [
          "thi",
          519
        ],
        [
          "ha",
          188
        ],
        [
          "israel",
          139
        ],
        [
          "republican",
          223
        ]
      ]
    },
    ]
```


Figura 5. Estrutura de retorno de módulo LDA.

Considerando, ainda, um cenário em que nossa análise não envolva o uso de LDA, podemos utilizar os dados previamente armazenados. Uma vez que temos acesso aos dados tratados, podemos direcionar nosso foco para outros processos, já que, com essas informações, somos capazes de gerar diversos tipos de análises. O código fonte relacionado ao módulo de análise também pode ser encontrado na organização da ferramenta, ele está armazenado no repositório analyzer-api [4].

5. RESULTADOS

Neste trabalho, exploramos os dados coletados diretamente do subreddit '/politics', ao longo de um período de dois dias. A comunidade utilizada aborda temas relacionados à política estadunidense. Utilizamos o módulo de LDA (Latent Dirichlet Allocation) descrito anteriormente para identificar e entender os principais tópicos de discussão nessa comunidade. Para fim de testes, consideramos apenas dois tópicos como configuração do módulo de LDA. A partir desses tópicos gerados, analisamos a frequência de palavras dentro desses tópicos. Isso foi feito através da observação crítica dos valores gerados com o uso do módulo. É importante considerar que os dados coletados apresentam um viés,

considerando que em eventos atuais, como conflitos mundiais, os tópicos são amplamente discutidos na esfera social analisada, resultando em uma grande quantidade de comentários e postagens relacionados a esses assuntos. Desta forma, hipotetizamos que as semelhanças entre tópicos, apresentada nas próximas seções, é resultado desse viés de popularidade de tema.

5.1 Frequência dos Termos

Primeiramente, examinamos a frequência dos termos mais relevantes em cada tópico identificado pelo módulo de LDA. Através das palavras geradas pelo módulo, foram criados dois gráficos de frequência, um para cada tópico configurado. Nas Figuras 6 e 7 são destacadas as vinte e cinco palavras com mais afinidade no tópico 1 e 2 respectivamente. Isso nos ajuda a compreender quais termos são mais discutidos e assim identificar os principais temas e preocupações do subreddit alvo.

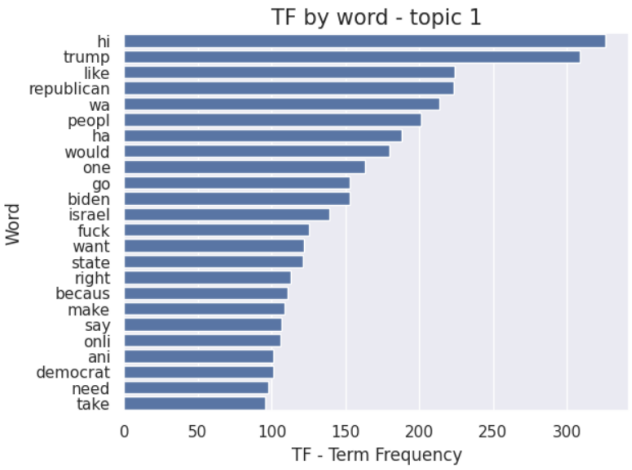

Figura 6. Gráfico que descreve frequência de palavras mais relacionadas ao tópico 1.

No primeiro tópico podemos perceber que entre as vinte e cinco palavras mais relacionadas ao tópico, temos termos como: Trump, Biden, e Israel. Isso indica que o tópico capta comentários ligados a discussões envolvendo os conflitos atuais que envolvem Israel, além de também mostrar que muito está sendo falado sobre o atual e ex-presidente do País.

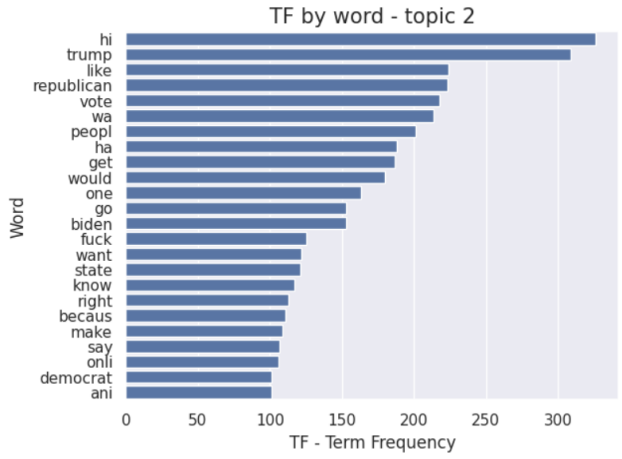

Figura 7. Gráfico que descreve frequência de palavras mais relacionadas ao tópico 2.

Por outro lado, no segundo tópico podemos observar que Israel saiu dos vinte e cinco termos mais frequentes no tópico. Dessa vez, encontramos termos como: Trump, Biden, Republican, Democrat, Vote. Isso indica que o tópico inclui comentários ligados a discussões envolvendo as duas principais frentes políticas dos Estados Unidos, processos eleitorais e presidentes dos Estados Unidos.

5.2 Comentários mais Relevantes

Esta análise visual busca destacar termos relvantes aos tópicos por meio da biblioteca word cloud [12] do python, que gera nuvem de palavras e termos. As visualizações foram geradas a partir dos cinco comentários mais relevantes em cada tópico e destacam visualmente as palavras-chave associadas a cada um deles. Na Figura 8 vemos a nuvem formada para nos ajudar a identificar as palavras mais representativas e significativas em relação aos tópicos específicos.

Figura 8. Nuvem de palavras montada através dos cinco comentários mais relevantes ao tópico 1.

Nesse conjunto de comentários podemos observar palavras como: Israel, Ukraine, Biden e concern. Com base nessas palavras é possível deduzir que os comentários têm foco nos assuntos relacionados aos eventos atuais, ligados às guerras e ao atual presidente dos Estados Unidos. Na Figura 9 vemos qual foi a organização em relação às palavras do segundo tópico.

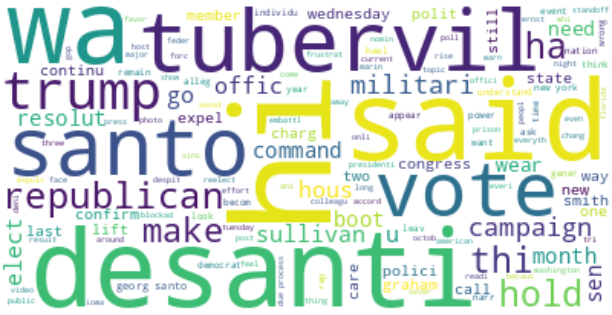

Figura 9. Nuvem de palavras montada através dos cinco comentários mais relevantes ao tópico 2.

Podemos ver que entre as palavras mais vistas nos comentários do tópico dois, estão algumas como: Trump, said, republican e vote. Isso pode apontar que esses comentários estão ligados a assuntos como o partido Republicano e o ex-presidente dos Estados Unidos.

6. CONCLUSÃO

Por meio do desenvolvimento da ferramenta e das análises apresentadas, podemos obter uma compreensão mais profunda das principais discussões e tópicos em destaque no subreddit '/politics'

durante o tempo de coleta de dados. A combinação de frequência de termos, comentários relevantes e nuvens de palavras nos permite identificar tendências, temas quentes e questões de interesse na comunidade política. Além dessa comunidade, também podemos expandir a análise em relação a outro nicho de assuntos, tornando assim a ferramenta útil em relação à análise de comportamento online em comunidades do Reddit.

A pesquisa e o desenvolvimento desta ferramenta representam valor significativo no campo do processamento de linguagem natural e ETL. O Reddit é uma plataforma rica em dados, com uma enorme quantidade de informações e opiniões compartilhadas por milhões de usuários em todo o mundo. Esta ferramenta oferece uma solução eficaz para coletar, processar e disponibilizar dados de forma eficiente e escalável.

O impacto dessa pesquisa não se limita apenas ao contexto acadêmico. Empresas e organizações também podem se beneficiar significativamente da aplicação desta ferramenta para diversos processos relacionados aos dados de plataformas semelhantes ao reddit. .

7. AGRADECIMENTOS

Gostaria de expressar meus mais profundos agradecimentos a Deus, que tem sido minha força e guia ao longo de toda a jornada acadêmica. Agradeço de coração aos meus pais e irmão, cujo amor, apoio incondicional e sacrifícios tornaram possível minha educação. Aos meus amigos e companheiros de curso, que compartilharam comigo desafios e vitórias, meu mais sincero agradecimento. Em particular, gostaria de destacar Daniel Carlos, que desempenhou um papel fundamental antes e durante minha formação na UFCG e Henry, um grande amigo que o curso nos presenteou e que foi essencial no início da minha trajetória acadêmica. Agradeço imensamente ao meu orientador, Fábio Jorge, pela orientação, sabedoria e apoio ao longo deste processo. A todos os professores que contribuíram para minha formação, acredito que cada um de vocês desempenhou um papel vital em minha jornada acadêmica. Estou profundamente grato a todos por fazerem parte dessa conquista.

8. REFERÊNCIAS

[1] collector-airflow. https://github.com/RTT-app/collector-airflow.

[2] collector-api. https://github.com/RTT-app/collector-api.

[3] db-api. https://github.com/RTT-app/db-api.

[4] analyzer-api. https://github.com/RTT-app/analyzer-api**.**

[5] Docker. https://docs.docker.com/.

[6] Airflow Documentation. https://airflow.apache.org/docs/.

[7] DAGs. https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/dags.html.

[8] MongoDB. https://www.mongodb.com/docs/.

[9] Redis. https://redis.io/docs/.

[10] Flask. https://flask.palletsprojects.com/en/3.0.x/.

[11] NLTK. https://www.nltk.org/.

[12] Word Cloud. https://github.com/amueller/word_cloud.

[13] ScikitLearn. https://scikit-learn.org/0.21/documentation.html.

[14] Willett, P. (2006). The Porter stemming algorithm: then and now. STEMM

[15] Mohamed Bakrey (2023), All about Latent Dirichlet Allocation (LDA) in NLP. https://mohamedbakrey094.medium.com/all-about-latent-dirichlet-allocation-lda-in-nlp-6cfa7825034e

[16] Rohan Josep (2018), Grid Search for model tuning. hhttps://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e

[17] Reddit API. https://www.reddit.com/dev/api/