Uma Abordagem Prática do Transformer

José A. A. Oliveira, José E. S. Junior, Vinicius J. M. T. B. Filho

Departamento de Engenharia Elétrica e de Computação – Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal 1.524 – 59.078-900 – Natal – RN – Brasil

Departamento de Informática e Matemática Aplicada – Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal 1.524 – 59.078-900 – Natal – RN – Brasil

augusto.oliveira.099@ufrn.edu.br, jose.edivandro.ja@gmail.com, vjmtbf@msn.com

Abstract. *This work realizes a study on the transformer architecture in a practical application with sentences, popular Brazilian phrases to verify the relationships between the inputs and predictions.*

Resumo. *Este trabalho realiza um estudo sobre a arquitetura tranformer em uma aplicação prática com frases, ditados populares brasileiros, com o intuito de se verificar as relações entre as entradas e as predições.*

1. Introdução

A inteligência artificial é um campo em crescimento, principalmente com o aumento computacional em processamento e memória, bem como das informações.

Nos últimos anos, pôde-se verificar a aplicação de lei de Moore que era a capacidade de dobrar o poder computacional a cada dois anos com a diminuição dos custos de produção. Para fugir do problema de temperatura, os processadores foram desenvolvidos para o uso de processamento paralelo, aumentando assim a capacidade computacional.

Também houve diminuição dos custos por armazenamento, permitindo que as empresas pudessem armazenar, cada vez mais, maiores quantidades de informações dando origem ao *bigdata, datalakes e datamarts* que são grandes volumes de dados.

Esse aumento na capacidade de processamento, armazenamento e memória foram os responsáveis pelo surgimento das nuvens dos grandes provedores. Todo esse cenário impulsionaram o uso da inteligência artificial.

O estudo da inteligência artificial teve seu início em 1943 com a criação de um modelo computacional para redes neurais pelo neurofisiologista Warren McCulloch e o matemático Walter Pitts apresentando o neurônio de McCulloch e Pitts. Em 2017, pesquisadores do Google propuseram uma arquitetura para se tratar textos e suas relações. Essa pesquisa se materializou no artigo *Attention is all you need* [Vaswani, A. et al. 2017] e impulsionou a capacidade gerativa das inteligências artificiais.

"O aprendizado profundo moderno oferece uma estrutura muito poderosa para o aprendizado supervisionado. Ao adicionar mais camadas e mais unidades dentro de uma camada, uma rede profunda pode representar funções de complexidade crescente. A

maioria das tarefas que consistem em mapear um vetor de entrada para um vetor de saída, e que são fáceis para uma pessoa fazer rapidamente, podem ser realizadas através do aprendizado profundo, desde que haja modelos suficientemente grandes e grandes conjuntos de dados de exemplos de treinamento rotulados" [Goodfellow I., Bengio Y. e Courville A. 2016].

O *Transformer* é um modelo de aprendizado profundo e várias pesquisas estão fazendo uso para resolver problemas de sequenciamento, principalmente pelo processamento paralelo em lotes. O artigo "A Transformer-based Approach for Translating Natural Language to Bash Commands" [Fu Q. et al. 2021] mostra o ganho de acurácia em relação ao estado da arte atual com o uso do *transformer*. O artigo "Treinamento e análise de um modelo de tradução automática baseado em Transformer" [Pimentel C.H.M.; Pires T.B. 2024] mostra um bom desempenho na tradução com o uso do *transformer*.

O artigo "Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages" [Gowtham R. et al. 2021], que traz um estudo sobre as línguas da índia, foi o propulsor para a implementação da arquitetura transformer para gerar um tradutor. Esses artigos foram impulsionadores para este trabalho.

Este estudo propõe fazer uso dessa arquitetura para realizar inferências em frases. A Seção 2 apresenta um estudo da arquitetura *transformer*, a seção 3 apresenta a metodologia para este trabalho, a seção 4 apresenta os resultados e a seção 5 apresenta a conclusão e próximos passos.

2. *Transformer*

O ponto fundamental do *transformer* é que existe uma relação entre os itens que compõem o modelo. "Uma grande vantagem dos transformers é que a transferência de aprendizado é muito eficaz, de modo que um modelo transformer pode ser treinado em um grande conjunto de dados e, em seguida, o modelo treinado pode ser aplicado a muitas tarefas subsequentes usando alguma forma de ajuste fino" [Bishop M.C. e Bishop H. 2023]. A Figura 1 mostra, na frase, a relação das palavras com a palavra *blank,* dando sentido a ela. No artigo *Attention is all you need*, os autores informam que "Transformer é o primeiro modelo de tradução que depende inteiramente da autoatenção para calcular representações de sua entrada e saída sem usar RNNs alinhadas por sequência ou convolução" [Vaswani, A. et al. 2017].

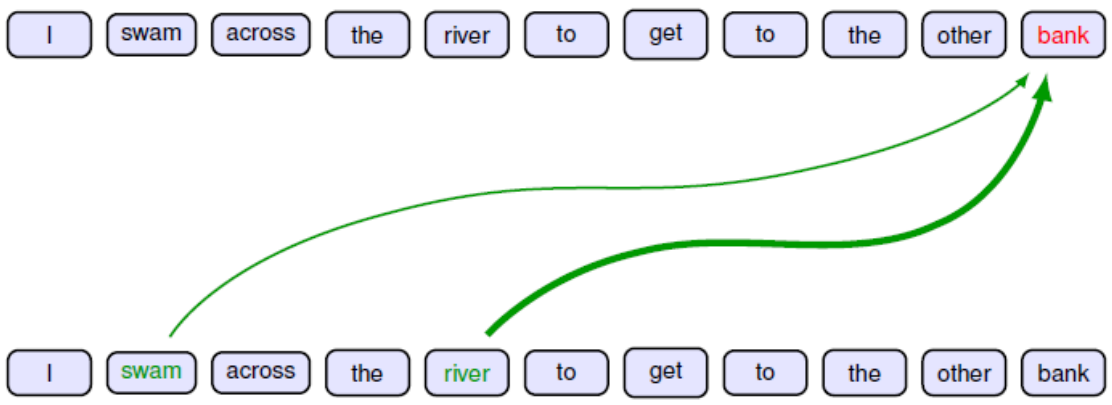

Figura 1. Relação entre as palavras dando sentido a palavra *blank*

A codificação posicional é uma técnica da arquitetura para permitir informar a posição de cada palavra na sequencia. Utiliza as funções seno e cosseno quando o índice na sequência for par e ímpar, respectivamente. A figura 2 mostra a fórmula.

$$r_{ni} = \begin{cases} \sin\left(\frac{n}{L^{i/D}}\right), \\ \cos\left(\frac{n}{L^{(i-1)/D}}\right), \end{cases}$$

Figura 2. Funções para codificações posicionais

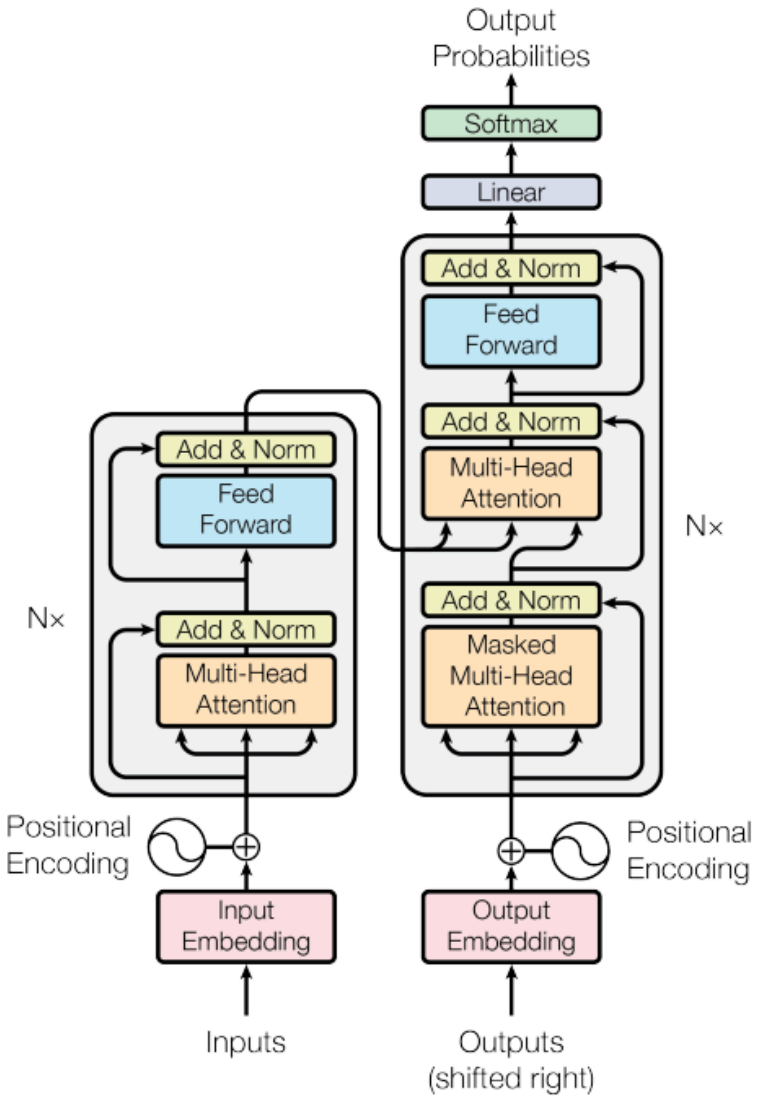

Figura 3. Arquitetura do *Transformer*

A figura 3 apresenta a arquitetura do *Transformer*. O mecanismo de atenção é o responsável por criar a relação entre as palavras permitindo a criação de contextos. O *Multi-head Attention* é a capacidade de permitir várias relações, permitindo a criação de vários contextos, entre as palavras. Viabiliza o processamento paralelo da entrada em lotes. A arquitetura permite que o modelo possa ter várias camadas.

A arquitetura possui duas principais estruturas: codificador e decodificador. O codificador é o responsável pelo treinamento dos modelos. Ele utiliza as matrizes Q (representa o que se pode perguntar sobre o item), K (representa os possíveis valores do item), V (representa o que realmente se sabe sobre o item) e as matrizes com os pesos do modelo. A figura 4 mostra a utilização das matrizes.

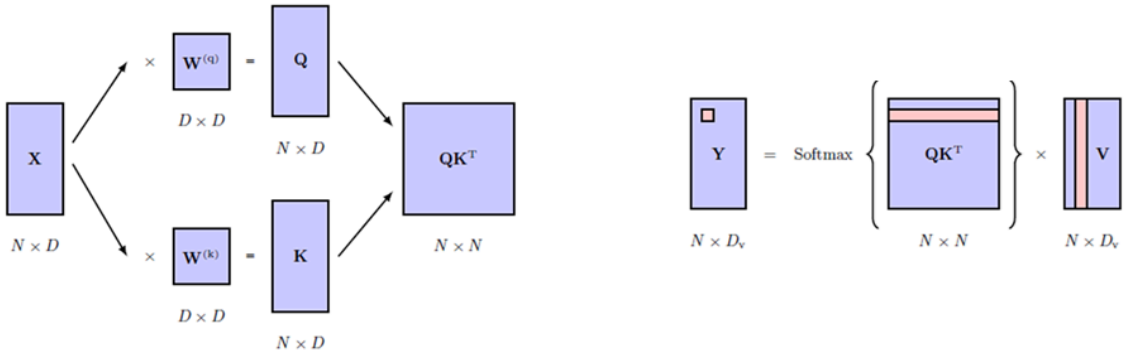

Figura 4. Utilização das matrizes da arquitetura

O decodificador é a estrutura responsável pelas inferências e gera as sequências baseadas na entrada.

3. Metodologia

A metodologia para esse trabalho consistiu na utilização da arquitetura *transformer*, baseada em caractere, para ser treinada com 10 frases de ditados populares e podermos verificar as inferências.

Fizemos o uso do *transformer* implementado na linguagem Python com 10 frases de ditado popular: Mais vale um pássaro na mão, do que dois voando; Quem com ferro fere, com ferro será ferido; Água mole em pedra dura, tanto bate até que fura; Em boca fechada, não entra mosquito; De grão em grão, a galinha enche o papo; Quem tem boca, vaia Roma; Não adianta chorar, pelo leite derramado; Quem não tem cão, caça com gato; Quem espera, sempre alcança; e Filho de peixe, peixinho é.

Os testes consistiram na variação da quantidade de camadas, épocas para se verificar a convergência bem como na variação na entrada para se verificar a predição. Para cada configuração dos parâmetros foram realizadas 10 repetições.

Para a variação da quantidade de camadas e de épocas, fizemos experimentos com os valores 1, 2, 4 e 8 para a quantidade de camadas e 10, 20, 50, 100, 200, 400, 1000, 2000, 4000 para a quantidade de épocas.

Variamos as entradas em tamanho com palavras chaves e fixamos o tamanho, modificando alguns caracteres.

4. Resultados

A tabela 1 mostra o comportamento do modelo com a variação da entrada em tamanho com palavras chaves. Como o modelo é baseado em caractere, nenhum resultado ficou satisfatório para esse teste.

Tabela 1. Resultado com a variação da entrada com palavras chaves

Entrada	Saída
Quem boca	sempre alccaça.<END>
Quem tem cão	sempre alcança.<END>
Filho peixe	sempre llcança.<END>
vale um pássaro	peixinho é.<END>
Quem com ferro	vaia Roma.<END>
água mole	sempre lccança.<END>
boca fechada	sempre alcança.<END>
de grão	sempre alcançaa.appppp.<END>
não adianta chorar	pelo leite do.<END>
espera	Não conseguiu fazer a predição para

A tabela 2 apresenta a quantidade de épocas necessárias para a convergência dentro do escopo da pesquisa e o tempo para o treinamento do modelo para cada camada.

Tabela 2. Quantidade de épocas e tempo de treinamento

Camadas	Épocas	Tempo médio de treinamento (em segundos)
1	200	20,20
2	200	39,33
4	400	151,51
8	4000	3135,28

A tabela 3 mostra as entradas originais, a entrada com variações, trocando algumas letras, e a saída que foi igual para as duas entradas em todos os experimentos com as configurações de convergências acima.

Tabela 3. Entradas e as respectivas saídas

Entrada Original	Entrada com variações	Saída
Mais vale um pássaro na mão,	Maix vate um passaro na não,	do que dois voando.
Quem com ferro fere,	Quen com fetro frre,	com ferro será ferido.

Água mole em pedra dura,	**Água nole em pedra duta,**	**tanto bate até que fura.**
Em boca fechada,	**En bola fecxada,**	**não entra mosquito**
De grão em grão,	**De grao en grão,**	**a galinha enche o papo.**
Quem tem boca,	**Quen tim bola,**	**vaia Roma.**
Não adianta chorar,	**Nao adiamta chorra,**	**pelo leite derramado.**
Quem não tem cão,	**Quen nao tem cao,**	**caça com gato.**
Quem espera,	**Quen espeta,**	**sempre alcança.**
Filho de peixe,	**Filto de peche,**	**peixinho é.**

5. Conclusão

Este trabalho propiciou entendermos o comportamento da arquitetura *transformer* baseada em caractere. Pudemos verificar uma relação direta entre a quantidade de camadas e a quantidade de épocas necessária para realizar o treinamento para o modelo convergir.

Quando realizamos o treinamento muitas camadas e poucas épocas foi possível verificar que não há a convergência e que quanto maior o número de camadas mais épocas são necessárias para que o modelo possa convergir.

Em relação a entrada de texto, pudemos verificar que com a tentativa de utilizar palavras principais de cada entrada não foi satisfatória, pois o modelo é baseado em caractere, mas que inferiu suficientemente bem quando trocamos alguns caracteres mantendo a estrutura da frase.

Como próximos passos, faremos um ajuste para que o modelo passe a utilizar palavras ao invés de caractere e repetiremos essa metodologia para verificarmos o comportamento tanto em treinamento como na inferência das respostas.

6. Referências

Vaswani, A. et al. (2017) Attention Is All You Need. Conference On Neural Information Processing Systems (NIPS), Long Beach.

Bishop M.C. e Bishop H. (2023) Deep Learning Foundations and Concepts. Editora Springer Cham, Cambridge.

Goodfellow I., Bengio Y. e Courville A. (2016) Deep Learning. Editora The MIT Press.

Fu Q. et al. (2021) A Transformer-based Approach for Translating Natural Language to Bash Commands. Conferencia 20th IEEE International Conference on Machine Learning and Applications (ICMLA).

Pimentel C.H.M.; Pires T.B. (2024) Treinamento e análise de um modelo de tradução automática baseado em Transformer. Disponível em https://doi.org/10.1590/1983-3652.2024.49118, acessado em 08/07/2025.

Gowtham R. et al. (2021) Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. Disponível em https://doi.org/10.48550/arXiv.2104.05596, acessado em 10/07/2025.