Transformers e Midjourney AI: Da teoria à prática

Universidade do Minho, Largo do Paço 4704-553 Braga, PT

Abstract. Nos dias de hoje, grande parte da população já utilizou ou conhece programas como o ChatGPT, Gemini, BERT, ou MidJourney. Todos estes sistemas têm uma base científica por trás, nomeadamente, redes neuronais artificiais.
Neste trabalho serão abordados dois temas com níveis de abstração distintos mas elucidativos do que estas tecnologias nos têm para oferecer.
O primeiro, uma arquitetura de Redes Neuronais, os Transformers, a qual surgiu em 2017 e revolucionou o mundo das Redes Neuronais. Discutiremos a sua história, as suas inúmeras aplicações e, com grande detalhe, o funcionamento por trás desta inovadora arquitetura.
O segundo, o MidJourney, o qual surgiu em 2022, é um modelo de inteligência artificial capaz de gerar imagens através de *prompts* de texto. Explicaremos como este utiliza modelos de Redes Neuronais para produzir estas imagens.

Keywords: Midjourney AI · Redes Neuronais · Transformers.

1 Introdução

No âmbito da Unidade curricular **Classificadores e Sistemas Conexionistas**, do perfil de Machine Learning do Mestrado em Matemática e Computação da Universidade do Minho, foi realizado um trabalho de investigação sobre Redes Neuronais. Mais especificamente, sobre a arquitetura **Transformers** e o **Midjourney AI**.

2 Transformers

2.1 Contexto Cultural vs Contexto Científico

No contexto cinematográfico e cultural, o nome **Transformers** é conhecido por grande parte da comunidade. No contexto do Deep Learning, a sua popularidade acaba por ser semelhante, porém, no que toca ao conteúdo há difereças significativas a ter em conta.

Os Transformers são uma arquitetura de Redes Neuronais que se tornou bastante popular com o lançamento de programas com modelos baseados nessa arquitetura, por exemplo, o ChatGPT, Bard, BERT ou o Midjourney. A arquitetura transformer é usada essencialmente para resolver problemas de processamento de linguagem natural mas também é aplicado a áreas como *Computer Vision* ou no desenvolvimento de medicamentos.

2.2 Contextualização histórica

Os primeiros conceitos teóricos de redes neuronais apareceram na década de 40 através do trabalho de Warren McCulloch e Walter Pitts, os quais proposeram um modelo computacional de neurónios no artigo intitulado *"A Logical Calculus of Ideas Immanent in Nervous Activity,"*.

Já na década de 50, Frank Rosenblatt criou uma rede neuronal com uma camada, o *Perceptron*, o qual era capaz de aprender padrões simples.

Na década de 70 foi desenvolvido o algoritmo de *backpropagation*, o qual viria a servir de base para o treino de redes neuronais com múltiplas camadas.

As redes neuronais recurrentes (RNNs) surgiram pela primeira vez em 1982 através da *Hopfield Network*.

Em 1989 foram introduzidas as redes neuronais convolucionais (CNNs), as quais obtiveram níveis de performance nunca antes obtidos, até à altura, para o reconhecimento de digitos escritos à mão.

Na década de 90, Sepp Hochreiter e Jürgen Schmidhuber desenvolveram redes neuronais *Long Short-Term Memory* (LSTM), as quais resolveram o problema da dissipação do gradiente nas RNNs (problema que ocorre durante o treino de uma rede neuronal, quando os gradientes propagados são multiplicados por valores menores que um a cada camada da rede que atravessam, levando a que quando ocorre a *backpropagation*, os valores que regressam às camadas iniciais são extremamente pequenos ou mesmo inexistentes).

AlexNet, uma *Deep Convolutional Neural Netwok*, é criada em 2012. Esta leva a uma melhoria significativa na precisão na classificação de imagens.

Em 2014 são criadas as *Generative Adversarial Networks* (GANs).

Entre estes e muitos outros avanços, aqui não mencionados, que ocorreram ao longo dos anos, chegamos então a 2017. Ano da publicação do artigo *"Attention is All You Need"* de Vaswani et al. Neste artigo foi proposto o *encoder-decoder Transformer* original, o qual, ao invés de camadas recurrentes ou convoluções, utilizadas pelos RNNs e CNNs, respetivamente, apresentava um mecanismo de *self-attention*. Este veio revolucionar por completo o processamento de linguagem natural, obtendo resultados de excelência em diversas tarefas como traduções e geração de texto.

3 Arquitetura do modelo

Nesta secção vamos descrever como funciona uma arquitetura transformer descendo gradualmente o nível de abstração dos conceitos apresentados.

Até ao ano de 2017, os modelos tradicionais usados para realizar tradução e geração de texto baseavam-se em *Recurrent Neural Networks* e *Convolutional Neural Networks* que continham um *encoder* e um *decoder*.

Estes modelos apresentavam várias lacunas, sendo eles de natureza sequencial, não conseguiam identificar aspetos fundamentais na transcrição de texto, como por exemplo, ordem e contexto. Com isto, no ano de 2017 foi lançado por investigadores da Google um paper chamado "*Attention is all you need*" [5] onde

enunciam uma arquitetura que acabou por impactar a forma como processamos informação, os **Transformers**.

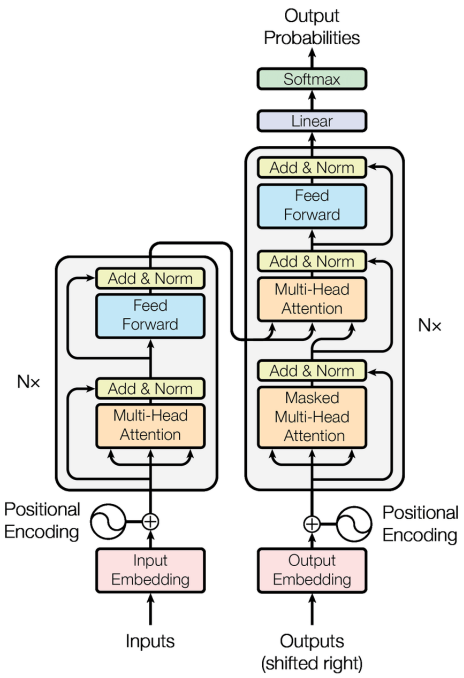

Fig. 1. Transformer - Arquitetura do modelo

Uma arquitetura Transformer, como podemos ver na figura, começa com uma fase de vetorização do input recebido (normalmente strings que são passadas para vetores) chamada de **Input Embedding**. Após a preparação do input, é realizado **Positional Encoding** que tem como objetivo dar contexto à rede acerca da posição das palavras na frase. Após a rede ter contexto do seu input, entramos no **Encoder** e, posteriormente, no **Decoder** que serão explicados em maior detalhe mais para a frente. Para finalizar, é feito um **Softmax** e a palavra gerada é a que tiver uma maior probabilidade.

Algo a ter em conta é o facto do modelo ser auto-regressivo, o que significa que cada output que é gerado tem como base acontecimentos passados.

3.1 Input Embedding

A ideia subjacente ao conceito de *Input Embedding* é a de vetorizar o nosso input que geralmente são strings.

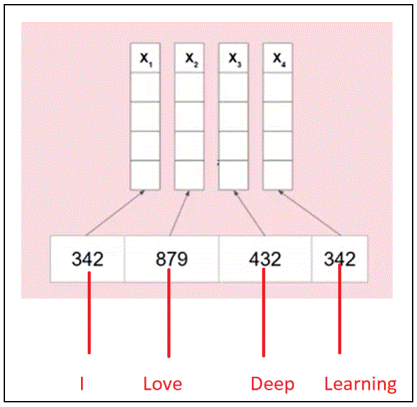

Fig. 2. Input Embedding

Tomemos o exemplo da figura acima[6], dado o input "I Love Deep Learning", partimos a frase em 4 tokens e atribuimos um valor, inicialmente aleatório, a cada palavra. De notar que tokens iguais em frases diferentes contêm o mesmo valor.

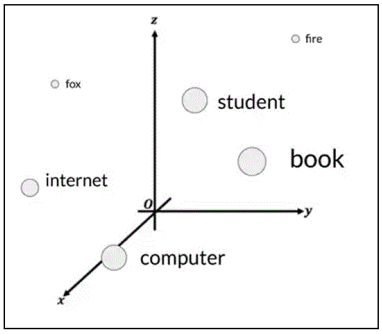

Fig. 3. Projeção no plano

O objetivo em partir a frase em tokens e vetoriza-los é, em primeiro lugar, as redes neuronais não aceitam strings, em segundo, a partir do momento em que conseguimos representar palavras como números, é possível projetar o input para um plano onde facilmente é visível a relação entre as palavras, como podemos ver na imagem acima. Em vários casos, o objetivo será minimizar a distância entre palavras que estão frequentemente relacionadas em várias frases.

3.2 Positional Encoding

A técnica de *Positional Encoding* surge dada a necessidade de ter a noção de ordem na sequência das palavras, quer seja ela relativa ou absoluta. Para isso, os autores do paper "*Attention is all you need*" sugeriram uma abordagem brilhante.

Para relacionar o token com a sua posição na frase foi usada a seguinte função:[5]

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right)$$

onde *pos* é o índice da palavra na frase, d_{model} é o tamanho do vetor que consideramos para os inputs (neste modelo é 512).

Dado isto, conseguimos construir a seguinte matriz:

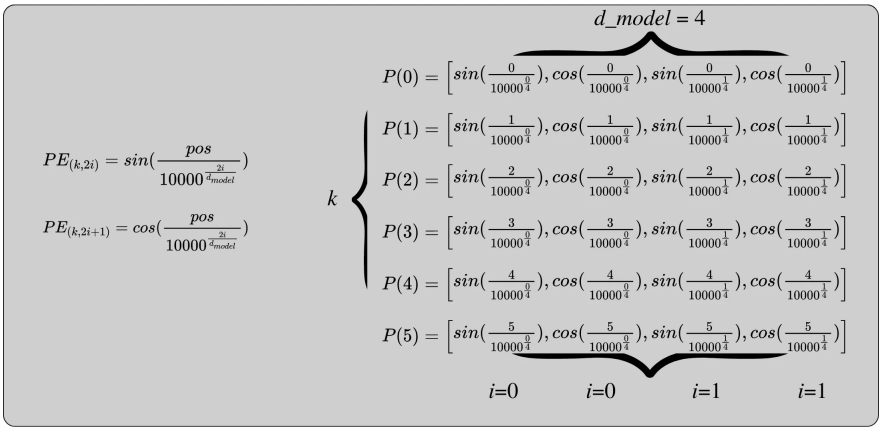

Fig. 4. Positional Encoding

Com isto, obtemos para cada palavra uma noção de ordem e posição na palavra.

Tendo os vetores com informação sobre a posição das palavras, basta-nos adicionar esse vetor ao vetor criado na fase de *Input Embedding* e temos assim um resultado que contém informação sobre a palavra e a sua posição na frase.

3.3 Attention Mechanism

Em linguagem natural, os seres humanos utilizam algo muito semelhante ao mecanismo de atenção implementado nos transformers. Por exemplo, se alguém num diálogo disser "Ontem fui ao **banco**. Ao chegar entrei e sentei-me num **banco**". Nesta frase há duas palavras iguais que têm significados diferentes. Ora, para os seres humanos isto não é um problema visto que o verbo "ir" está associado à entidade "banco" e o verbo "sentar" está associado ao objeto "banco". No modelo transformer tentamos simular este tipo de mecanismo.

3.4 Multi-Head Attention

Este mecanismo é utilizado na parte inicial do *encoder* e serve para calcular a relação entre todas as palavras na frase, de forma a que o nosso modelo dê maior importância às palavras com as quais identifica uma maior relação.

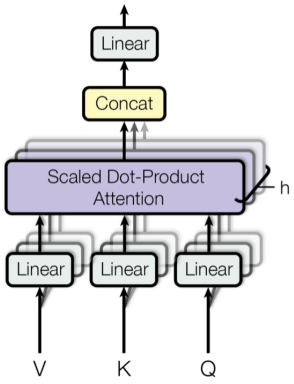

Fig. 5. Multi-Head Attention Mechanism

O método chama-se multi-head attention [5]porque é feita uma paralelização de h medidas de atenção que queremos ver nas palavras.

Na figura vemos três letras V, K e Q que correspondem a *value*, *key* e *query*, respetivamente. Estas 3 variáveis inicialmente têm o mesmo valor que é o output dado após ser realizado positional encoding.

Vamos imaginar que o nosso input é **"When you play the game of thrones"**, isto significa que vamos obter um vetor 7×1 em V, K e Q. O que fazemos para calcular a relação entre as palavras é calcular um *attention filter* que se baseia em calcular $Q \times K^T$, o que vai originar uma matriz 7x7, como podemos ver na figura abaixo.

Após termos esta matriz realizamos um softmax e multiplicamos pelo vetor *Value*. Com isto obtemos um vetor com valores que têm em conta a relação entre as palavras.

	When	you	play	the	game	of	thrones
When	89	20	41	10	55	10	59
you	20	90	81	22	70	15	72
play	41	81	95	10	90	30	92
the	10	22	10	92	88	40	89
game	55	70	90	88	98	44	87
of	10	15	30	40	44	85	59
thrones	59	72	92	90	95	59	99

Fig. 6. Attention matrix

Ora, todo este processo é realizado para as h camadas. Após ter os h resultados calculados, **concatenamos** os vetores e aplicamos uma *linear layer* para dar reshape ao resultado.

Este processo é então definido pelas seguintes funções:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O$$

$$where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Após estes procedimentos os vetores passam por uma rede *point-wise feed-forward*

3.5 Decoder

O *decoder* funciona de forma muito semelhante ao *encoder* mas com algumas diferenças. O *encoder* recebe 1 argumento que é o input do utilizador, porém, já anteriormente foi dito que uma caraterística da arquitetura *transformer* é a de olhar ao texto anteriormente gerado para gerar novas palavras.

Portanto, o *decoder* recebe 2 argumentos sendo que o primeiro é o output do *encoder* e o segundo é o texto gerado até ao momento que vai passar por camadas de *embedding*, *positional encoding* e de *attention*. Com estes dois argumentos realiza-se o mesmo processo que é feito no *encoder* com a variante de que a *key* e a *query* são o output do *encoder* e o *value* é o resultado de aplicar as transformações ao texto que já foi gerado.

Mais uma vez, no fim do processo de *decoding* é aplicada uma função *Softmax* e a palavra que tiver maior probabilidade é a que vai ser escolhida.

4 Midjourney AI

4.1 Definição e contextualização

A tecnologia **Midjourney AI** é um serviço de inteligência artifical desenvolvido pela Midjourney, Inc., um laboratório de pesquisa independente baseado em São Francisco (Califórnia). Este laboratório é constituído por uma equipa relativamente pequena, atualmente com 11 funcionários e um conjunto de consultores, autofinanciada e totalmente distribuída encontrando-se atualmente em recrutamento. Os objetivos desta equipa passam por explorar novos meios de pensamento e expandir os poderes imaginativos da espécie humana através de **inteligência artificial**, design e infraestrutura humana.

Relativamente ao serviço Midjourney AI, desenvolvido por este laboratório, é um serviço de arte de inteligência artificial que permite **gerar imagens com base em texto**. Para usar o serviço, primeiro é necessário entrar no servidor *Discord* gratuito da Midjourney. De seguida, existem vários canais de texto onde um Bot está disponível, devendo-se digitar o comando **"/imagine"** e, imediatamente a seguir a esse comando, deve ser inserida uma frase/texto. O Bot, em cerca de 1 minuto, fornecerá 4 imagens relacionadas com o texto digitado, podendo posteriormente obter-se escalas ou variações dessas imagens.

4.2 Arquitetura do modelo

Por detrás desta tecnologia está um modelo de inteligência artificial denominado de difusão estável (**Stable Diffusion** como é universalmente reconhecido) cujo principal objetivo, nesta tecnologia em específico, passa por converter texto para imagens. De notar que este tipo de modelo é versátil e poderia também ser usado para alterar imagens isto é, dado um texto de entrada e uma imagem, alterar essa imagem de forma a englobar o texto de entrada mais fidedignamente.

Nesta tecnologia e no seu respetivo modelo, existem duas principais fases desde o texto de entrada até a imagem final que é fornecida: a fase de codificação do texto (**Text Encoder**) e a fase de criação da imagem em si (**Image Generator**). Este processo encadeia-se como ilustra a imagem abaixo (de forma abstrata).

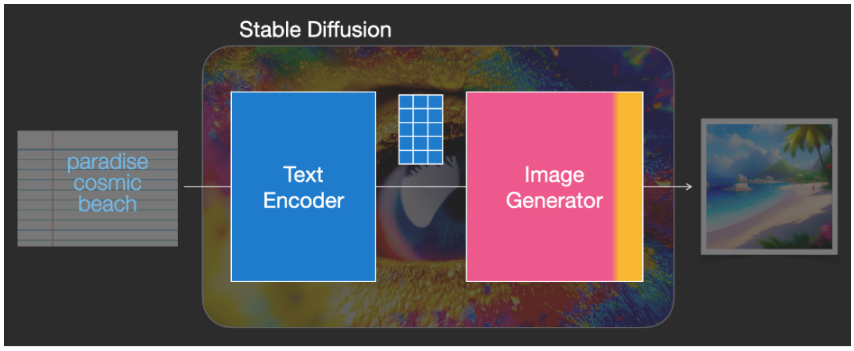

Fig. 7. Modelo abstrato

4.3 Text Encoder

O codificador de texto é um modelo especial baseado na arquitetura **Transformer** onde o objetivo passa por receber o texto de entrada e gerar uma lista com vários vetores de números que representam cada palavra/token no texto onde **cada palavra/token está associado a um vetor de números**.

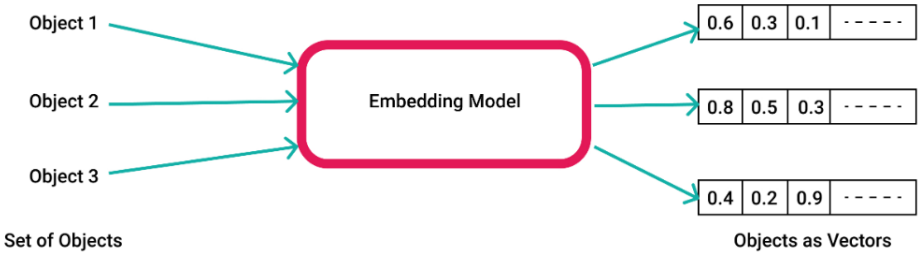

Fig. 8. Esquema representativo do Text Encoder

4.4 Image Generator

Os vetores provenientes do Text Encoder são então apresentados ao Gerador de Imagens, que é composto por outras duas componentes: a componente de criação de informações da imagem (**Image Information Creator**) e a componente de descodificação da imagem (**Image Decoder**).

4.4.1 Explicação das duas componentes e respetivo encadeamento

É nesta fase que se situa uma boa parte da explicação do bom desempenho deste modelo e este segmento trabalha sobre o espaço da informação das imagens ou espaço latente. Esta fase é composta por uma rede neuronal denominada de **UNet** e um algoritmo **Scheduler**. A palavra "**difusão**" descreve o que acontece neste componente, isto é, é o processamento passo a passo da informação onde **em cada passo existe um ganho de informação** relativamente ao anterior sendo isto responsabilidade da rede neuronal UNet que procura **prever o ruído** que existe numa matriz de informação. Posteriormente, o objetivo passa por **subtrair o ruído** para obter uma imagem mais próxima das imagens em que o modelo foi treinado, não as imagens exatas em si, mas a distruibição/arranjo dos pixéis da mesma (o céu é geralmente azul e acima do solo, as pessoas têm dois olhos, os gatos olham de uma certa maneira e têm orelhas pontiagudas). O algoritmo **Scheduler** procura **otimizar a convergência deste método** melhorando a taxa de aprendizagem em cada passo. A **criação de uma imagem visual** em si e a respetiva pintura é feita pelo próximo componente, **Image Decoder**. Todo este processo inicia-se então com uma **matriz de informação aleatória** da qual poderá ser extraído o ruído sendo depois transformada numa imagem efetivamente pelo Image Decoder e, após isso, em cada passo, é produzida outra matriz de informação que melhor se assemelha ao texto de entrada e a todas as informações visuais que o modelo captou de todas as imagens em que um modelo denominado "**CLIP**" foi treinado. Este modelo CLIP é treinado com uma base de dados grande composta por imagens e as suas legendas com o objetivo de as relacionar. A imagem abaixo ilustra, de forma abstrata, todo este processo.

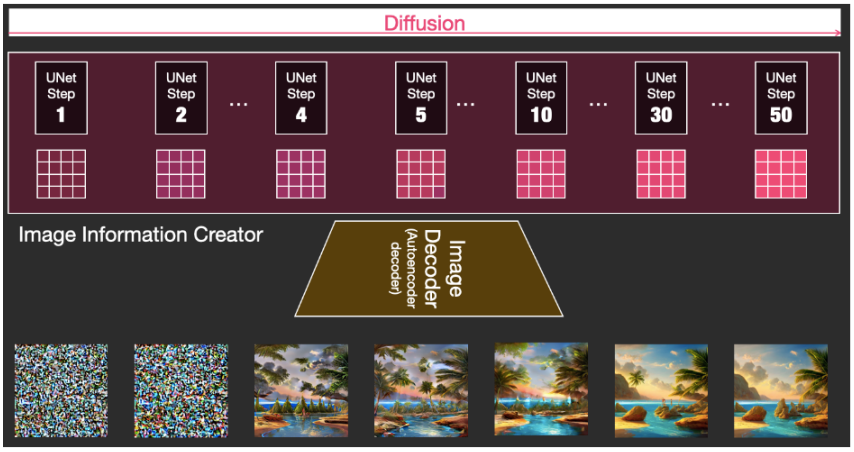

Fig. 9. Esquema representativo do processo de criação de imagens

No final, e ao fim de um **número de passos variável** no processo de difusão, espera-se ter uma imagem limpa e que caracteriza fidedignamente o texto de entrada.

4.5 Alguns exemplos da utilização da tecnologia Midjourney AI

O programa já foi usado por várias revistas/jornais conceituados pelo mundo fora. A revista The Economist criou a sua capa de junho de 2022 com base nesta tecnologia, o jornal Corriere della Sera, líder de audiência na Itália, publicou uma tira de banda desenhada criada com o Midjourney e escrita por Vanni Santoni, em agosto de 2022. Já a revista norte-americana The Atlantic, através de um de seus escritores, Charlie Warzel, usou o Midjourney para criar uma imagem de Alex Jones um norte-americano de extrema direita condenado pela justiça, o que gerou controversa por parte de artistas que sentiram que seus empregos estavam ameaçados. O programa de televisão norte-americano Last Week Tonight with John Oliver também utilizou o Midjourney assim como vários livros infantis causando bastante polémica e opiniões adversas a esta tecnologia nomeadamente por parte de artistas que viam o seu trabalho ser ameaçado e que alegavam mesmo que a tecnologia era treinada com imagens que resultaram de obras de artistas que nunca consentiram o seu uso. Seguem abaixo algumas das mais famosas imagens geradas por esta tecnologia.

Fig. 10. Papa Francisco com um "casacão"

Fig. 11. Albert Einstein

Fig. 12. Bill Gates

5 Prós e contras de cada tecnologia

Relativamente às vantagens e desvantagens da arquitetura Transformers, comparativamente a outros modelos, podemos dizer que esta apresenta uma computação altamente paralelizável como foi mostrado no processo de *Multi-Head Attention*, o que leva a um processo mais rápido/eficiente. Esta apresenta também um mecanismo de self-attention, explicado em detalhe anteriormente e que permite transmitir contexto à rede neuronal do input que estamos a receber.

Comparativamente a RNNs, os Transformers apresentam uma grande vantagem principalmente no processamento de linguagem natural visto que nas RNNs o processamento da informação é sequencial, isto é, palavra a palavra, enquanto que na arquitetura Transformer é considerado apenas partes importantes do input e tem em consideração o que vai sendo gerado para gerar novos outputs.

Todas estas vantagens vêm ao preço de um custo computacional bastante elevado e de um consumo também elevado de memória. Normalmente são também necessários um conjunto bastante extenso de dados de treino de forma a atingir resultados ótimos, o que pode causar problemas quando os dados são escassos ou o seu custo é demasiado elevado.

No que toca à tecnologia Midjourney AI a mesma é extremamente cómoda e fácil de utilizar visto que com apenas um comando com texto à escolha é possível gerar imagens rapidamente. Além disso é bastante versátil pois oferece uma variabilidade de filtros que permite a personalização das imagens após estas serem geradas. Na plataforma *Discord* existe uma comunidade para os usuários desta tecnologia o que permite que os mesmos se ajudem mutuamente e partilhem o seu trabalho. Alguns dos pontos não tão apelativos da tecnologia é o facto da mesma apenas estar disponível através da plataforma *Discord* e ser um serviço que atualmente só pode ser obtido pagando. Além disso é um modelo de código fechado o que o que dificulta a personalização e melhoria por parte de pesquisadores/desenvolvedores. Apesar disto, ambas as arquiteturas são populares dentro do mundo da IA sendo a arquitetura Transformers largamente utilizada e uma referência na área do processamento de linguagem natural. Já a Midjourney AI, para a finalidade da criação de arte visual, também é uma das referências mais utilizadas inclusivamente por jornais e revistas e várias imagens de inteligência artificial que se tornaram virais, surgiram desta tecnologia.

References

1. Midjourney Homepage, `https://www.midjourney.com/home`
2. Wikipedia, `https://pt.wikipedia.org/wiki/Midjourney`
3. Hanna, Magdy, D.: The Use of Artificial Intelligence Art Generator "Midjourney" in Artistic and Advertising Creativity, Journal of Design Sciences and Applied Arts (June 2023)
4. Allamar, J.: The Illustrated Stable Diffusion, (November 2022), `https://jalammar.github.io/illustrated-stable-diffusion/`

5. Vaswani, Ashish et al. “Attention is All you Need.” Neural Information Processing Systems (2017).
6. LinkedIn, Deep dive positional encodings transformer neural network, https://www.linkedin.com/pulse/deep-dive-positional-encodings-transformer-neural-network-ajay-taneja/?trackingId=hj8daztzsW2OxzkDYaFMtA%3D%3D