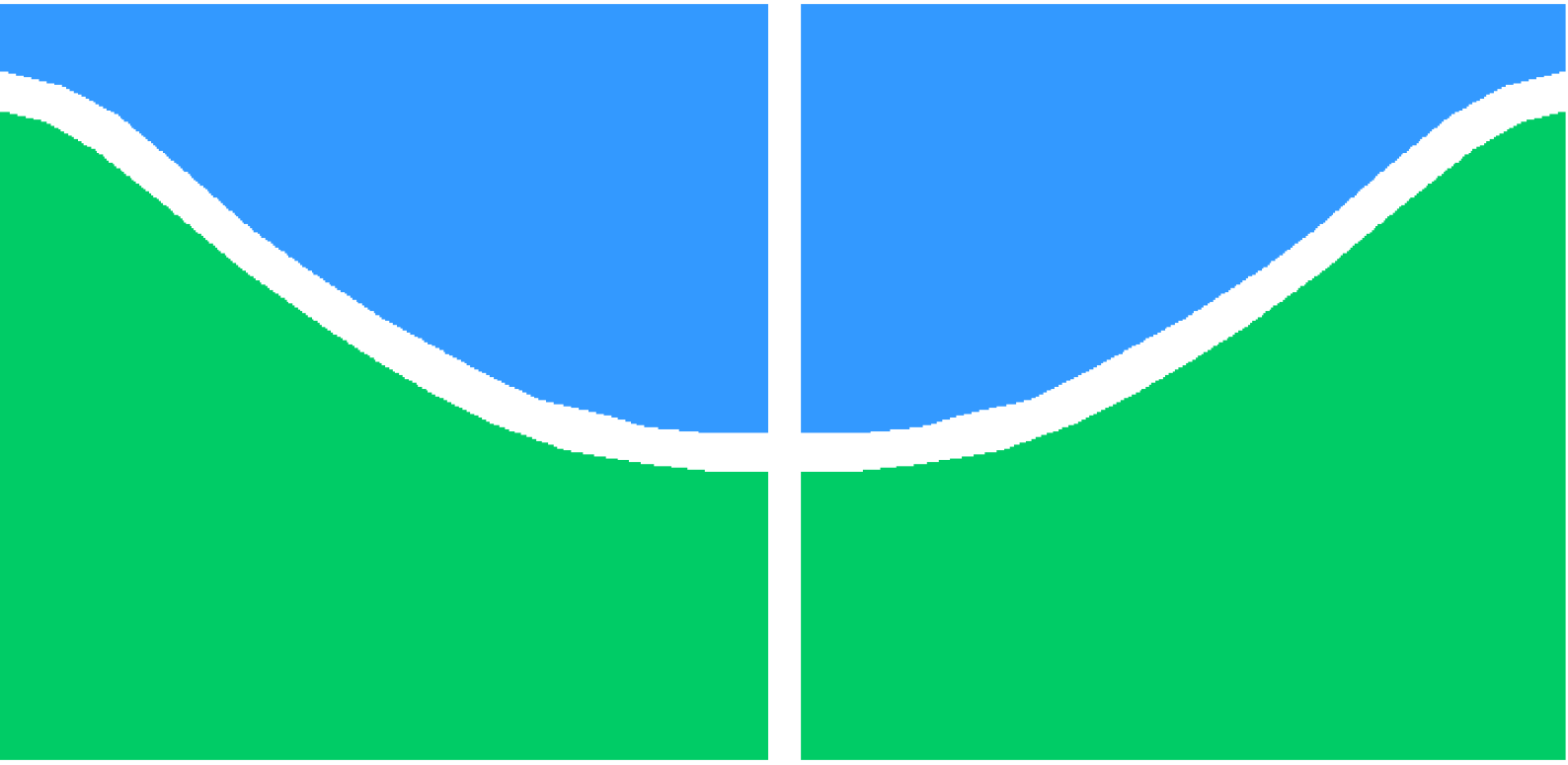


Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Engenharia de Software

Construção de uma *pipeline* de aprendizagem ativa para modelos de reconhecimento de entidades nomeadas

Autor: Matheus Gabriel Alves Rodrigues

Orientador: Prof. Dr. Fabricio Ataíde Braz

Brasília, DF

2023

Matheus Gabriel Alves Rodrigues

Construção de uma *pipeline* de aprendizagem ativa para modelos de reconhecimento de entidades nomeadas

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Fabricio Ataíde Braz

Brasília, DF

2023

Matheus Gabriel Alves Rodrigues

Construção de uma *pipeline* de aprendizagem ativa para modelos de reconhecimento de entidades nomeadas

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 24 de setembro de 2023:

Prof. Dr. Fabricio Ataíde Braz
Orientador

Dr. Nilton Correia da Silva
Convidado 1

Dr. Henrique Marra Taira Menegaz
Convidado 2

Brasília, DF
2023

Agradecimentos

Gostaria de agradecer a minha familia por ter me apoiado em todos os momentos de minha vida. Agradeço a minha namorada Shakira por corrigir e opinar sobre meu trabalho e mais importante que isso por me incentivar e apoiar todos os dias,

Sou grato também aos meus colegas de graduação que sempre me ajudaram durante todo o curso, em especial para o Roberto Martins, Pedro Henrique e Jonathan Jorge pelo apoio nesse trabalho.

"Um sapo em um poço não conhece o grande oceano."
(provérbio asiático)

Resumo

Com a crescente demanda por aplicações que fazem o uso de modelos de inteligência artificial, o custo envolvido na construção de um modelo tem sido uma preocupação constante, dentre as etapas que mais encarecem um modelo está a de rotulação dos dados que serão utilizados para treinamento. Uma das soluções existentes para este problema é o uso de uma abordagem conhecida como aprendizagem ativa, que consiste em permitir que durante o período de treinamento de um ou mais modelos, ocorra uma interação dos mesmo com um humano especialista no assunto dos dados, de modo que este consiga realizar validações em apenas alguns dados previamente selecionados.

O objetivo deste trabalho é desenvolver uma *pipeline* de *active learning* focada em modelos que fazem a atividade de reconhecimento de entidades nomeadas. Foram coletados dados abertos que permitiram a criação de um fluxo por completo, também foi selecionada uma ferramenta de rotulação que possibilita a interação dos oráculos com os dados selecionados a cada etapa do fluxo.

A *pipeline* elaborada foi experimentada para a construção de um modelo de classificação binária de dados textuais, isso possibilitou verificar que o fluxo funciona corretamente e levantar possíveis melhorias para serem elaboradas no decorrer da continuação dessa pesquisa.

Palavras-chave:

Aprendizagem ativa, Aprendizado de maquina, Reconhecimento de entidade nomeada, NER, NLP, Processamento de linguagem natural, Inteligência artificial

Abstract

The increasing demand for applications that use artificial intelligence models has made the cost involved in building a model a constant concern. One of the stages that most increase the cost of a model is the labeling of the data to be used for training. One of the solutions to this problem is the use of an approach known as active learning, which consists of allowing the models to interact with a human expert in the subject of the data during the training period, so that the expert can validate data in only a few previously selected data.

The objective of this work is to develop an active learning pipeline focused on models that perform named entity recognition. Open data was collected to create a complete flow, and a labeling tool was selected that allows the oracles to interact with the selected data at each stage of the flow.

The pipeline developed was tested for building a binary classification model of textual data, which allowed us to verify that the flow works correctly and to raise possible improvements to be developed in the continuation of this research.

Key-words:

Active learning, Machine learning, Named entity recognition, NER, NLP, Natural language processing, Artificial intelligence

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

UnB	Universidade de Brasília
IA	Inteligência Artificial
NER	Reconhecimento de Entidade Nomeada
NLP	Processamento de Linguagem Natural
LSTM	Long Short Term Memory
BiLSTM	Bidirectional Long Short Term Memory

Sumário

1 Introdução

1.1 Contexto

Com a crescente demanda por modelos de inteligência artificial nas mais diversas áreas do nosso cotidiano, como em aplicações de mobilidade urbana, sites de entretenimento, entre outras, a demanda por dados que possam ser utilizados para treinar esses modelos também cresce, desse modo, garantir a qualidade dos dados que são utilizados para realizar o treinamento desses modelos se torna uma tarefa muito importante para empresas de diferentes áreas.

Em modelos supervisionados, onde os rótulos atribuídos a um determinado dado de treinamento são responsáveis pelas inferências que os modelos irão fazer, realizar uma etapa de rotulação com qualidade é muito importante. Esse processo ocorre de maneira com que pessoas que possuam familiaridade com o conteúdo e o contexto dos dados atribuam os rótulos para os mesmos. Esta etapa costuma ser onerosa por manter pessoas realizando trabalhos repetitivos e muitas vezes manuais.

1.2 Problema

Em modelos de Reconhecimento de Entidades Nomeadas (NER) onde a rotulação dos dados ocorre de maneira com que os dados rotulados são *tokens* presentes em um ou mais textos, é comum a ocorrência de problemas de integridade dos rótulos, tendo em vista que a etapa de rotulação dos dados é realizada manualmente e também pelo fato do contexto da ocorrência da palavra poder influenciar a classificação da mesma.

Desta forma, se faz interessante que a etapa de rotulação dos dados seja completa o mais rápido possível e com a menor quantidade de recursos aplicados possível, de maneira que a criação dos modelos de IA seja mais veloz e assim agregue valor para o cliente final de maneira mais rápida.

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo desse trabalho consiste em desenvolver um fluxo de dados que utilize aprendizagem ativa na construção de modelos que realizam a tarefa de reconhecimento de entidades nomeadas.

1.3.2 Objetivos específicos

Para alcançar o objetivo geral do estudo, foram elicitados outros objetivos específicos que são metas a serem alcançadas para se obter o resultado esperado no objetivo específico. (OBJETIVOS. . . ,)

Esses objetivos são:

- Selecionar uma ferramenta de rotulação de dados que possa ser utilizada no fluxo.
- Desenvolver uma *pipeline* de aprendizagem ativa para modelos de classificação textual.
- Desenvolver o fluxo de *active learning* para modelos de reconhecimento de entidades nomeadas.
- Aplicar a *pipeline* desenvolvida em um conjunto de dados.

1.4 Organização do Trabalho

Este trabalho de conclusão de curso está estruturado em vários capítulos, cada um abordando uma etapa específica do projeto. A seguir, será fornecida uma visão geral de cada capítulo, destacando os principais tópicos abordados.

- **Capítulo 1 - Introdução:**

  Neste capítulo, é apresentado o contexto do trabalho, incluindo uma breve descrição da problemática com rotulação de dados. Também são delineados os objetivos do trabalho.

- **Capítulo 2 - Referencial teórico:**

  Neste capítulo, são descritos os conceitos teóricos que fundamentam o trabalho. São apresentados estudos, teorias e pesquisas relevantes sobre aprendizagem ativa, arquitetura de rede neural que será utilizada no projeto, etc, fornecendo um embasamento teórico para as etapas subsequentes.

- **Capítulo 3 - Materiais e Métodos:**

  Este capítulo detalha as etapas que foram realizadas para se chegar ao fluxo de *Active learning*, também são explicados os objetivos que foram levantados para serem alcançados ao fim de cada etapa.

- **Capítulo 4 - Resultados e discussão:**

Neste capítulo, são apresentados os resultados obtidos ao realizar as etapas mencionadas anteriormente. Também são demonstrados como foram realizadas essas etapas, demonstrando a metodologia utilizada e análises sobre os objetivos obtidos.

- **Capítulo 5 - Conclusão**

  No último capítulo, são apresentadas as conclusões gerais do trabalho. Com base nos resultados e nas discussões realizadas, são tiradas inferências e resumidos os principais pontos abordados ao longo do trabalho. Também foram levantados possíveis melhorias para trabalhos futuros de modo que o fluxo de aprendizagem ativa obtido neste trabalho possa ser melhorado por outros pesquisadores.

2 Referencial Teórico

2.1 Considerações Iniciais

Para se abordar o tema específico desse estudo, anteriormente se faz necessário, realizar uma pesquisa sobre os tópicos que compõem o mesmo e são importantes para dar base ao tema proposto.

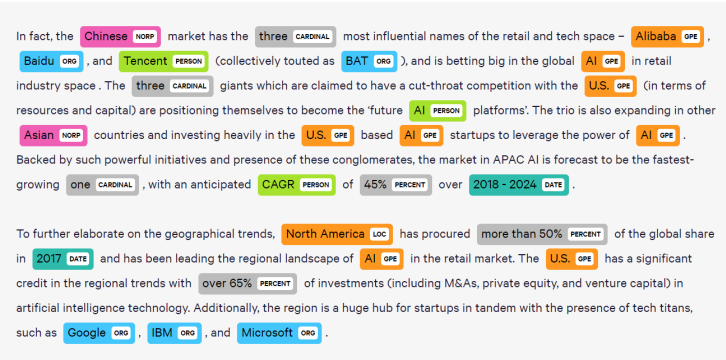
2.2 Reconhecimento de entidades nomeadas

O Reconhecimento de entidades nomeadas (NER) consiste em um ramo da área de processamento textual que tem como objetivo utilizar modelos de inteligência artificial para identificar e classificar entidades presentes em um determinado texto. Essas entidades consistem em substantivos próprios que são selecionados como categorias pré-definidas e variam de acordo com o contexto dos dados em questão (JURAFSKY; MARTIN, 2020). Alguns exemplos de possíveis entidades são:

- Nome de pessoas;
- Lugares;
- Empresas;
- Leis;
- Datas.

A Figura 1 apresenta um exemplo de uma aplicação que utiliza reconhecimento de entidades nomeadas.

Figura 1 – Exemplo de aplicação de NER



Fonte: (NAMED. . . ,)

Esse tipo modelo de aprendizagem de máquina pode ser utilizado para diferentes tipos de atividades em nosso cotidiano. Algumas dessas atividades são:

- *Chatbots*;
- Ter conhecimento de onde uma entidade está sendo informada em uma determinada frase;
- Categorização de textos de acordo com seu conteúdo.

Outra aplicação útil dessa técnica é que com ela pode-se retirar do texto em questão termos que não agregam para a análise desejada, diminuindo assim a chance de que um modelo aprenda que estas entidades sejam relevantes para o estudo e também reduz a quantidade de termos que existirão no vocabulário do modelo economizando assim memória (MURTHY; KHAPRA; BHATTACHARYYA, 2018).

Um exemplo é:

Rodolfo estuda engenharia de *software* na **Universidade de Brasília**

Nessa frase os termos grifados não acrescentam a classificação geral do enunciado, desse modo utilizar um modelo de NER para substituir esses termos por entidades que sejam genéricas e apresentem um significado igual para o contexto da frase pode ser um processamento interessante, modificando o exemplo em questão para algo parecido com:

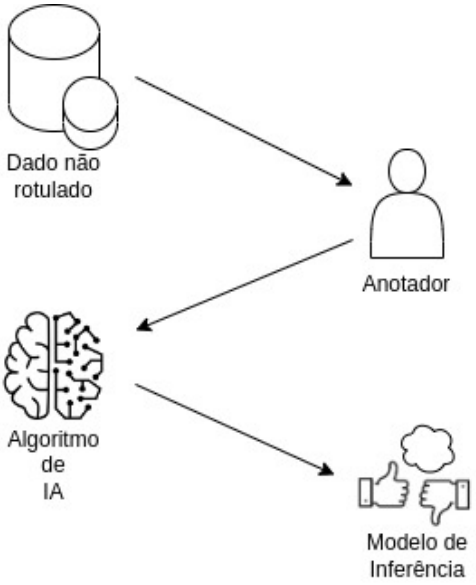
PESSOA estuda engenharia de *software* na **INSTITUIÇÃO**

2.3 *Active Learning*

Para realizar o treinamento e validação de modelos supervisionados se faz importante a coleta e rotulação dos dados que serão utilizados nesse processo, porém a etapa de rotulação por muitas vezes se mostra como sendo custosa e lenta, dessa maneira atrasando a geração de valor para o cliente final do modelo em questão.

O ciclo de vida habitual de um modelo de inteligência artificial supervisionado consiste em utilizar dados previamente rotulados por especialistas no assunto do qual o modelo se refere para treinar um modelo que posteriormente depois de validações será utilizado para gerar inferências sobre novos dados. Esse fluxo é também chamado de ***Passive Learning*** (SETTLES, 2009) e o mesmo é exemplificado na Figura 2.

Figura 2 – Fluxo de aprendizagem passiva



Fonte: autoral

Um diferente fluxo de modelagem é proposto pela aprendizagem ativa, sendo ele descrito assim:

> ... o fluxo esperado é ter inicialmente um Pool com um grande ou crescente número de instâncias e o *Labelled* com poucas ou nenhuma instância. Determinado um método para selecionar (ou gerar) instâncias para inquerir ao oráculo, cada iteração do algoritmo utiliza a *Pool* para montar o *Selected*, que após classificados são removidos de *Selected* e adicionados ao *Labelled*. (CORTI, 2021)

O trecho acima trás quatro novos termos utilizados em fluxos de aprendizagem ativa, sendo eles:

- **Oráculo**

  Refere-se a um especialista que será consultado durante as rodadas do fluxo da aprendizagem ativa. Este será responsável pela validação e rotulação de amostras que estejam no *Selected*.

- ***Pool***

  Conjunto total dos dados do estudo, inclui dados rotulados e não rotulados.

- ***Labelled***

  Consiste em um conjunto de dados que já possui rótulo, está incluso dentro do *Pool*.

- ***Selected***

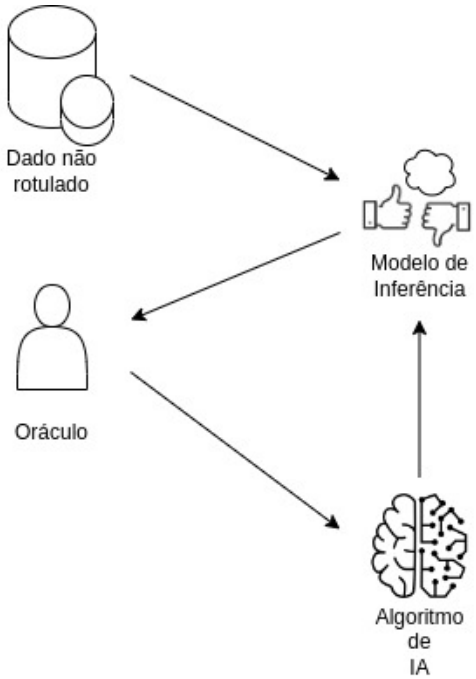
  Se trata dos dados que foram selecionados baseando-se nas respectivas heurísticas escolhidas para o estudo, esses dados são separados para que sejam validados pelo *Oráculo.*

O fluxo de aprendizagem ativa consiste que um modelo treinado previamente com uma menor quantidade de dados, classifique dados do ***Pool***, utilize alguma heurística para selecionar dados para compor o ***Selected*** e utilize o mesmo para realizar consultas ao **oráculo**, que por sua vez será responsável por validar essas consultas realizando uma rotulação nos dados oferecidos, com base nisso uma nova rodada é executada, onde o modelo recebe um novo treinamento com o ***Labelled*** sendo incrementado com os novos rótulos. (SETTLES, 2009)

Quando um novo modelo é treinado no início de uma rodada é importante realizar uma validação do mesmo para verificar se sua performance está aceitável para uma nova rodada ou até mesmo para se tornar uma versão final do modelo.

O fluxo é exemplificado na Figura 3:

Figura 3 – Fluxo de aprendizagem ativa



Fonte: autoral

Nota-se que, quando se utiliza um fluxo de aprendizagem ativa é necessário que se passe por um fluxo de aprendizagem passiva anteriormente, pois desde a primeira rodada do fluxo já se utiliza um modelo treinado.

2.4 Heurísticas para fluxos de *active learning*

As heurísticas em um fluxo de *active learning* consistem em métodos pelo qual são selecionados os dados que serão enviados para a validação do oráculo. Em um único fluxo podem ser empregadas mais de uma heurísticas.

2.4.1 *Uncertainty Sampling*

Como o próprio nome sugere, essa heurística consiste em selecionar para consulta do oráculo os dados em que o o modelo possui maior dúvida, baseado em alguma métrica previamente escolhida. (LEWIS, 1995)

A métrica mais comum a ser utilizada para medir a incerteza dos modelos é a entropia , a mesma pode ser calculada da seguinte maneira:

$$Entropy = -\sum_{i=1}^{N} Pi \log_2 Pi \tag{2.1}$$

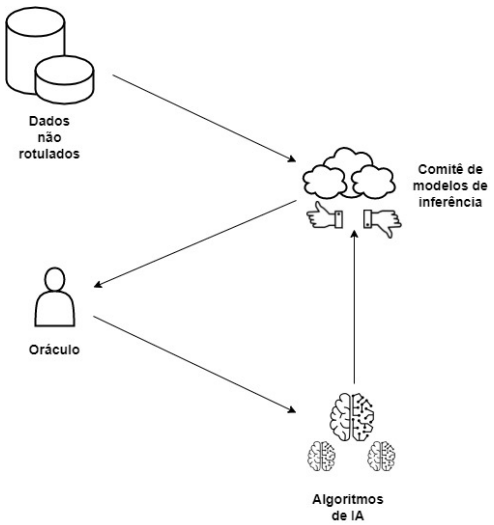
Onde:

Pi - corresponde a probabilidade de aleatoriamente escolher uma amostra de um determinado tipo. (SHANNON, 1948)

2.4.2 *Query-By-Committee*

A heurística de *Query-By-Committee* consiste em utilizar mais de um tipo de modelo no fluxo de aprendizagem ativa de maneira com que os diversos modelos identifiquem particularidades no dado (SEUNG; OPPER; SOMPOLINSKY, 1992)

Figura 4 – Fluxo de aprendizagem ativa utilizando *Query-by-Committee*



Fonte: autoral

A Figura acima demonstra o funcionamento de um fluxo de aprendizagem ativa utilizando comitês de modelos de inteligencia artificial. Diferentemente do fluxo de aprendizagem ativa comum, observado na Figura 4, que não utiliza comitês de modelos , neste fluxo observa-se que são utilizados mais de um algoritmo de aprendizado de maquina e com isso mais de um modelo é gerado, possibilitando que essa variedade de modelo aprendam sejam mais eficientes na etapa de escolha dos dados para enviar para o Oráculo.

2.5 Long Short Term Memory (LSTM)

As LSTMs são uma versão mais avançada e poderosa das redes neurais recorrentes simples (CHARNIAK, 2018) e foram criadas para solucionar o problema da perda de informações a longo prazo nas RNNs convencionais. A estrutura das LSTMs é mais complexa e inclui componentes como células de memória e estados ocultos, que permitem ao modelo capturar e manter dependências de longo prazo presentes nos dados de entrada. Graças a essa característica, as LSTMs são extremamente úteis em tarefas relacionadas ao processamento de sequências, como previsão de séries temporais e processamento de linguagem natural.

Uma LSTM bidirecional é uma combinação de duas LSTMs: uma que é executada do início ao fim da sequência e outra que é executada do fim ao início. Essa abordagem permite que a rede neural capture informações contextuais adicionais, já que ela tem acesso tanto ao passado quanto ao futuro em relação a cada ponto na sequência. Isso é especialmente útil em problemas de classificação de sequências, pois a rede pode aprender padrões e dependências tanto anteriores quanto posteriores a um determinado ponto. (SIAMI-NAMINI; TAVAKOLI; NAMIN, 2019)

A LSTM bidirecional tem a vantagem de acelerar e melhorar ainda mais o aprendizado do modelo, resultando em um desempenho aprimorado na classificação de sequências. Essa técnica tem sido amplamente utilizada em várias aplicações, incluindo reconhecimento de fala, tradução automática e análise de sentimentos, onde a contextualização bidirecional é crucial para obter resultados mais precisos e robustos.

3 Materiais e Métodos

3.1 Considerações Iniciais

Neste capítulo apresenta-se o plano metodológico utilizado para alcançar o objetivo deste estudo, isto é a maneira que o esta pesquisa foi estruturada.

O mesmo foi estruturado de maneira a demonstrar o planejamento das etapas que compõem esse estudo, de maneira a se alcançar cada um dos objetivos, mostrados na Seção 1.3, desse trabalho.

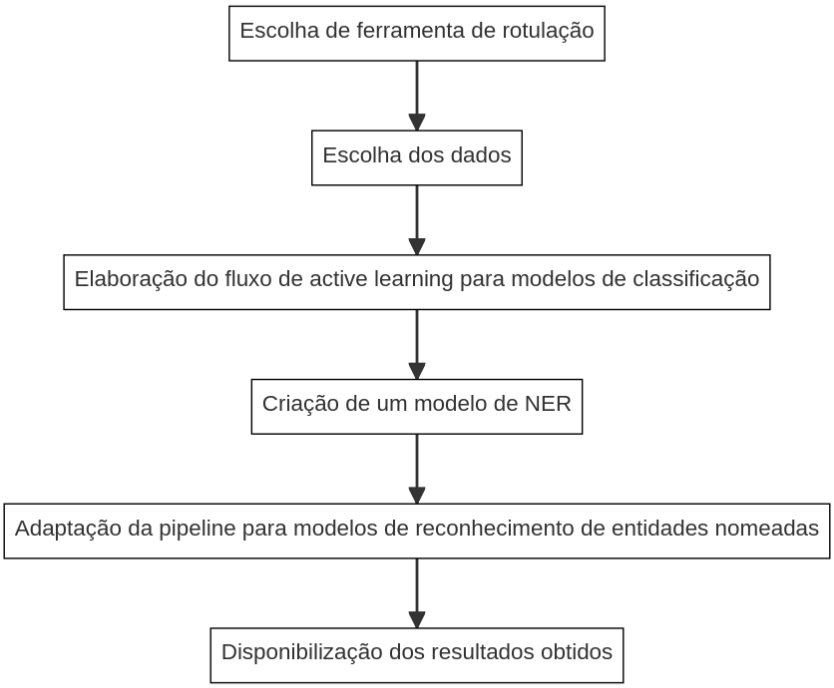
3.2 Plano Metodológico

O plano metodológico adotado nesse estudo foi dividido em 6 partes, sendo elas:

- Escolha de ferramenta de rotulação;
- Escolha dos dados;
- Elaboração do fluxo de *active learning* para modelos de classificação;
- Criação de um modelo de NER;
- Adaptação da *pipeline* para modelos de reconhecimento de entidades nomeadas;
- Disponibilização dos resultados obtidos;

Essas etapas estão organizadas na Figura 5

Figura 5 – Fluxo de atividades planejadas



3.2.1 Escolha de ferramenta de rotulação

A etapa de interação do oráculo com os dados é essencial para o funcionamento de um fluxo de aprendizagem ativa. Tendo em vista o objetivo principal do estudo, se faz necessário buscar uma ferramenta que funcione como interface para que o oráculo realize a rotulação e validação dos dados separados no *Selected*.

Essa ferramenta deve cumprir uma sequência de requisitos, sendo estes:

- Possuir acesso livre e gratuito;
- Permitir o uso para rotulação de dados textuais;
- Possuir interface para diferentes tipos de modelagens;
- Possibilitar integração com códigos autorais;
- Disponibilizar um controle de acesso aos dados;

3.2.2 Escolha dos Dados

Em projetos que fazem o uso de inteligência artificial a escolha dos dados a serem utilizados é uma etapa de muita importância, pois só após a mesma podemos pensar no

tipo de tratamento que será empregado, no tipo de modelagem que será desenvolvido, entre outras escolhas que são dependentes dessa escolha.

Portanto, realizar a escolha dos dados que serão utilizados nesse estudo compõe uma etapa bastante importante do mesmo. Tendo em vista os objetivos levantados na Seção 1.3, se faz importante perceber que neste trabalho devem ser utilizadas duas bases de dados distintas, de modo que em um primeiro momento pretende-se desenvolver uma fluxo de aprendizagem ativa para modelos de classificação e ao fim do estudo o uso em modelos de NER.

Os requisitos para as bases de dados a serem selecionadas são:

- Permitir acesso livre e gratuito;
- Possuir quantidade de dados que permita dividir os mesmo para a criação do fluxo de *active learning*;
- Possibilitar a fácil rotulação por um oráculo não especialista;
- Ser composto por dados textuais;

3.2.3 Elaboração do fluxo de *active learning* para modelos de classificação

Com o intuito de exercitar os conhecimentos obtidos com os temas contidos no Capitulo 2 se faz importante montar um fluxo simples de aprendizagem ativa para uma atividade menos complexa. Essa pipeline também tem como objetivo validar a ferramenta escolhida e permitir que a integração com ela seja experimentada de maneira a verificar se seu uso no fluxo principal se faz viável.

Para essa etapa intermediária foi escolhida a atividade de classificação por ser uma atividade que possui maior documentação de experimentos na internet e menor quantidade de rótulos em comparação com a atividade de NER.

Para realizar essa etapa pretende-se dividir os dados escolhidos na etapa anterior, mostrada na Seção 3.2.2 de maneira que se mantenham apenas parte dos dados na amostra *Labelled* enquanto o restante dos dados perderá os seus rótulos e irá compor o pool, de modo com que o dado do já rotulado será utilizado para treinar a primeira versão do modelo, iniciando assim o fluxo de aprendizagem ativa.

Ao fim dessa etapa espera-se verificar se a ferramenta escolhida na etapa anterior, mostrada na Seção 3.2.1, se comportou de maneira adequada na experimentação.

Outro objetivo dessa etapa é possibilitar que possíveis melhorias interessantes sejam levantadas e implementadas na versão final do estudo quando se utilizará a pipeline com modelos de reconhecimento de entidades nomeadas.

3.2.4 Criação de um modelo de NER

Após a construção do fluxo de classificação utilizando as ferramentas escolhidas anteriormente, é essencial realizar a experimentação e construção de um ou mais modelos de Reconhecimento de Entidades Nomeadas (NER). A construção desses modelos difere significativamente dos modelos de classificação, e é necessário explorar bibliotecas e tecnologias que possibilitem a integração com o fluxo de Active Learning.

A etapa de experimentação e construção de modelos de NER requer uma abordagem mais especializada devido à natureza específica do reconhecimento de entidades nomeadas. Nesse processo, é necessário identificar e extrair informações relevantes, como nomes de pessoas, locais, organizações, datas e outras entidades específicas em um texto. Isso exige a aplicação de técnicas avançadas de processamento de linguagem natural como o uso de redes neurais e também a utilização de bibliotecas e ferramentas especializadas, como Spacy, NLTK ou Stanford NER.

Os objetivos dessa etapa são:

- Obter um modelo com uma acurácia alta para a base de dado escolhida anteriormente;
- Aprender como fazer e as peculiaridades desse tipo de modelagem.

3.2.5 Adaptação da *pipeline* para modelos de reconhecimento de entidades nomeadas

Nessa etapa do estudo pretende-se realizar a elaboração do objetivo principal dele, isso é construir uma *pipeline* de aprendizagem ativa para a elaboração de modelos que realizem a atividade de reconhecimento de entidade nomeada.

Como objetivos a serem alcançados ao fim da construção dessa *pipeline* estão:

- Implementar melhorias que tenham sido verificadas na etapa de construção do fluxo anterior, mostrado na Seção 3.2.3
- Verificar a viabilidade da criação de um fluxo de *active learning*

3.2.6 Disponibilização dos resultados obtidos

Nessa etapa final tem se como objetivo realizar a divulgação dos resultados obtidos, isso se dará por diferentes maneiras. Será disponibilizados os dados para realizar a validação dos resultados apresentados por todas as etapas aqui contidas.

O código fonte construído para a criação dos dois fluxos deve ser disponibilizado de modo a permitir a reprodução desse estudo por interessados. Deve-se atentar pois o

mesmo deve ser viabilizado de modo que permita a execução do fluxo em diversos sistemas operacionais.

4 Resultados e discussão

4.1 Escolha de Ferramentas

Durante esta etapa do estudo foram verificadas diversas ferramentas de rotulação de dados para servir como interface de interação do oráculo com o fluxo de aprendizagem ativa.

As ferramentas analisadas foram avaliadas de acordo com os requisitos previamente levantados na Seção 3.2.1.

As ferramentas candidatas no estudo foram:

- LabelBox
- Label Studio
- Light Tag
- Make Sense

Durante as experimentações envolvendo a ferramenta *Light Tag* foi constatado que a mesma é uma ferramenta paga para seu uso coorporativo, sua versão gratuita permite apenas um anotador o que seria um limitante para grandes volumes de dados, a versão gratuita também não permite integração com dados pré anotados por IA.

A ferramenta *Make Sense* não apresenta instalação complexa, funcionando diretamente do navegador. Porém a mesma nos testes realizados apresentou interface apenas para a rotulação de imagens, não atendendo assim as demandas desse trabalho.

A *LabelBox* teve como principal problema em seu uso uma instalação que se mostrou mais complexa que a da *Label Studio* de maneira que com o curto período de tempo para a realização desse trabalho foi optado por uma configuração mais simples.

Devido a essas experimentações a ferramenta escolhida para realizar o propósito do estudo foi o *Label Studio*.

4.1.1 *Label Studio*

O *Label Studio* consiste em uma poderosa ferramenta aberta de rotulação de dado. Possui como principais vantagens para o estudo:

- **Flexibilidade**

A ferramenta suporta diversos tipos de dados, permitindo então que seja utilizada por diferentes tipos de estudos. Esse ponto se mostrou importante pois na etapa de conhecimento da ferramenta foi construído um fluxo de aprendizagem ativa utilizando dados textuais, porém para a tarefa de classificação.

- **Facilidade**

  A ferramenta possui uma boa documentação e tutoriais que permitem uma fácil instalação e uso correto a ferramenta.

- **Comunidade Ativa**

  A comunidade que utiliza o *software* é bastante ativa e participativa em fóruns, o que é interessante para solucionar possíveis problemas que venham a existir no desenvolver do fluxo.

- **Integração**

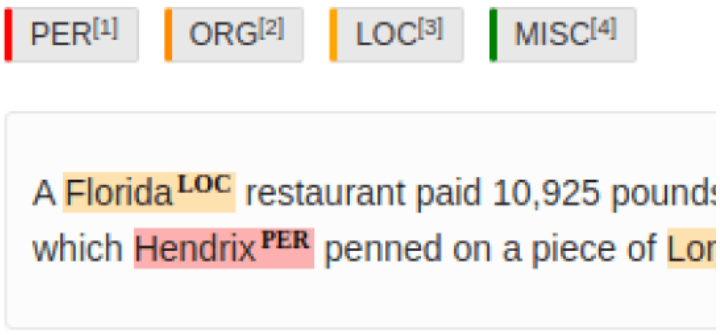
  A ferramenta possui diversas interfaces de comunicação para que seja utilizada em um fluxo de dados autoral, isso se mostra interessante para o projeto, de modo que isso possibilita que a mesma seja integrada facilmente ao fluxo construído.

- **Integração com Modelos de IA**

  A ferramenta possui a possibilidade de ser integrada com rótulos já estabelecidos anteriormente por um modelo de inteligencia artificial, desse modo permitindo que um fluxo de aprendizagem ativa seja integrado a mesma.

A Figura 6 apresenta um exemplo da interface de rotulação da ferramenta utilizando dados de NER.

Figura 6 – Exemplo de funcionamento do *Label Studio*



Fonte: (LABEL. . . ,)

4.2 Escolha dos dados

Como planejado para esta etapa mostrada na Seção 3.2.2 foram estudadas diversas bases de dados para serem utilizadas para que fossem utilizadas na criação das *pipelines* de aprendizagem ativa.

Foram buscados dados para os dois fluxos previstos no estudo, dados para classificação textual e dados para reconhecimento de entidade nomeada.

4.2.1 Dados para classificação

Para o fluxo de classificação de dados foram estudadas diversas bases de dados baseados nos critérios já levantados na etapa mostrada na Seção 3.2.2 e com base nisso a base de dados escolhida foi a ***Portuguese Tweets for Sentiment Analysis***.

Essa base de dados é composta por cerca de 800 mil *tweets* em português que foram divididos em sentimentos, positivos e negativos, para a realização de estudos de análise de sentimentos. (PORTUGUESE. . . ,)

Esta base de dados está dividida em 4 arquivos, sendo eles:

- ***Tweets with Theme***

  Dado relacionado a *tweets* que possuem termos politicos, possui apenas 60 mil tuplas.

- ***No theme Tweets***

  Dados coletados utilizando uma heurística com base em emoticons para distinguir quais dos tweets são positivos e quai são negativos.

  Possui cerca de 780 mil tuplas e por conta desse volume foi o arquivo escolhido para alimentar o fluxo de aprendizagem ativa.

- ***Neutral Tweets from Hashtags***

  Tweets coletados utilizando *hashtags*, possui cerca de 15 mil tuplas.

- ***Neutral Tweets form News accounts***

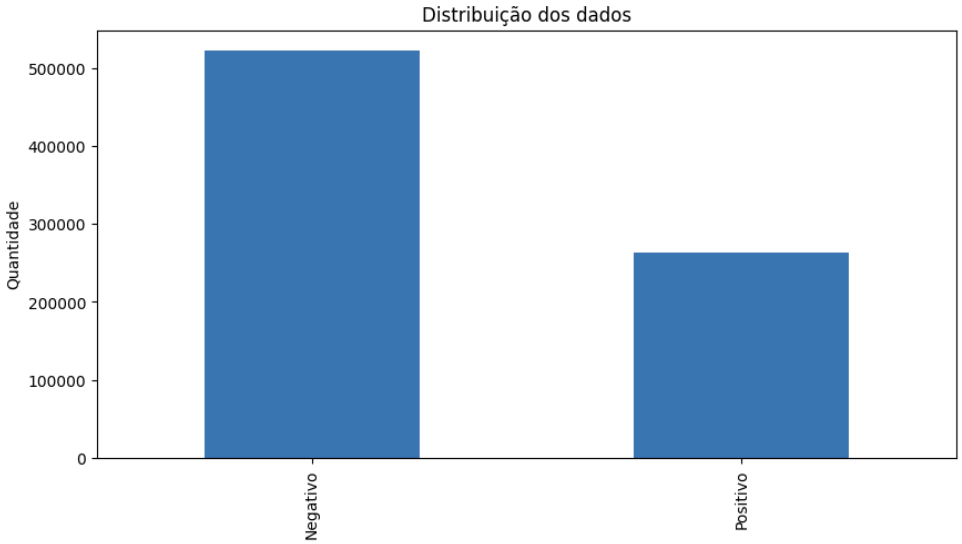
  Tweets coletados de contas que compartilham notícias e por tanto tendem a serem neutras, possui por volta de 35 mil tuplas.

Os dados contidos no *dataset* escolhido estão distribuídos de acordo com a Tabela 1 e podem ser visualizados na Figura 7.

Tabela 1 – Distribuição dos dados de classificação

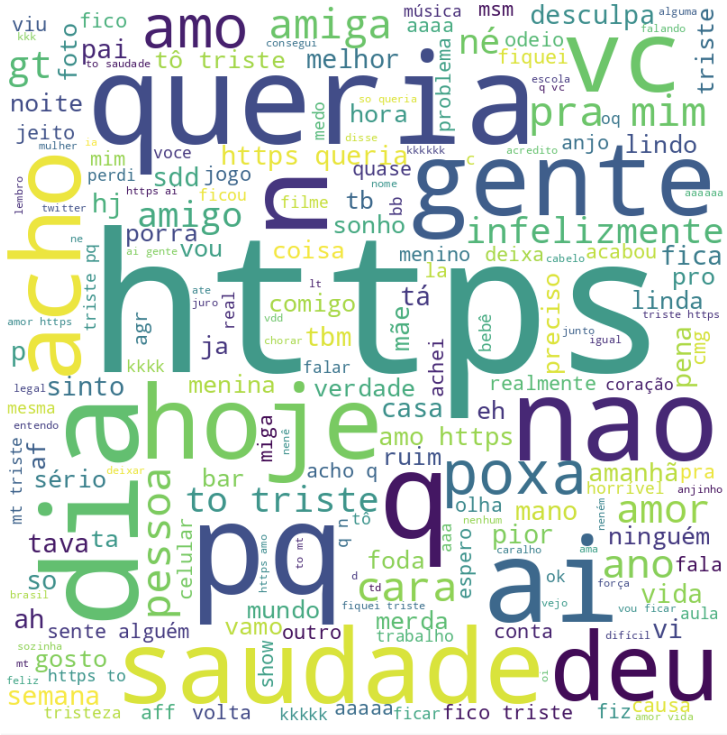
Rótulo	**Quantidade**	**Frequência**
Negativo	522707	66.52%
Positivo	263107	33.48%

Figura 7 – Gráfico de distribuição dos dados



A Figura 8 apresenta uma amostra dos dados contidos na base de dados utilizando o gráfico chamado de nuvem de palavras.

Figura 8 – *Word Cloud* dos dados



4.2.2 Dados para NER

Para a modelagem do fluxo de aprendizagem ativa para modelos de reconhecimento de entidades nomeadas, o dataset escolhido foi o Name Entity Recognition (NER) Dataset, disponível no Kaggle [1]. Essa seleção foi baseada na riqueza do conjunto de dados, que possui diversas versões e é amplamente utilizado em estudos e tutoriais relacionados à construção de modelos de NER.

O fato de haver várias versões e trabalhos relacionados ao NER Dataset torna possível explorar diferentes abordagens, técnicas e algoritmos de aprendizado de máquina para construir modelos de NER. A existência de tutoriais e estudos de caso que utilizam esse dataset facilita o aprendizado e a compreensão dos conceitos e das melhores práticas envolvidas na construção de modelos de reconhecimento de entidades nomeadas.

Essa base de dados foi criada realizando anotações de entidades nomeadas em uma outra base de dados que é a *Groningen Meaning Bank (GBM)*, que consiste em um dataset com diversas anotações textuais de diversos tipos e origens. Esse dado é mantido pelo Instituto de Linguística Computacional da Universidade de Groningen, na Holanda. O mesmo é anotado manualmente por linguistas especializados, o que garante a alta qualidade das anotações. (BOS et al., 2017)

O *dataset* possui no total 6 entidades nomeadas ,sendo essas:

- `geo` *(Geographical Entity)*

  Esta *tag* é usada para identificar entidades que representam locais geográficos, como países, cidades, rios e montanhas.

- `org` *(Organization)*

  Esta *tag* é utilizada para marcar entidades que são nomes de organizações ou estruturas similares, como empresas, instituições governamentais ou organizações sem fins lucrativos.

- `per` *(Person)*

  Esta *tag* é aplicada para identificar nomes de pessoas presentes em uma determinada sentença.

- `gpe` *(Geopolitical Entity)*

  Esta *tag* é usada para representar entidades geopolíticas na frase, tais como países, estados, regiões ou outras divisões políticas.

- `tim` *(Time indicator)*

[1] Kaggle, Disponível em: <https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus>

Esta *tag* é atribuída a entidades que representam informações temporais, como datas, horários ou expressões relacionadas ao tempo.

- `art` *(Artifact)*

  Esta *tag* é utilizada para marcar entidades que representam um ou mais artefatos na sentença, como produtos manufaturados, obras de arte ou objetos específicos.

- `eve` *(Event)*

  Esta *tag* é aplicada a entidades que correspondem a eventos, como reuniões, conferências, shows ou ocorrências específicas.

- `nat` *(Natural Phenomenon)*

  Esta *tag* é atribuída a termos que representam fenômenos naturais, como chuva, vento, terremotos ou outros eventos relacionados à natureza.

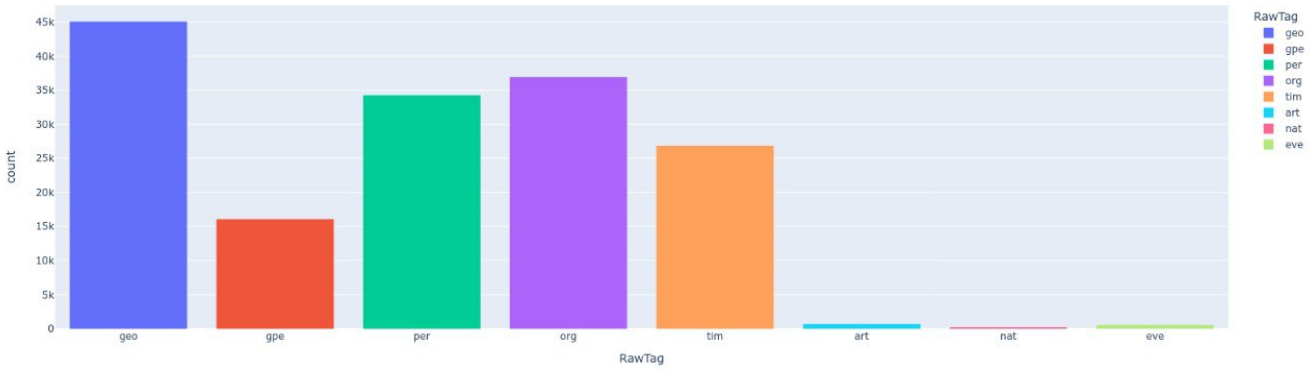
A Tabela 2 mostra a divisão das entidades em cada uma dessas entidades.

Tabela 2 – Quantidade de tokens por tag

Tag	Quantidade de tokens
O	887.908
geo	45.058
org	36.927
per	34.241
tim	26.861
gpe	16.068
art	699
eve	561
nat	252

A Figura 9 representa como estão dispostos os dados entre essas entidades dentro do dado escolhido

Figura 9 – Distribuição dos dados entre as *tags*



4.3 Elaboração do fluxo de *active learning* para modelos de classificação

Durante esta etapa do projeto, na qual tem como objetivo experimentar os conhecimentos adquiridos anteriormente em um fluxo mais simples como mostrado na Seção 3.2.3 , foi construído essa pipeline de maneira a atender os requisitos levantados também na Seção 3.2.3.

Esta fase do estudo foi feita com o intuito de verificar o correto funcionamento do fluxo de aprendizagem ativa e não com o intuito de construir um modelo que possua grande assertividade na atividade em questão. Por conta disso, foi optado por construir um modelo mais simples de classificação binária de dados textuais, pois assim será possível verificar as etapas de tratamento dos dados, integração dos micro-serviços, construção de um modelo de inteligência artificial e também da análise dos modelos gerados em cada rodada do estudo.

Os dados utilizados para alimentar todo este processo foram os citados na Seção 4.2.1.

A solução foi construída utilizando microsserviços com o intuito de permitir que algum módulo do mesmo seja reutilizado em outros fluxos e também por permitir que a solução aqui construída seja escalável. Para realizar o isolamento de dependências entre os microsserviços foi utilizada a tecnologia docker, pois além de trazer esse benefício o mesmo ainda facilita que o fluxo seja executado em ambientes distintos.

Os microsserviços que compõem a solução e suas respectivas responsabilidades são:

- **backend**

  Código gerado em python para integrar a geração dos modelos com o **label-studio** para fazer o fluxo de *active learning*.

- **label-studio**

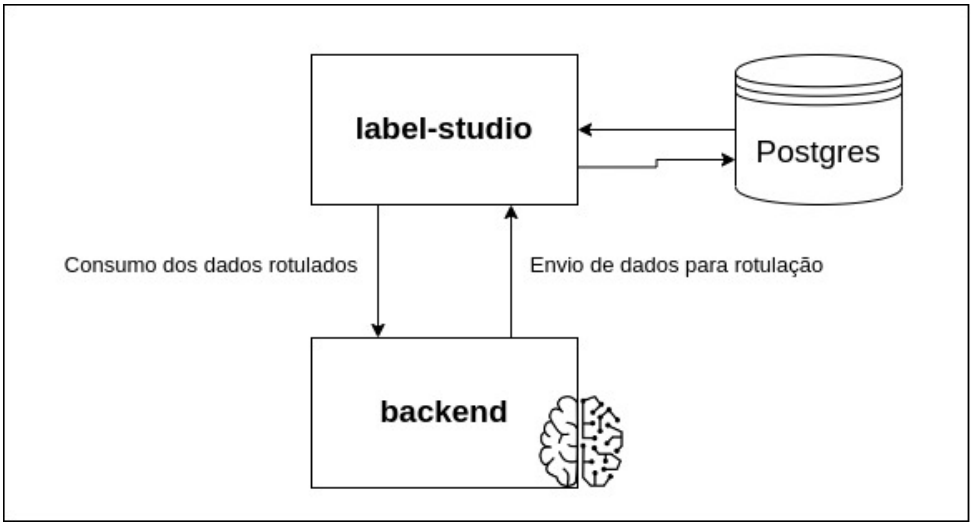
  Ferramenta escolhida para ser a interface de rotulação do oráculo, como mostrado na Seção 4.1

- **db**

  Banco de dados postgres que é utilizado para armazenar os dados de rotulação que são enviados ao oráculo, armazena também usuários, projetos e outros dados de responsabilidade do **label-studio**.

A Figura 10 apresenta o diagrama de microsserviços do projeto.

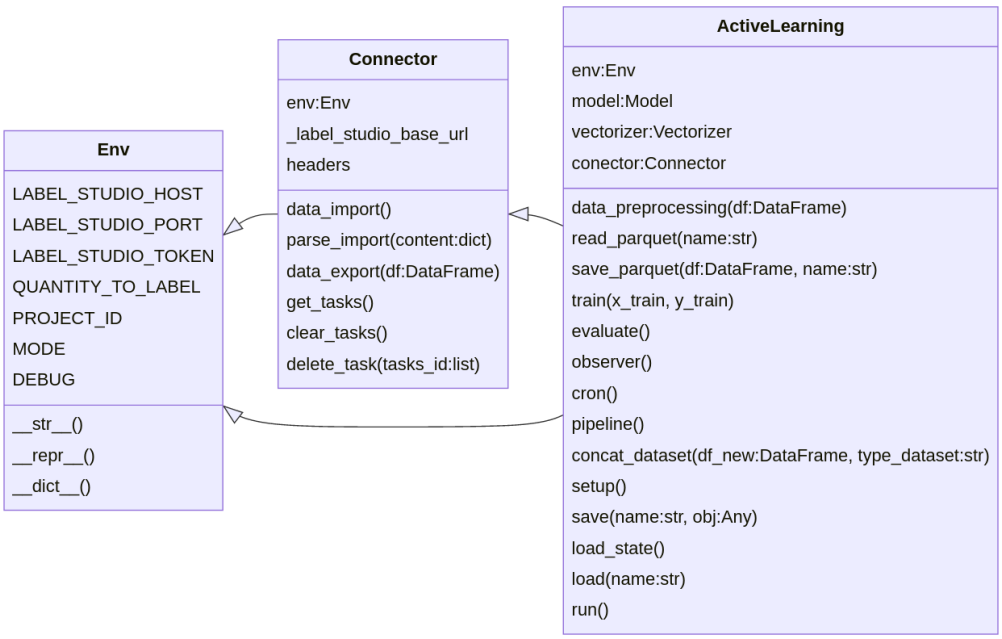
Figura 10 – Diagrama de pacotes da solução desenvolvida



Fonte: Autoral

Durante a fase de desenvolvimento do código fonte foi utilizado o paradigma da orientação a objetos pois o mesmo permite uma boa reutilização de código, modularização e também que posteriormente na construção do fluxo que irá usar modelos de reconhecimento de entidades nomeadas seja facilmente adaptado o mesmo código fonte. A Figura 11 mostra o diagrama de classes da solução construída.

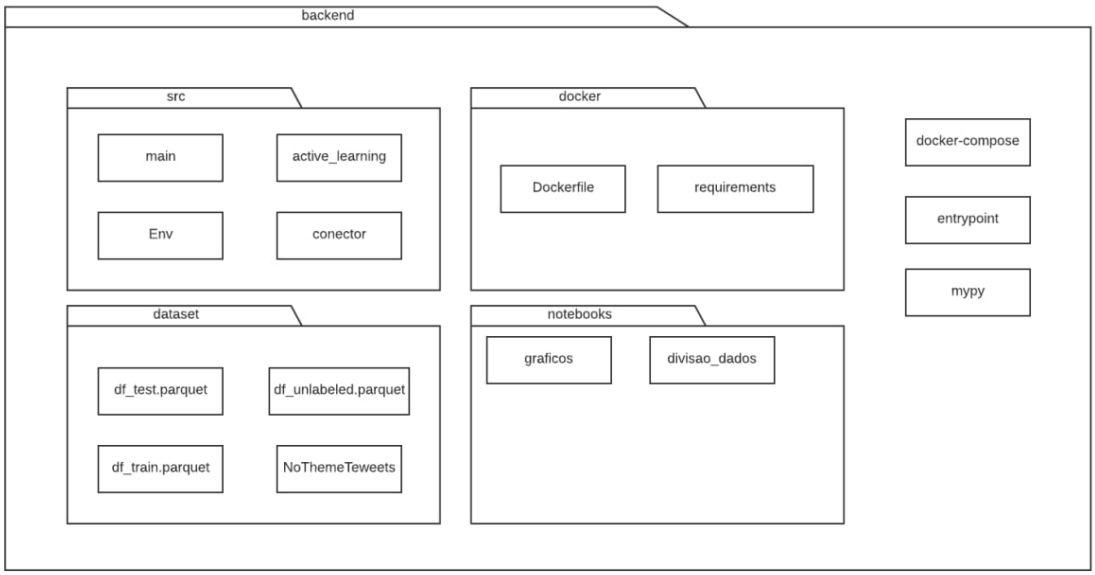
Figura 11 – Diagrama UML da solução desenvolvida



Fonte: Autoral

A Figura 12 apresenta o diagrama de pacotes da solução gerada nesta etapa:

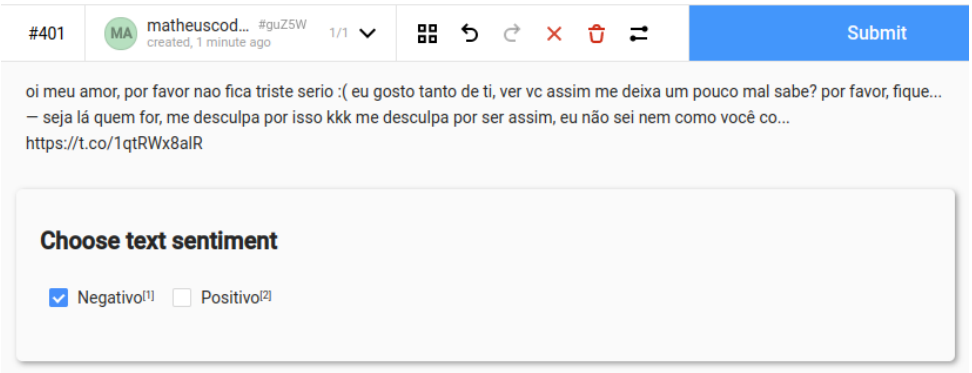
Figura 12 – Diagrama de pacotes da solução desenvolvida



Fonte: Autoral

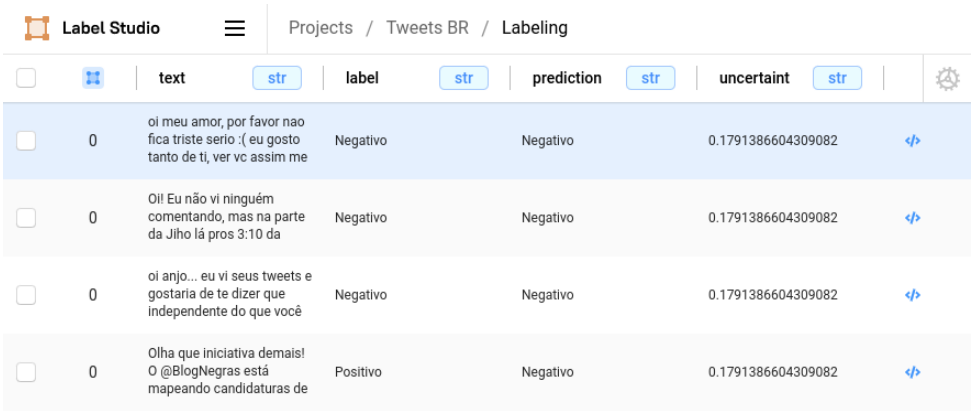
Durante esta etapa de experimentação foi executado o fluxo por 3 rodadas sendo enviado para o oráculo por etapa uma quantidade de 50 frases. A pouca quantidade de dados enviados foi decidida pelo foco dessa etapa não ser em gerar um modelo performático e sim em realizar a validação de que o fluxo funciona. As Figuras 13 e 14 demonstram a interface do *Label studio* em funcionamento com os dados selecionados.

Figura 13 – Exemplo do funcionamento do fluxo



Fonte: Autoral

Figura 14 – Exemplo do fluxo em funcionamento



Fonte: Autoral

Durante as etapas da pipeline as métricas dos modelos gerados são armazenadas em um arquivo json, de maneira com que se consiga comparar cada um dos modelos.

4.4 Criação de um modelo de NER

Durante esta etapa, foram realizadas diversas tentativas de modelagem com os dados selecionados, com o objetivo de comparar os resultados obtidos em cada abordagem e identificar a arquitetura de modelo mais adequada ao fluxo de *active learning* construído anteriormente, conforme apresentado na seção correspondente.

Nessa fase de estudo, foram conduzidos experimentos que envolveram duas abordagens distintas, sendo elas:

- Utilização do *Spacy*
- Implementação de uma rede *Bi-LSTM* com *Tensorflow*

Com base na estrutura descrita na Seção 4.3, foi optado por utilizar a abordagem que emprega o modelo do *Tensorflow*. Essa escolha se deu devido aos resultados gerados por esse modelo, que possibilitam uma integração mais eficiente. Ao contrário do Spacy, cujos dados gerados estão em formato JSON, o modelo do *Tensorflow* gera dados no formato CSV. Essa diferença de formato torna a integração mais conveniente e compatível com outras etapas do processo.

O modelo gerado nesta etapa possui uma camada de entrada com tamanho 50. Em seguida, são adicionadas camadas intermediárias antes de finalizar com uma camada que utiliza *BiLSTM*. Essa configuração pode ser visualizada na Figura 15 e descrita de-

talhadamente na Tabela 3. Essa arquitetura foi construída com base em materiais que possuem o intuito de ensinar a construção de modelos de NER [2].

Figura 15 – Arquitetura da rede neural treinada

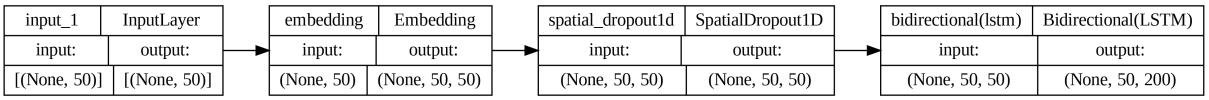


Tabela 3 – Arquitetura da rede neural

Camada	Formato de Saída
Embedding	(None, 50, 50)
Spatial dropout1d	(None, 50, 50)
Bidirectional LSTM	(None, 50, 200)

A Figura 15 apresenta a arquitetura do modelo, destacando as camadas e a sequência de operações. A camada de entrada recebe dados com tamanho 50, que são processados por camadas intermediárias para extração de características. A última camada utiliza *BiLSTM* para modelar as relações sequenciais nos dados.

A Tabela 3 fornece uma visão geral das camadas do modelo e seus respectivos formatos de saída. A camada de Embedding converte os dados de entrada para representações densas. A camada *Spatial dropout1d* aplica uma técnica de desativação espacial para evitar *overfitting*, permitindo que toda a *feature map 1D* seja desativada em todos os canais. A camada *Bidirectional LSTM* realiza a modelagem das sequências de entrada nos dois sentidos, fornecendo uma saída com formato (None, 50, 200) (ZHOU, 2022).

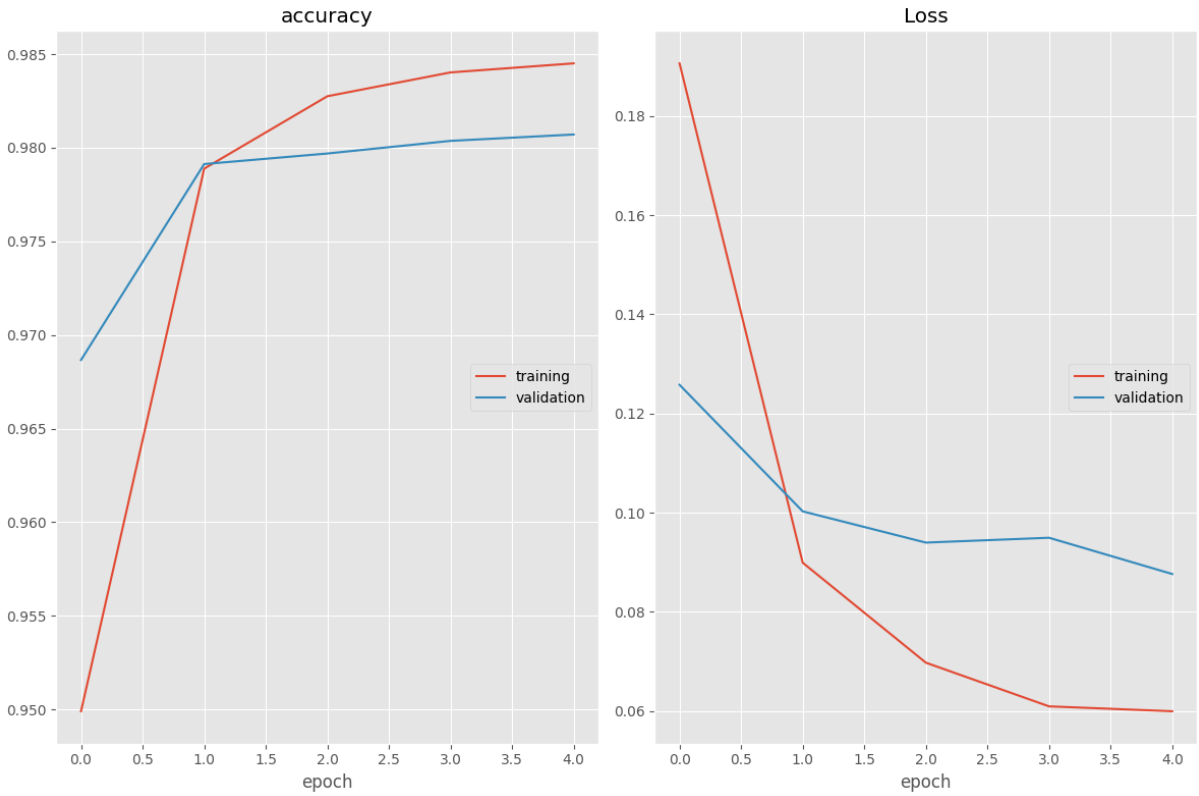
Para compilar o modelo, foi utilizado o otimizador Adam e a perda de entropia cruzada esparsa porque são escolhas comuns para problemas de classificação com várias classes, como o reconhecimento de entidades (NER).

A Figura 16 ilustra o progresso da acurácia (lado esquerdo) e da perda (*loss*) (lado direito) durante o treinamento do modelo. Observamos que a acurácia do modelo atinge um valor superior a 90% já na segunda época e se mantém consistente nas épocas subsequentes. Isso indica que o modelo está aprendendo com eficácia as relações entre as entradas e as classes alvo.

Por outro lado, o valor da perda diminui durante a maior parte do treinamento do modelo. Isso sugere que o modelo está otimizando os parâmetros para reduzir a discrepância entre as probabilidades previstas e as classes reais. À medida que o treinamento progride, a perda diminui, indicando que o modelo está se ajustando cada vez mais aos dados de treinamento.

[2] Medium, Disponível em:
<https://zhoubeiqi.medium.com/named-entity-recognition-ner-using-keras-lstm-spacy-da3ea63d24c5>

Figura 16 – Treinamento do modelo



Após realizar o treinamento do modelo, utilizamos os dados de validação para avaliar seu desempenho. Os resultados obtidos foram registrados na Tabela 4. Essa tabela apresenta as métricas de *loss* e acurácia, que fornecem informações cruciais sobre a capacidade do modelo de generalizar os padrões aprendidos durante o treinamento. A métrica de *loss* indica o quão bem o modelo está se ajustando aos dados, sendo desejável que os valores sejam baixos. Por outro lado, a acurácia fornece uma medida da taxa de classificações corretas realizadas pelo modelo. Com base nessas métricas, podemos obter uma visão detalhada do desempenho do modelo em relação aos dados de validação, permitindo-nos avaliar sua eficácia e identificar possíveis áreas de melhoria.

Tabela 4 – Métricas do modelo produzido

Métricas	Valor obtido
Loss de validação	0.0876503437757492
Acurácia	0.9807088971138

A Tabela 5 apresenta a saída do modelo quando utilizado em uma sentença do dado de validação.

Tabela 5 – Arquitetura da rede neural

Palavra	Verdadeiro	Predição
The	O	O
United	B-geo	B-geo
States	I-geo	I-geo
has	O	O
103	O	O
nuclear	O	O
power	O	O
plants	O	O
in	O	O
31	O	B-tim
states	O	O
.	O	O
Stand-By	O	O
Stand-By	O	O
Stand-By	O	O
Stand-By	O	O
Stand-By	O	O
Stand-By	O	O

4.5 Elaboração do fluxo de *active learning* para modelos de NER

Após obter o modelo treinado na Seção 4.4, foi realizada a etapa de integração com o fluxo de *active learning*, como descrito na Seção 3.2.5. Essa integração permite aprimorar ainda mais o desempenho do modelo, aproveitando os benefícios do *active learning* para melhorar a qualidade das anotações e reduzir a necessidade de rotulação manual extensiva.

A estrutura da solução segue o que foi apresentado na Seção 4.3, mantendo a abordagem e os componentes essenciais. Estruturas como as classes descritas e arquitetura de micro-serviços são mantidas nessa etapa, tendo em vista que essa modelagem foi idealizada com a intenção de permitir que modelos de diferentes tipos possam ser integrados no fluxo.

Ao incorporar o modelo treinado no fluxo de *active learning*, foi possível criar um ciclo iterativo de refinamento, onde o modelo é constantemente atualizado com novos dados rotulados e aprimorado com base nesses novos exemplos.

Para realizar o experimento de integração do fluxo de *active learning* com o modelo de reconhecimento de entidades nomeadas, os dados foram divididos em três partes distintas, cada uma com um propósito específico:

- **Dado de treino**

Essa porção dos dados foi utilizada para treinar a primeira versão do modelo. Através desse treinamento inicial, obtivemos um modelo de base que serviu como ponto de partida para o processo de *fine-tuning*. Durante o fluxo de *active learning*, esse modelo inicial foi refinado e ajustado continuamente à medida que novos dados rotulados foram incorporados. Além disso, com base nessa primeira versão do modelo, foi possível selecionar as amostras na qual o modelo possuía uma maior incerteza para a primeira rodada de rotulação, direcionando o esforço de rotulação para as áreas de maior relevância.

- **Dado de teste**

  Essa parte dos dados foi reservada para avaliar e coletar métricas dos modelos gerados ao longo das etapas do fluxo de *active learning*. A cada iteração do fluxo, o modelo atualizado foi aplicado a esse conjunto de teste para mensurar sua eficácia e avaliar seu desempenho. Essas métricas são armazenadas para que futuramente seja analisadas.

- **Dado não rotulado**

  Essa parcela de dados teve sua rotulação desconsiderada e foi destinada ao oráculo para a etapa de rotulação. Esses dados não rotulados foram selecionados com base nos critérios estabelecidos pelo fluxo de *active learning*, baseado na incerteza de classificação. Ao enviar esses dados para o oráculo rotulá-los, foi possível obter as anotações necessárias para expandir o conjunto de treinamento e continuar o processo de refinamento do modelo.

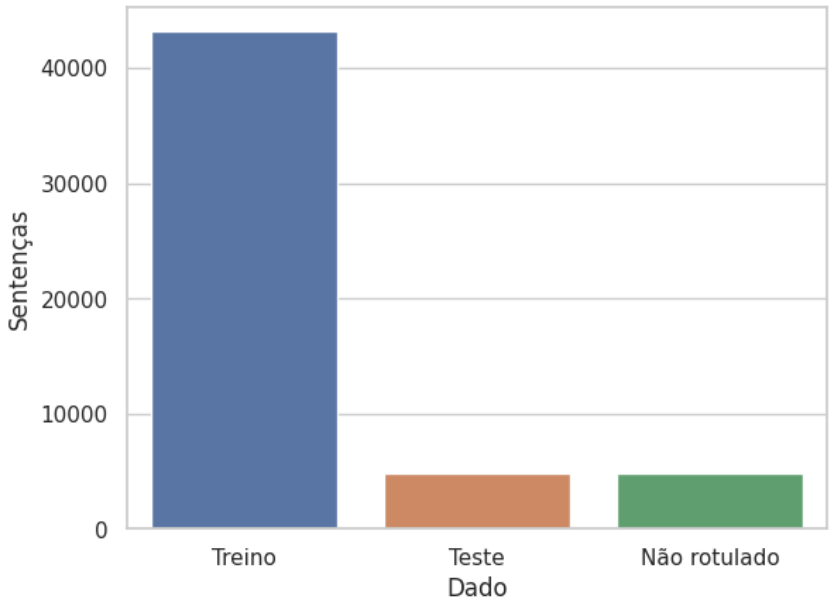
Essa divisão dos dados em três partes distintas permitiu um fluxo contínuo e eficiente no processo de treinamento e melhoria do modelo de reconhecimento de entidades nomeadas. Através da interação entre o dado de treino, o dado de teste e o dado não rotulado, foi possível iterar e aprimorar gradualmente o modelo, direcionando o esforço de rotulação para as amostras mais relevantes e avaliando seu desempenho em dados independentes.

A distribuição dos dados entre as diferentes partes do experimento pode ser visualizada na Tabela 6, que apresenta a porcentagem que representa do dado original, quantidade de sentenças e *tokens* em cada uma delas. Além disso, a Figura 17 oferece uma representação gráfica da distribuição das sentenças entre esses conjuntos de dados.

Tabela 6 – Divisão dos dados

Dado	Porcentagem	Quant. de sentenças	Quant. de tokens
Treino	80%	43164	838860
Teste	10%	4795	209715
Não rotuladas	10%	4795	103933
Total	100%	52754	1152508

Figura 17 – Divisão das sentenças no dado



Os percentuais escolhidos para realizar a divisão do dado para esse experimento foram definidos tendo em vista que para experimentar o funcionamento do fluxo em poucas rodadas de rotulação e obter um modelo com qualidade se faz necessário que a primeira versão do modelo tenha a maior quantidade de dados para treinar possível. Essa escolha também foi importante tendo em vista que para a execução do fluxo tinha-se apenas um oráculo para realizar a rotulação dos dados que fazem parte do *pool*.

Os percentuais escolhidos para a divisão dos dados neste experimento foram determinados com base em dois principais objetivos. Primeiramente, buscamos obter uma quantidade significativa de dados de treinamento para a primeira versão do modelo, a fim de garantir um treinamento inicial robusto. Isso é crucial para estabelecer uma base sólida e permitir que o modelo capture os padrões e informações relevantes presentes nos dados. Quanto mais dados de treinamento disponíveis, maior é a probabilidade de o modelo aprender de forma eficaz e gerar resultados de melhor qualidade.

Em segundo lugar, consideramos a disponibilidade limitada de recursos humanos para a tarefa de rotulação. Com apenas um oráculo responsável pela rotulação dos dados no *pool*, é importante otimizar a distribuição dos dados para garantir uma alocação eficiente de tempo e esforço de rotulação. Assim, foi necessário encontrar um equilíbrio entre

a quantidade de dados rotulados necessários para melhorar o modelo e a capacidade do oráculo de realizar essa rotulação no tempo disponível.

Com essa abordagem, é possível realizar um experimento eficiente, aproveitando ao máximo os dados disponíveis e otimizando o processo de rotulação para obter um modelo com qualidade em um curto período de tempo.

Para realizar a rotulação dos dados, utilizamos novamente o *Label Studio* como interface principal. Optamos por essa escolha devido aos benefícios e recursos que foram discutidos em detalhes na Seção mencionada anteriormente. O *Label Studio* nos permitiu de forma eficiente e intuitiva rotular os dados de entidades nomeadas (NER) para o nosso projeto.

Graças à flexibilidade do *Label Studio*, pudemos estruturar os dados de NER de forma organizada e coerente. As Figuras mencionadas (Figuras 18 e 19) ilustram como os dados de NER foram apresentados dentro do projeto.

Ao utilizar o *Label Studio*, fomos capazes de visualizar os dados de maneira clara e precisa, tornando o processo de rotulação mais eficiente e eficaz. A interface intuitiva do *Label Studio* permitiu que os anotadores trabalhassem de forma ágil, facilitando a identificação e rotulação das entidades nomeadas nos textos.

Figura 18 – Funcionamento do fluxo

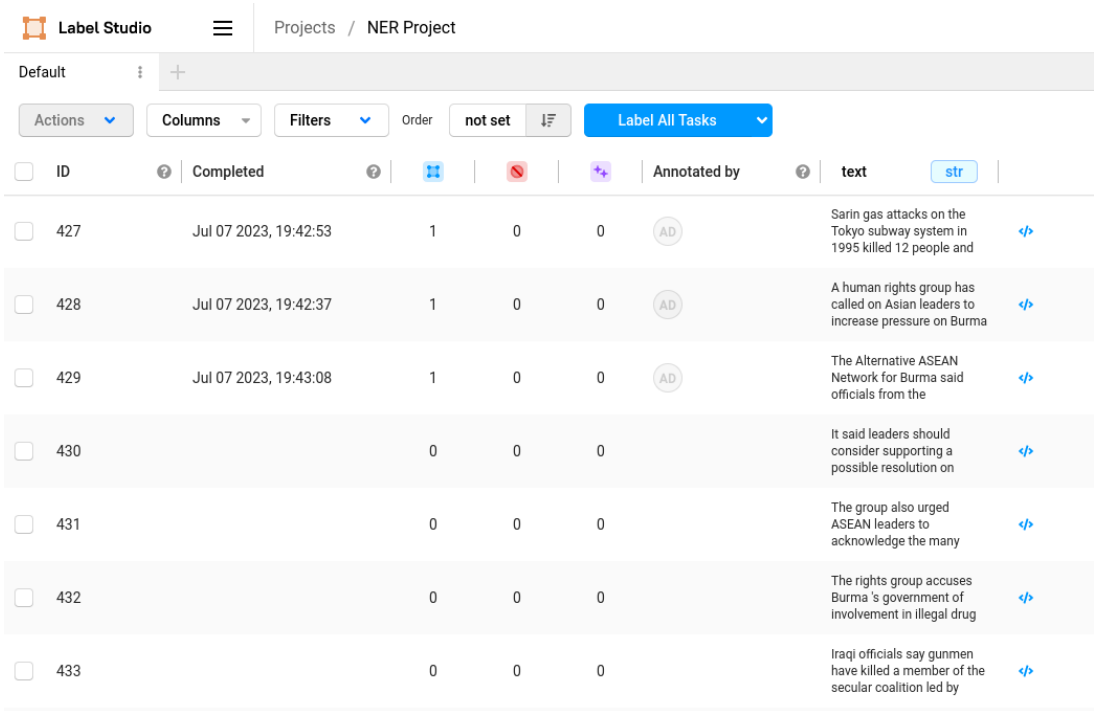
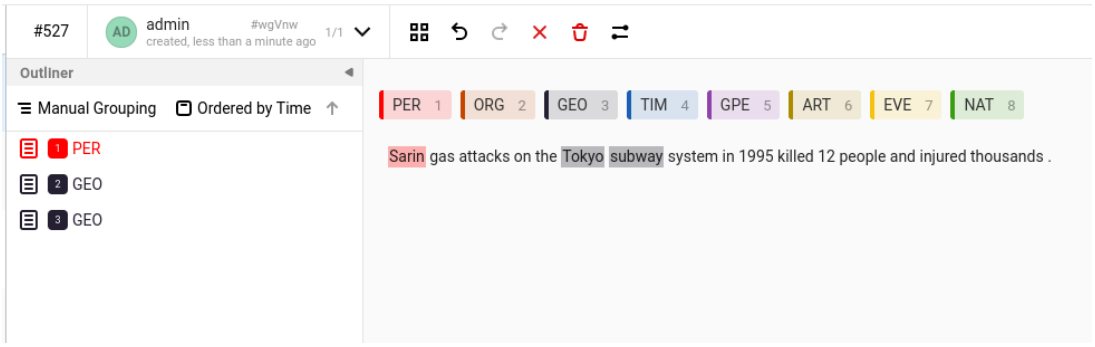


Figura 19 – Anotação do dado de NER



4.6 Disponibilização dos resultados obtidos

Esta etapa tem como objetivo levantar os resultados obtidos nesse trabalho de maneira a disponibilizar e assim permitir que os resultados sejam avaliados e também reproduzidos.

Como apresentado nas seções anteriores esse trabalho possui diversos resultados que podem ser analisados, sendo eles:

A etapa atual tem como objetivo fornecer uma visão geral dos resultados obtidos neste trabalho, permitindo que sejam avaliados e reproduzidos. Abaixo, apresento os principais resultados alcançados nas seções anteriores:

- **Fluxo de active learing para modelos de classificação**

  Neste trabalho, desenvolvemos um fluxo de trabalho de *Active Learning* para modelos de classificação. Demonstramos como o *Active Learning* pode ser aplicado para otimizar a rotulação dos dados, melhorar o desempenho do modelo e reduzir o esforço de rotulação manual. Apresentamos os módulos do fluxo, incluindo micro-serviços que compõem a arquitetura, dados, modelo escolhido e estrutura da solução.

- **Construção de um modelo de NER**

  Desenvolvemos um modelo de Reconhecimento de Entidades Nomeadas (NER) para identificar e classificar entidades em textos. Utilizamos uma abordagem baseada em aprendizado supervisionado, onde treinamos o modelo com um conjunto rotulado de dados de treinamento. Exploramos diferentes arquiteturas de modelos, técnicas de pré-processamento de texto e algoritmos de treinamento. Avaliamos o desempenho do modelo utilizando métricas de precisão. Os resultados mostraram uma capacidade promissora do modelo em identificar entidades nomeadas com precisão.

- **Fluxo de active learing para construção de um modelo de NER**

  Extendendo o fluxo de *Active Learning* mencionado anteriormente, aplicamos esse fluxo para a construção do modelo de NER. Demonstramos como o *Active Learning* pode ser adaptado para selecionar amostras relevantes para a rotulação de entidades nomeadas e melhorar o desempenho do modelo. Combinamos o fluxo de *Active Learning* com técnicas de transferência de aprendizado para otimizar ainda mais o processo de treinamento.

Esses resultados são fundamentais para a compreensão do trabalho realizado, permitindo que os resultados sejam avaliados por outros pesquisadores e reproduzidos em futuros estudos. As abordagens propostas, o fluxo de Active Learning e o modelo de NER

construído, fornecem insights valiosos para o desenvolvimento de sistemas de processamento de linguagem natural e têm o potencial de contribuir para avanços na área de aprendizado de máquina aplicado a textos.

Além da estrutura organizacional do trabalho mencionada anteriormente, é importante destacar que os resultados obtidos durante as etapas do projeto, bem como os códigos desenvolvidos, foram disponibilizados em um repositório no *GitHub* [3]. Essa abordagem permite que qualquer pessoa interessada possa acessar e experimentar o fluxo proposto no trabalho.

Ao disponibilizar os resultados e códigos no *GitHub*, promove-se a transparência e a reprodutibilidade do trabalho, permitindo que outros verifiquem os métodos utilizados, reproduzam os resultados e construam sobre a base já estabelecida. Isso contribui para o avanço do conhecimento na área e incentiva a colaboração entre os pesquisadores.

[3]Github, Disponível em: <https://github.com/Matheus73/TCC>

5 Conclusão

Este trabalho acadêmico teve como foco primordial o desenvolvimento de um fluxo de trabalho de *Active Learning* em modelos de classificação, especificamente no campo do Reconhecimento de Entidades Nomeadas (NER). O *Active Learning* foi demonstrado como uma ferramenta crucial para otimizar o processo de rotulação dos dados, aperfeiçoando o desempenho do modelo e diminuindo a demanda de rotulação manual.

No início, detalhamos os módulos distintos que compõem o fluxo de *Active Learning*, incluindo a arquitetura baseada em micro-serviços, a seleção criteriosa dos dados, a escolha do modelo apropriado e a estrutura global da solução proposta. Os resultados obtidos enfatizaram a eficácia desse procedimento, que se traduziu em uma solução robusta e modular que possibilitou mais adiante que fosse aproveitada em um fluxo com modelos de NER, além de proporcionar uma maneira mais eficiente para a rotulação dos dados.

Posteriormente, voltamos nossa atenção para a construção de um modelo de NER. Utilizando a metodologia do aprendizado supervisionado, treinamos o modelo com um conjunto de dados de treinamento previamente rotulado. Diversas arquiteturas de modelos, técnicas de pré-processamento de texto e algoritmos de treinamento foram explorados. Os resultados evidenciaram a capacidade promissora do modelo em identificar e classificar as entidades nomeadas de forma precisa.

Finalmente, ampliamos a abrangência do fluxo de *Active Learning* e o implementamos na construção do modelo de NER. Demonstramos como o *Active Learning* pode ser ajustado para selecionar as amostras mais relevantes para a rotulação de entidades nomeadas. Também aliamos o fluxo de *Active Learning* a técnicas de transferência de aprendizado, culminando em um processo de treinamento mais eficaz e otimizado.

Os resultados alcançados neste trabalho realçam o potencial do *Active Learning* como uma ferramenta valiosa na construção de modelos de classificação, especialmente no domínio do Reconhecimento de Entidades Nomeadas. Esta abordagem não apenas aprimora a eficiência da rotulação dos dados, mas também intensifica o desempenho dos modelos em contextos práticos. A sinergia do *Active Learning* com outras técnicas, como a transferência de aprendizado, abre novos horizontes para pesquisas futuras e desenvolvimentos neste campo.

Contudo, é importante salientar que a aplicação do fluxo de *Active Learning* em grandes volumes de dados pode ser restrita pela disponibilidade de tempo e a necessidade de especialistas para rotulação manual. Ainda que tenhamos evidenciado a eficácia deste procedimento na redução do esforço de rotulação, a aplicação em conjuntos de da-

dos volumosos exigirá considerações adicionais, como a implementação de estratégias de amostragem eficazes e a distribuição da tarefa de rotulação entre múltiplos especialistas.

Portanto, embora os resultados obtidos sejam inspiradores e indiquem a viabilidade do *Active Learning* como uma estratégia eficiente, é imprescindível considerar os desafios práticos e logísticos associados à sua implementação em larga escala. Futuras pesquisas podem se concentrar em encontrar soluções para esses obstáculos, como a otimização dos algoritmos de seleção de amostras, o uso de técnicas de rotulação parcial e a colaboração entre diferentes especialistas para a rotulação manual. Estas abordagens podem permitir uma aplicação mais efetiva e abrangente do *Active Learning* em problemas de classificação e, particularmente, no Reconhecimento de Entidades Nomeadas.

Referências

BOS, J. et al. The groningen meaning bank. In: IDE, N.; PUSTEJOVSKY, J. (Ed.). *Handbook of Linguistic Annotation*. [S.l.]: Springer, 2017. v. 2, p. 463–496. Citado na página 30.

CHARNIAK, E. *Introduction to deep learning*. [S.l.]: Mit Press, 2018. ISBN 9780262039512. Citado na página 20.

CORTI, M. de S. Métodos de active learning para algoritmos de classificação. 2021. Citado na página 17.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. [S.l.]: Pearson, 2020. Citado na página 15.

LABEL Studio — Text Named Entity Recognition Data Labeling Template. <https://labelstud.io/templates/named_entity.html>. (Accessed on 01/27/2023). Citado na página 27.

LEWIS, D. D. A sequential algorithm for training text classifiers: Corrigendum and additional data. In: ACM NEW YORK, NY, USA. *Acm Sigir Forum*. [S.l.], 1995. v. 29, n. 2, p. 13–19. Citado na página 19.

MURTHY, R.; KHAPRA, M. M.; BHATTACHARYYA, P. Improving ner tagging performance in low-resource languages via multilingual learning. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, ACM New York, NY, USA, v. 18, n. 2, p. 1–20, 2018. Citado na página 16.

NAMED Entity Recognition using SpaCy (NER) | by Akshay Sharma | The HumAIn Blog | Medium. <https://medium.com/in-pursuit-of-artificial-intelligence/named-entity-recognition-using-spacy-ner-da6eebd3d08>. (Accessed on 02/07/2023). Citado na página 15.

OBJETIVOS específicos do TCC: veja como escrever e exemplos. <https://regrasparatcc.com.br/primeiros-passos/objetivos-especificos-do-tcc/>. (Accessed on 01/27/2023). Citado na página 13.

PORTUGUESE Tweets for Sentiment Analysis | Kaggle. <https://www.kaggle.com/datasets/augustop/portuguese-tweets-for-sentiment-analysis>. (Accessed on 02/04/2023). Citado na página 28.

SETTLES, B. Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, 2009. Citado 2 vezes nas páginas 16 e 18.

SEUNG, H. S.; OPPER, M.; SOMPOLINSKY, H. Query by committee. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY, USA: Association for Computing Machinery, 1992. (COLT '92), p. 287–294. ISBN 089791497X. Disponível em: <https://doi.org/10.1145/130385.130417>. Citado na página 19.

SHANNON, C. E. A mathematical theory of communication. *The Bell system technical journal*, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948. Citado na página 19.

SIAMI-NAMINI, S.; TAVAKOLI, N.; NAMIN, A. S. The performance of lstm and bilstm in forecasting time series. In: IEEE. *2019 IEEE International conference on big data (Big Data)*. [S.l.], 2019. p. 3285–3292. Citado na página 20.

ZHOU, B. Named Entity Recognition (NER) using Keras LSTM & Spacy. *Medium*, Medium, jan. 2022. Disponível em: <https://zhoubeiqi.medium.com/named-entity-recognition-ner-using-keras-lstm-spacy-da3ea63d24c5>. Citado na página 36.

Apêndices