fpcc2-reprodutibilidade

Matheus Lisboa Oliveira dos Santos

July 2024

1 Introdução

O artigo descreve uma abordagem para a geração automática de questões utilizando modelos de linguagem. Foi avaliado o desempenho do modelo T5, que é composto de uma arquitetura de codificador-decodificador, baseado em *transformers*. Para a avaliação do modelo foi utilizado o conjunto de dados SQuAD[1], a base é composta por sentenças de texto, e perguntas sobre essas sentenças. Durante o treinamento era fornecido para o codificador a sentença, e para o decodificador a pergunta alvo; com isso, o modelo conseguiu aprender a extrair questões sobre um determinado texto alvo.

O trabalho foi escolhido por se alinhar com a proposta do mestrado, que também trata de geração de questões. Com isso poderemos ter uma visão de como estão sendo implementados os estudos da área, e também como disponibilizar os artefatos de maneira que seja fácil para outras pessoas replicarem os resultados.

2 Metodologia

A busca de artigos se deu pela plataforma *papers-with-code*[2], onde é possível pesquisar por uma área de pesquisa em ciência da computação, e são retornados artigos que contém código associado, o que favorece a reprodutibilidade de artigos. Foram pesquisadas palavras-chave, como: question generation, automatic question generation, generated questions; e em seguida filtrados artigos que tivessem uma abordagem mais parecida com a proposta do mestrado.

Com o artigo e a base de código em mãos, foi conduzido um estudo de como foi implementada a solução do autor, a fim de podermos replicar. Foi adotado o princípio de replicação, onde verificaremos se a abordagem e o conjunto de dados conseguem chegar no mesmo resultado reportado.

[1] https://rajpurkar.github.io/SQuAD-explorer/
[2] https://paperswithcode.com/

	Autor	Esse Experimento
BLEU-4	18.5	19.0
METEOR	24.9	25.3
ROUGE-L	40.1	40.8

Table 1: Contendo os resultados reportados pelo autor e obtidos na reprodução

3 Reprodução

Foi escolhida uma máquina disponível no laboratório, equipada uma Nvidia RTX 2080, com 12GB de Vram, que se mostrou capaz de executar o treinamento do modelo. Contudo, a replicação não se deu de maneira idêntica, visto que o autor utilizou o tamanho de *batch* de 32, e a memória da GPU não foi o suficiente, então foi diminuído para 28. O treinamento do modelo levou cerca de 3 horas.

Durante o processo de execução foram encontrados diversos empecilhos, visto que a base de código foi publicada há 4 anos. Destacam-se problemas com versões do Python, utilização de bibliotecas sem a especificação de versão e também não foi dito qual hardware foi utilizado para execução dos modelos.

Uma vez que o treinamento e avaliação terminou, conseguimos comparar os resultados, que estão disponíveis na Tabela 1. Os resultados foram um pouco superiores aos reportados, mas podemos atribuir esse comportamento a natureza não-determinística do treinamento de redes neurais; e também a mudança no parâmetro de *batch-size*.

4 Conclusões

Durante o processo de replicação foram encontrados obstáculos para reproduzir os resultados, contudo, não foram impossíveis de se resolver. É importante notar também que o autor teve muito cuidado para disponibilizar todos os artefatos utilizados na pesquisa, a documentação do repositório era bastante clara, e fácil de seguir. Utilizaremos o que foi aprendido na disponibilização dos artefatos gerados no mestrado.

Também foram verificados os resultados reportados pelo autor, atestando a reprodutibilidade do artigo.