

Trabalho de Redes Neurais Artificiais

Aluno: Elcio Fagundes Boin

Links:

Projeto no Github: https://github.com/elciofagundesboin/trabalho_ia

Pasta com as músicas usadas na execução e o vídeo de apresentação:

https://drive.google.com/drive/folders/1Xm1kEoPf7b7ZA1pG_u9wmhLSbmD7qTiD?usp=sharing

Questão de Pesquisa Escolhida:

“Qual é a diferença na eficiência do modelo considerando amostras de um mesmo áudio em qualidades (taxas de bits) ou formatos de arquivo distintos?”

O trabalho foi desenvolvido buscando calcular a taxa de erro de palavras no resultado da transcrição de músicas com diferentes taxas de bits por segundo (bitrates). Para este, foram usadas 5 amostras (músicas), cada uma com 5 bitrates distintos: 64, 128, 192, 256 e 320. Para cada música, em cada taxa de bits, foi processado em todos os modelos disponíveis no modelo Whisper: tiny, base, small, medium, e large.

Cálculo do WER

A Taxa de Erro por Palavra, ou Word Error Rate (WER) pode ser calculada usando a fórmula:

$$\text{WER} = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

Onde:

- S é o número de substituições,
- D é o número de deleções,
- I é o número de inserções,
- C é o número de palavras corretas,
- N é o número de palavras na referência (N=S+D+C)

Para o trabalho, foi usada uma biblioteca pronta para este cálculo: Jiwer (<https://pypi.org/project/jiwer/1.3/>), que calcula a taxa de erro dadas duas entradas de texto.

Execução (resumo)

1 - Processar as músicas com o Wisper e salvar cada resultado em um arquivo:

Inicialmente o programa processa todas as músicas desejadas, informadas em um *array*, previamente salvas no diretório *"audio/[bitrate]/Nome da Música.mp3"*. Nesta etapa, cada música será processada com 5 modos disponíveis no modelo Wisper (tiny, base, small, medium, large), em todas as 5 taxas de bits escolhidas (64, 128, 192, 256, 320). Cada processamento resultará em um arquivo de texto com a transcrição do modelo, e será salvo no diretório *"saída_wer/"*.

Durante a execução do processamento do Wisper, para cada caso, é salvo o tempo de execução, em segundos, de cada uma das execuções, e salvas em um arquivo individual (para cada música) no diretório *"resultado/tempos/"*.

2 - Comparar cada resultado obtido com a letra original:

Após obter todos os resultados do passo anterior, o programa chama a função que irá calcular a taxa de erro (WER) de cada música, comparando o resultado transcrito com a letra oficial da música, previamente salva no diretório *"letras/Nome da Música – Letra.txt"*. Observação: para esta comparação, o único tratamento feito no arquivo com a letra original, é a remoção de 'espaços em branco', tais como quebras de linha ou tabulações; fazemos isso através do parâmetro `RemoveWhiteSpace` do `Jiwer` (conforme a documentação da biblioteca). Os resultados da comparação de cada música são salvos em arquivos individuais, no diretório *"resultado/wer/"*.

3 - Fazer uma média dos valores

Agora com os valores do WER e dos Tempos de execução salvos, o programa calcula a média dos valores de todas as músicas selecionadas, para cada um dos casos, e salva os resultados das médias em dois arquivos, um com as médias do WER (*"resultados/Media-wer.txt"*), e outro com as médias dos tempos (*"resultados/Media-tempos.txt"*).

4 - Com as médias, montar um gráfico

Com os arquivos de médias montados e salvos, o programa faz a leitura destes e exibe na tela um gráfico para cada caso (WER e Tempos). Com este gráfico é possível fazer uma análise visual dos resultados obtidos.

5 - Salvar os resultados em um arquivo final

Ao final, o programa chama uma função que lê todos os arquivos de resultados de cada música, e também os de média, e gera um arquivo de texto com os resultados para leitura. Este arquivo chama-se *"Resultado Final.txt"* e é salvo na pasta raiz do projeto.

Músicas:

Bohemian Rhapsody: https://youtube.com/watch?v=bSnIKI_PoQU

Counting Stars: https://youtube.com/watch?v=_GWKkqNoyEA

The Sound of Silence: <https://youtube.com/watch?v=bjkPQYWNTlg>

Wellerman: <https://youtube.com/watch?v=DPe2diiGJRE>

Summertime Sadness: https://youtube.com/watch?v=fRQg_9uIGuI

Bibliotecas usadas no projeto:

Whisper: <https://github.com/openai/whisper>

Jiwer: <https://github.com/jitsi/jiwer>

Matplotlib: <https://matplotlib.org/>