NLP 101 (4): A invenção da Attention

Muita gente acredita que a revolução da IA começou quando saiu o paper *Attention Is All You Need* em 2017, mas o mecanismo original de Attention já existia desde 2014. E ele nasceu para resolver um problema da epoca, a

2026-07-01



Figure 1: Capa do artigo

Muita gente acredita que a revolução da IA começou quando saiu o paper *Attention Is All You Need* em 2017, mas o mecanismo original de Attention já existia desde 2014. E ele nasceu para resolver um problema da epoca, a perda de informação em sequências longas.

Neste quarto artigo da série NLP 101, vamos entender como os modelos de atenção mudaram para sempre a forma como as máquinas processam linguagem.

O Gargalo das Redes Recorrentes

Na thread anterior vimos que RNNs, LSTMs e GRUs tentaram resolver o problema da memória. Elas conseguiram avanços enormes, mas ainda tinham uma limitação importante. Toda a informação precisava ser comprimida em uma representação interna, uma espécie de resumo do que foi lido até aquele momento.

Uma arquitetura popular daquela época eram os modelos Seq2Seq. Eles eram muito usados em tradução au-

© Artemisia Weyl 2024–2026 — Todos os direitos reservados.

tomática, resumo de textos e outras tarefas onde uma sequência precisava ser transformada em outra.

A arquitetura era dividida em duas partes. Um Encoder e um Decoder. O Encoder tinha uma função relativamente simples. Ler a frase inteira e transformar em uma representação matemática compacta, uma espécie de vetor-resumo.

Ex:Eu gosto de física." -> [0.12, -0.43, 0.89, ...]

Se a frase tivesse 5 palavras ou 50 palavras, o objetivo era condensar toda a informação relevante em um único vetor. Depois o Decoder recebia esse vetor e gerava a saída.

Ex: "Eu gosto de física." -> "I like physics."

Porém, na prática o Encoder estava realizando algo parecido com uma compressão de dados, e toda compressão possui limites. Imagine tentar resumir um livro inteiro em apenas um parágrafo. Você preserva as ideias principais, mas inevitavelmente perde detalhes.

Quanto maiores as sequências, maior a quantidade de informação que precisava passar pelo mesmo gargalo e quanto mais informação é comprimida, maior a chance de perder partes importantes.

Lógica da Attention

Mas e se se ele não precisasse depender exclusivamente daquele vetor- resumo? Essa pergunta deu origem ao mecanismo de Attention.

Quando você lê um texto, nem todas as palavras possuem a mesma importância, considere a frase: *"O telescópio observou a galáxia distante."* Se eu perguntar o que foi observado? Sua atenção provavelmente vai para "galáxia", porque você não distribui atenção igualmente para todas as palavras. Você procura aquelas que parecem mais relevantes para responder à pergunta.

A Attention tenta reproduzir exatamente esse comportamento. Em vez de tratar todas as informações igualmente, o modelo aprende quais partes merecem mais atenção em cada momento.

Mas como ele faz isso?

Para cada palavra o modelo constrói três representações matemáticas diferentes:

Query (Q)

Key (K)

Value (V)

Pense em uma biblioteca. A Query representa aquilo que você está procurando. A Key representa a etiqueta de cada livro. E a Value representa a informação armazenada naquele livro.

Quando uma nova palavra chega, o modelo compara sua Query com todas as Keys disponíveis. O objetivo é descobrir quais informações parecem mais relevantes naquele contexto. Mas como essa comparação acontece?

Através de uma operação chamada produto escalar, quem já trabalhou com álgebra linear provavelmente encontrou essa operação. Se dois vetores apontam para direções parecidas, o produto escalar cresce. Se apontam para direções diferentes, o resultado diminui. Estamos medindo a compatibilidade.

Matemática da Attention

Quanto maior o valor obtido, mais relevante aquela informação parece para a tarefa atual. Matematicamente essa etapa pode ser representada por:

© Artemisia Weyl 2024–2026 — Todos os direitos reservados.

QK'

Os resultados podem assumir praticamente qualquer valor, alguns podem ser enormes e outros muito pequenos. O modelo precisa transformar essas pontuações em algo interpretável.

É aqui que entra o **Softmax**

O Softmax é uma função matemática, que converte essas pontuações em pesos. Todos os pesos ficam entre 0 e 1, e a soma deles é sempre igual a 1. Agora podemos interpretar esses valores como uma distribuição de atenção.

Se uma palavra receber peso 0,70 e outra receber 0,05, o modelo está sinalizando que a primeira parece muito mais relevante naquele momento. Depois disso os pesos são utilizados para combinar os Values.

É dessa combinação ponderada que surge a informação que será utilizada pela rede. A fórmula completa ficou famosa:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

A ideia por trás dela é simples, encontrar quais partes do texto são mais relevantes e dar mais importância a elas.

Outro detalhe importante: Essas Queries, Keys e Values não são definidas manualmente por humanos, Ninguém senta e escreve regras dizendo ao modelo o que procurar. Durante o treinamento, a própria rede aprende quais transformações matemáticas tornam essas representações úteis para resolver a tarefa. Em outras palavras o odelo aprende sozinho quais relações merecem atenção e isso muda completamente a filosofia das arquiteturas anteriores.

Uma Mudança de Filosofia

Uma RNN tentava carregar toda a informação através de uma memória que era atualizada palavra após palavra. A Attention segue outra lógica, ela mantém acesso direto à informação e ao invez de perguntar: "O que eu ainda lembro?"** ela pergunta: **"Onde está a informação que eu preciso?".**

Essa mudança parece pequena, mas resolveu um dos maiores gargalos do NLP daquela época. Pela primeira vez os modelos começaram a lidar muito melhor com dependências longas.

Se percebeu também que os pesos de atenção permitiam visualizar quais partes do texto estavam influenciando determinadas decisões, os chamados mapas de atenção. Era quase possível enxergar para onde o modelo estava olhando.

A Attention rapidamente se tornou uma das ideias mais importantes da área, mas naquele momento ela ainda era apenas um componente auxiliar, uma peça acoplada às arquiteturas recorrentes.

Poucos anos depois alguém faria uma pergunta muito mais ousada. Se a Attention funciona tão bem por que continuar usando RNNs e LSTMs? Por que continuar processando palavras uma após a outra?

A resposta viria em 2017 com o artigo *Attention Is All you need* mudaria a história da Inteligência Artificial.

Postagem original no LinkedIn: NLP 101 (4): A invenção da Attention

© Artemisia Weyl 2024–2026 — Todos os direitos reservados.