UC Aprendizagem Profunda

Trabalho prático em grupo – Módulo 1

Docente: Miguel Rocha

1. Introdução e objetivos

Objetivo. desenvolver modelos de Machine/ Deep Learning que permitam distinguir entre texto gerado por modelos ou aplicações de Inteligência Artificial (e.g. chatbots) e texto escrito por seres humanos. Deverão ser treinados datasets para língua inglesa. Os modelos deverão receber um pequeno texto e classificar numa das duas classes (ai, human).

Os grupos terão liberdade de escolher e/ou gerar os dados que quiserem usar para treinar os seus modelos, dados os recursos que tiverem disponíveis.

Cada grupo deverá criar um repositório no github que possa conter todo o código desenvolvido, bem como os respetivos dados e os pipelines de análise de dados e validação e teste dos modelos criados em notebooks. O github deve ter na sua primeira página a constituição dos grupos e um pequeno resumo da organização do mesmo para permitir a sua melhor navegação.

2. Tarefas a desenvolver

O trabalho será organizado em várias tarefas que se detalham abaixo.

Tarefa 1. Construção dos datasets

Nesta fase, os grupos deverão colecionar dados que possam ser usados no treino dos seus modelos. Poderão usar dados de datasets existentes em vários repositórios. Alguns datasets existentes em língua inglesa são os seguintes:

- https://huggingface.co/datasets/artem9k/ai-text-detection-pile
- https://huggingface.co/datasets/dmitva/human_ai_generated_text
- https://huggingface.co/datasets/artem9k/ai-text-detection-pile
- https://huggingface.co/datasets/NicolaiSivesind/human-vs-machine
- https://www.kaggle.com/datasets/prajwaldongre/llm-detect-ai-generated-vs-student-generated-text
- https://www.kaggle.com/datasets/heleneeriksen/gpt-vs-human-a-corpus-of-research-abstracts
- https://github.com/LorenzM97/human-AI-generatedTextCorpus

Diversos outros existem e podem ser igualmente usados se considerados úteis.

Os grupos poderão ainda optar por gerar os seus próprios conjuntos de dados, quer por procura em fontes relevantes, quer por uso de APIs dos chatbots atuais, entre outros.

Tarefa 2: modelos com implementação própria

Os grupos deverão implementar os seus próprios modelos com base em código próprio escrito de raiz ou adaptado das aulas. Não poderão usar bibliotecas de machine ou deep learning (e.g. scikit-learn, tensorflow/keras, pytorch, ou outros). Os modelos devem ser implementados usando apenas numpy, podendo ser usado como base o código dado na aula 3. Deverão implementar, no mínimo, Deep Neural Networks (DNNs) e Recurrent Neural Networks (RNNs) simples, acrescido de um modelo de base de regressão logística. As DNNs deverão incorporar melhorias ao código dado nas funções de ativação, nos algoritmos de treino e/ ou no tratamento de overfitting (e.g. regularização ou Dropout, early stopping).

Estes modelos deverão ser treinados com os dados recolhidos na tarefa 1, ou parte destes se não for possível usar todos, e recorrendo a estratégias de gerar features sobre os textos que conduzam a datasets tabulares (para as DNNs).

Tarefa 3: modelos com implementação em Tensorflow

Deverão ser implementados e testados na tarefa acima descrita vários tipos de modelos usando o Tensorflow/ Keras. Entre as alternativas a implementar sugerem-se:

- Deep Neural Networks (DNNs), aplicadas a datasets tabulares; as features a gerar poderão ser derivadas das palavras/ tokens contidas nos textos, e outras variantes de NLP clássico;
- Embeddings, que podem ser treinados de novo ou considerado o uso de embeddings pré-treinados; estes podem alimentar DNNs ou outros modelos;
- Recurrent Neural Networks (RNNs) simples e RNNs com memória de longo prazo (LSTMs, GRUs) aplicadas diretamente aos textos após pré-processamento ou a embeddings;
- Transformers que podem ser treinados de raiz, ou fazer uso de modelos pré-treinados (e.g. BERT e outros modelos da família) para gerar representações adequadas que possam alimentar outros tipos de modelos;
- Outras abordagens que o grupo considere de interesse, podendo por exemplo ser usados Large Language Models (LLMs) diretamente com estratégias de zero-shot, one-shot ou outras, bem como estratégias de prompt engineering ou RAG.

Estes modelos deverão ser criados, treinados e validados em Notebooks onde se descrevam todas as opções metodológicas tomadas, mostrem e comentem os resultados obtidos.

Tarefa 4: avaliação dos modelos

Os grupos deverão criar os seus próprios datasets, dividindo-os nas diversas partições (treino, teste, validação) que permitam avaliar e comparar os vários modelos de forma não enviesada.

Por outro lado, o docente disponibilizará datasets que serão usados para validação externa para poder avaliar os modelos gerados por cada grupo e gerar rankings. No final do trabalho, o docente testará ainda os modelos em datasets de teste independentes gerados de forma análoga aos datasets de validação.

Assim, cada grupo submeterá os seus modelos em 3 fases (modelos implementados de raiz, 2 vezes modelos implementados no tensorflow). Em cada uma destas submissões o docente correrá contra o dataset de validação e gerará um ranking com os resultados, podendo no final de cada fase tornar públicos parte dos dados.

Tarefa 5: relatório

O grupo deverá resumir o trabalho realizado num relatório, sob a forma de um artigo no máximo com 10 páginas, onde colocará: introdução/ contexto breve; metodologias para criação do dataset; breve descrição da implementação dos modelos de raiz; modelos de deep learning e implementação em tensorflow; resultados obtidos; conclusões. Este será dado em formato PDF e colocado no github.

Este relatório será complementado pelo código da implementação dos modelos em numpy e pelos notebooks detalhados para treino e validação dos modelos de DL.

As submissões dos modelos serão realizadas pela implementação de um notebook que leia o dataset no formato dado e corra o modelo criado. Os modelos deverão estar treinados e guardados num ficheiro. O notebook será colocado no github do grupo até final do dia indicado e corrido pelo docente no dia seguinte sobre os dados.

3. Datas importantes

- Disponibilização de um dataset exemplo inicial por parte do docente: 6 março
- 1ª Submissão - dois melhores modelos por grupo (com implementação própria): 13 março
- 2ª Submissão - dois melhores modelos por grupo (tensorflow): 20 março
- 3ª Submissão - dois melhores modelos por grupo (tensorflow): 27 março

- Apresentação – 28 março – 15 m. por grupo; apresentar metodologias, implementação, principais resultados
- Atualização final dos repositórios, dos notebooks de análise; submissão dos modelos finais; relatório – 2 abril

4. Avaliação

O trabalho será avaliado de acordo com a qualidade do trabalho realizado nas 5 fases, bem como pela apresentação do mesmo, segundo os pesos:

- Tarefa 1 – datasets – 15%
- Tarefa 2 – modelos com implementação própria / qualidade código – 15%
- Tarefa 3 – modelos deep learning em tensorflow / metodologias e notebooks – 15%
- Tarefa 4 – resultados nos datasets validação e teste/ rankings dos modelos – 15%
- Tarefa 5 – relatório – 15%
- Apresentação – 25%

Estes pesos podem ser adaptados ao longo do trabalho sem prejudicar nenhum dos grupos.