Modelos de deteção de textos escritos por AI

APRENDIZAGEM PROFUNDA - TP

José Costa PG55970
Pedro Azevedo PG7897
Ricardo Jesus PG57898
Rui Pinto PG56010

Construção dos Datasets

Primeira fase: uso de datasets sugeridos no enunciado (HuggingFace, Kaggle, etc.)

Fases seguintes: geração de dataset próprio

API da Wikipédia para os dados **humanos**
Textos com LLMs (chatGPT) para os dados **AI**

Pré-processamento:

Remoção de valores nulos e textos duplicados
Tokenização com Tokenizer do Keras
Divisão estratificada dos dados em treino, validação e teste

Modelos com NumPy

Modelos implementados

- Logistic Regression
- Deep Neural Network (DNN)
- Recurrent Neural Network (RNN)

Recurrent Neural Network (NumPy)

- Entrada: matriz de embeddings
- Camada recorrente: 64 neurônios
- Dropout: 50%
- Otimização: Adam
- LR: 0.0005
- Batch: 16
- Epochs: 15

Deep Neural Network (NumPy)

- Primeira camada igual ao input
- Camada oculta: 32 neurônios + ReLU + L2 (0.05) + Dropout (50%)
- Segunda camada: 16 neurônios (semelhante à 1ª)
- Saída: sigmoide
- Otimização:
  - Focal Loss (foco em accuracy)
  - SGD com momentum 0.9
  - LR = 0.0001
  - Batch size: 128
  - Early Stopping + LR Scheduler (decay 0.2)
  - Mixup Alpha: 0.2

Modelos com TensorFlow

Modelos implementados

- DNN
- RNN (Bidirecioanal)
- CNN
- LSTM
- GRU
- Ensemble

Pré-processamento

- Tokenizer do Keras

DNN (TensorFlow)

- Entrada: features tabulares (Input(shape=X_train.shape[1]))
- 3 camadas densas (128 → 64 → 32) com L2, BatchNorm, Leaky ReLU e Dropout (0.5)
- Saída: 1 neurónio sigmoide
- Otimizador: Adam (lr=0.0001), loss: binary_crossentropy

CNN (TensorFlow)

- Embedding (128, vocab 30k, input len 200)
- 2 camadas Conv1D (64 e 32 filtros, kernel 5) + MaxPooling + Dropout (0.4)
- Flatten + BatchNormalization
- Densas (64 e 32 neurónios, ReLU, L2) + Dropout (0.5)
- Saída: 1 neurónio sigmoide
- Adam (lr=0.0002, weight_decay=0.01) + binary_crossentropy

RNN (TensorFlow)

- Embedding (128, vocab 30k, input len 200)
- 2 camadas Bidirectional LSTM (64 e 32 unidades) com dropout e recurrent dropout de 30%
- BatchNormalization
- Densa (32 neurónios, ReLU, L2) + Dropout (0.5)
- Saída: 1 neurónio sigmoide (L2)
- Otimizador: Adam (lr=0.001) + binary_crossentropy

GRU (TensorFlow)

- Embedding (64, vocab 10k, input len 100)
- GRU (32) com dropout e recurrent dropout de 30%
- BatchNormalization + camada densa (ReLU, L2)
- Dropout (50%) + saída sigmoide (L2)
- Adam (lr=0.001) + binary_crossentropy

Ensemble (TensorFlow)

- BERT pré-treinado (bert-base-uncased) + fine-tuning (3 epochs)
- LSTM com embedding (300) + 128 unidades
- CNN com Conv1D (128 filtros) + GlobalMaxPooling
- Cada modelo gera predições sobre os mesmos dados
- Combinação via meta-modelo (Logistic Regression)
- Meta-modelo treinado com as predições (stacking)

Avaliação dos modelos

Modelo	Hyperparametos	Accuracy
RNN	epochs=50, batch_size=32, learning_rate=0.001, dropout=0.3	60
DNN	epochs=200, batch_size=32, learning_rate=0.0001	47
Gated Recurrent Unit (GRU)	epochs=30, batch_size=16, learning_rate=0.001, dropout=0.3	53
lstm	**epochs=20, batch_size=32, learning_rate=0.0001, dropout=0.5**	**73**
LMM	**Gemini, Retrieval-Augmented Generation (RAG)**	**67**
BiRnn	epochs=50, batch_size=32, learning_rate=0.001, dropout=0.3	50
CNN	epochs=30, batch_size=64, learning_rate=0.0008	46
Bert/lstm/cnn	epochs=3, batch_size=2	51

P R O M P T E N G I N E E R I N G

Zero-shot

- Pergunta direta ao modelo, sem exemplos prévios.

Few-show

- O modelo analisa vários exemplos antes de gerar a resposta.

One-shot

- O modelo recebe um único exemplo antes de responder.

RAG

- Modelo gera respostas baseadas em informações recuperadas de documentos externos.

Conclusões e Trabalho Futuro

- Implementação manual com NumPy fortaleceu conceitos
- Generalização com validação + early stopping
- Testes com diversas arquiteturas: DNN, RNN, LSTM, GRU
- Melhorias futuras:
  - LSTM/GRU mais profundos
  - Transformers (BERT, etc.)
  - Embeddings mais ricos e contextualizados

Modelos de deteção de textos escritos por AI

APRENDIZAGEM PROFUNDA - TP

José Costa PG55970
Pedro Azevedo PG7897
Ricardo Jesus PG57898
Rui Pinto PG56010