(Grandes) Modelos de Linguagem

SARAJANE MARQUES PERES

Dados – textos

Para realização da análise automática de textos, seja com fins de resolução de tarefas de **mineração sobre dados textuais**, **processamento de língua natural** ou **recuperação de informação**, é necessário preparar a coleção de documentos textuais (o corpus) a fim de adequá-la ao processamento automático.

Historicamente, tem-se usado uma definição para a organização de um corpus. Porém, com a evolução da IA generativa, a maneira de olhar para documentos mudou. Vamos olhar para alguns conceitos do processamento de dados textuais ...

Dados - texto

Um conjunto de dados organizado a partir de um *corpus*

Um conjunto de n documentos $\mathcal{X} = \{doc_1, doc_2, \ldots, doc_n\}$. Cada um dos documentos, por sua vez, é definido como um conjunto de m termos (radicais, palavras ou conjunto de palavras), na forma $doc_i = \{wt_1, wt_2, \ldots, wt_m\}$, sendo que wt_j pode assumir valores booleanos ou reais.

Dados – textos

Preparação dos documentos

Há uma série de procedimentos que são úteis para preparar uma coleção de documentos antes que ela seja representada como um conjunto de dados $\mathcal{X}$.

- análise léxica (*tokenizing*)
- eliminação de *stopwords*;
- redução dos termos aos seus radicais;

***Vector Space Model* - VSM**

Modelo de representação dos textos em um espaço de vetores. Cada documento é um vetor de termos (index term).

Tokenizing

Um tipo de normalização do texto. O tokenizing transforma o texto em unidades, chamadas tokens.

Top-down tokenization: definimos um padrão e implementamos regras para fazer a separação dos tokens. Por exemplo: separar as palavras usando os espaços entre elas. Decisões precisam ser tomadas em relação a palavras compostas, pontuação, emojis, entiadades nomeadas (nomes, datas etc). Também pode ser feita via caracteres (no Chinês por exemplo).

Botton-up tokenization: usamos estatísticas sobre as sequências de letras para quebrar palavras em tokens do tipo “sub-palavras”. Sub-palavras podem ser strings arbitrárias ou morfemas. Exemplo de algoritmo: byte-pair encoding. O algoritmo inicia por separar caracteres dentro das palavras (já separadas por uma regra – espaço em branco, por exemplo). Então, por contagem de frequência junta caracteres (dentro das palavras) e forma os tokens. O algoritmo executa milhares de junções: palavras frequentes serão tokens, palavras raras serão representadas por suas partes (tokens menores).

Dados – texto

Bag of words: representação para textos baseadas em palavras independentes.

N-gramas:

- Sequência contíguas de **n** termos em texto ou discurso. Tem sido usada para enriquecer a representação para textos do tipo *bag of words* (BOW).
- Sequências em ordem alfabética de **n** termos de palavras consecutivas em uma sentença (após retirada de *stopwords*).
- Sequência de palavras, considerando apenas um subconjunto de classe de palavras que aparecem de forma consecutiva em um texto.

$n = 1 \rightarrow$ unigrama
$n = 2 \rightarrow$ bigrama
$n = 3 \rightarrow$ trigrama
....

Dados – textos

Representação binária

Representação simplificada na qual valores binários (ou pesos binários) indicam a presença ou ausência do termo em um documento. Os termos presentes possuem todos a mesma importância na representação de um documento.

$$doc_1 = \{0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1\}$$

Representação por frequência - *term frequency* – *tf*

Representação na qual a frequência de um termo no documento é associada ao peso do termo na representação daquele documento.

$$doc_1 = \{3, 6, 1, 0, 0, 4, 9, 1, 1, 0, 0, 0, 0, 6, 0, 0, 3, 0, 0, 12, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 3\}$$

Dados – textos

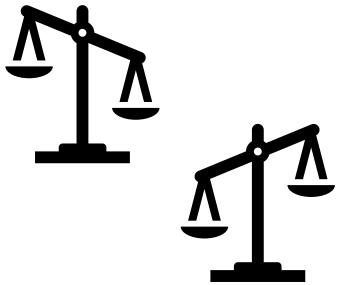
Dois fatores importantes podem direcionar a efetividade de uma representação: a **exaustividade (completude) da indexação** e a **especificidade da representação**.

Exaustividade (completude) da indexação - *exhaustivity of indexing*

- Número de termos associados a um dado documento.
- Número de tópicos diferentes que foram indexados.

Especificidade da representação - *specificity of the index language*

- Número de documentos ao qual um dado termo está associado dentro de uma coleção de documentos.
- Habilidade da representação em descrever um tópico precisamente.



Dados – textos

Frequência inversa nos documentos – *inverse document frequency idf*

$$idf(t_j) = log \frac{n}{nt_j} \quad (1)$$

em que n é o número de documentos no *corpus* e nt_j é o número de documentos nos quais o termo t aparece.

Representação por td-idf

Representação na qual o peso associado a uma palavra é calculando considerando tanto a frequência com a qual ele aparece no texto, quanto ao número de documentos no qual ele aparece.

$$tf\_idf(t_j, doc_i) = tf(t_j, doc_i) * idf(t_j) \quad (2)$$

Nessa representação

- quanto maior a frequência do termo no documento, maior é a representatividade do termo para aquele documento;
- quanto maior o número de documentos no qual um termo aparece, menos discriminante o termo é;

Dados - palavras

Representação para palavras: one-hot-encoding

Representação binária para dados categóricos ou, no caso dessa discussão, para palavras em um vocabulário.

A dimensionalidade do espaço é igual ao número de palavras no vocabulário.

Vocabulário: [casa, prédio, iglu, oca]

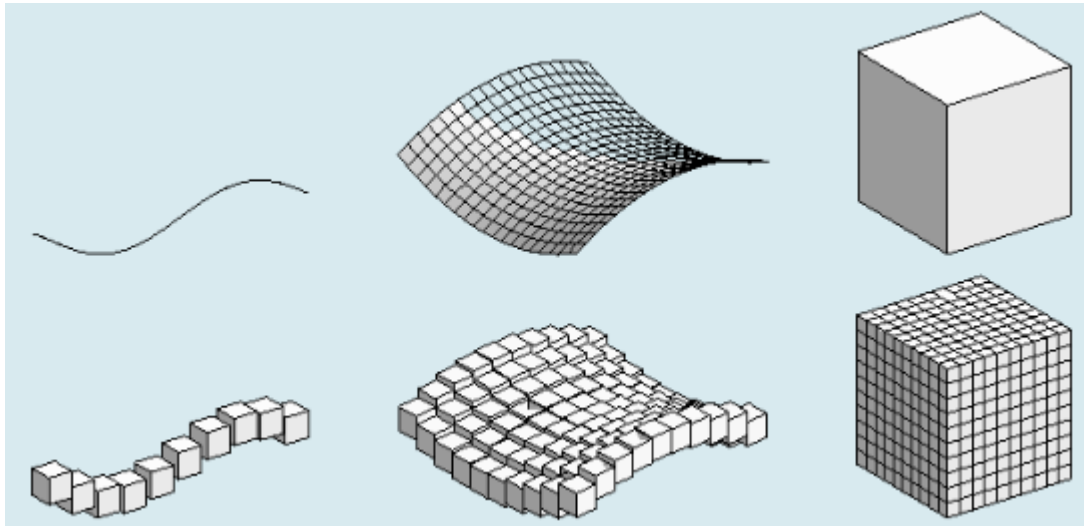
casa = [1 0 0 0] **prédio = [0 1 0 0]** **iglu = [0 0 1 0]** **oca = [0 0 0 1]**

Mal da dimensionalidade

Alta dimensionalidade e alto grau de esparsidade: os conjuntos de dados gerados a partir de um corpus possuem alta dimensionalidade e alto grau de esparsidade (a proporção de zeros na matriz é muito alta).

Maldição da dimensionalidade: diz respeito ao aumento exponencial do volume associado quando se adiciona dimensões extras a um espaço matemático.

Assim, quanto mais características descritivas for necessário processar, maior a quantidade de exemplares necessários para obter um modelo que explique os dados (em algum sentido).



Dados – palavras

Embeddings: quando um objeto é representado em um espaço abstrato topológico, eles usualmente passam a ser chamados de *embeddings*. Atualmente, nós temos associado os *embeddings* a uma codificação numérica para objetos não numéricos, ou às representações densas para textos – usando modelos de linguagem.

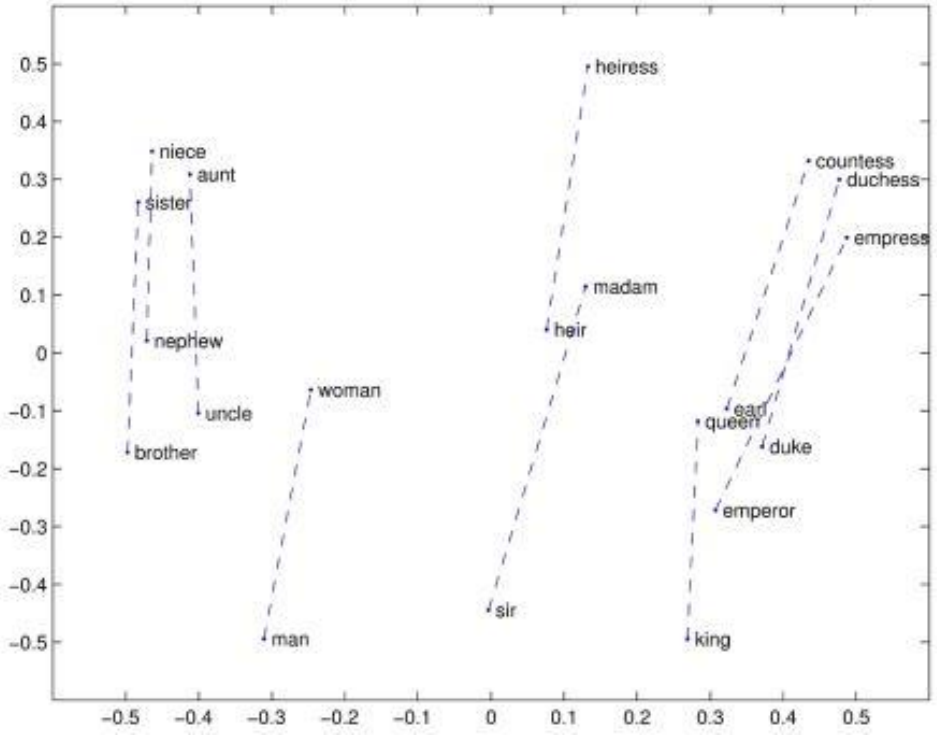
Representações aprendidas:

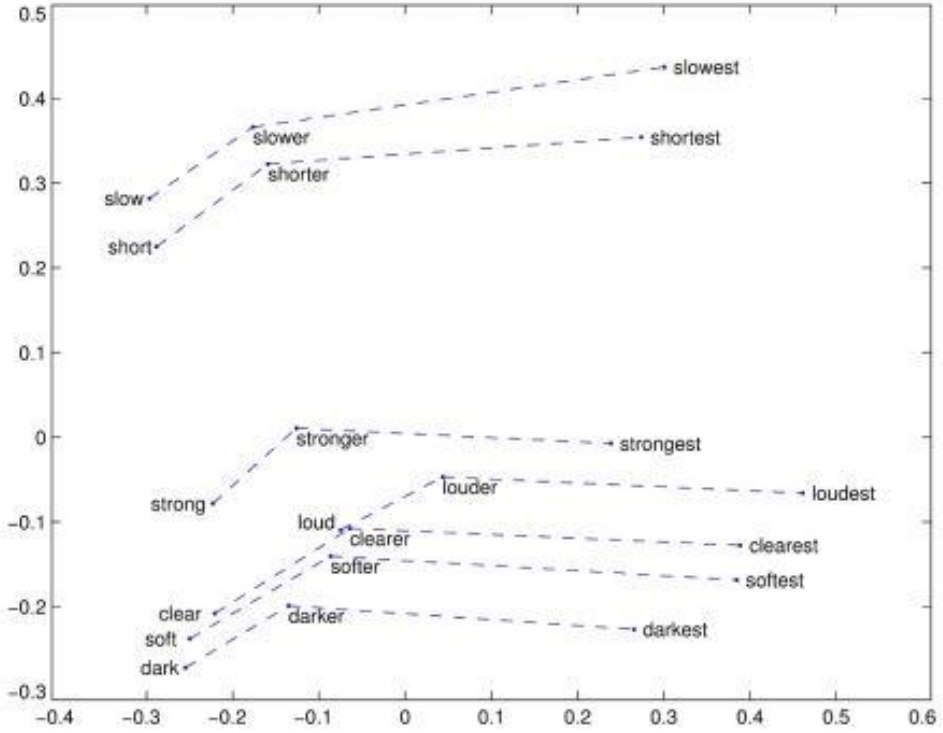
- *Embeddings* estáticos: o método aprende um *embedding* fixo para cada palavra no vocabulário (*word2vec*)
- *Embeddings* dinâmicos: o vetor para cada palavra é diferente em contextos diferentes (*BERT – Bidirectional Encoder Representation from Transformers*)

gato = {0,5871 0,9871 0,5695 0,0147 ... 0,6999}

Dados – palavras

Exemplos de palavras representadas por *embeddings* – com dimensionalidade reduzida





Aprendizado por rede neural - dados

Janela de tamanho = 3

Eu comprei um martelo de cabo vermelho muito caro	(eu, comprei)	(eu, um)	(eu, martelo)			
Eu comprei um martelo de cabo vermelho muito caro	...	...	...	...		
Eu comprei um martelo de cabo vermelho muito caro	...	...	...	....	...	
Eu comprei um martelo de cabo vermelho muito caro	**(martelo, eu)**	**(martelo, comprei)**	**(martelo, um)**	**(martelo, de)**	**(martelo, cabo)**	**(martelo, vermelho)**
Eu comprei um martelo de cabo vermelho muito caro	...	...	...	...	...	...
Eu comprei um martelo de cabo vermelho muito caro	...	...	...	...	...	...
...	...	...	...	...	...	...

Aprendizado por rede neural – skip-gram

Interesse = **martelo** Vizinhança = **(eu, comprei, um, de, cabo, vermelho)** Pares = como destacado no slide anterior

Arquitetura:

- Dimensão do vetor de entrada = V: número de palavras no vocabulário
- Representação da entrada = one-hot-encoding
- Camada escondida = número de neurônios é a dimensão da representação *embeddings* (os pesos -- **palavra-neurônios** -- são os **embeddings** de cada palavra do vocabulário)
- Camada de saída = camada softmax de tamanho V (probabilidade de cada palavra do vocabulário)

Tarefa: dada a palavra de interesse, o modelo prediz as palavras da vizinhança.

Treinamento:

- Para a palavra de interesse como entrada (martelo) execute o feedforward considerando cada palavra da vizinhança como rótulo (eu, comprei ....)
- O esperado é ativar o rótulo com maior probabilidade em cada passada.
- Calcule os “seis” vetores de erro, some-os ponto a ponto e corrija o pesos da camada escondida.

Aprendizado por rede neural – cbow

Interesse = **martelo** Vizinhança = **(eu, comprei, um, de, cabo, vermelho)** Pares = como destacado no slide anterior

Arquitetura:

- Dimensão do vetor de entrada = V: número de palavras no vocabulário
- Representação da entrada = one-hot-encoding
- Camada escondida = número de neurônio é a dimensão da representação *embeddings* (os pesos -- **palavra-neurônios** -- são os **embeddings** de cada palavra do vocabulário)
- Camada de saída = camada softmax de tamanho V (probabilidade de cada palavra do vocabulário)

Tarefa: dada as palavras da vizinhança, o modelo prediz a palavra de interesse.

Treinamento:

- Para a palavra de interesse (martelo) execute o feedforward considerando cada entrada da vizinhança (eu, comprei ....)
- O esperado é ativar “martelo” na saída com maior probabilidade, considerando a média das ativações da camada escondida para toda vizinhança.
- Calcule os “o” vetor de erro e corrija o pesos da camada escondida.

Dados – texto (sentenças – parágrafos)

- Média dos *embeddings* das palavras
- Soma dos *embeddings* das palavras
- Concatenação dos *embeddings*
- *Embedding* da sentença

Modelo codificador-decodificador

Modelo codificador-decodificador é também conhecido como modelo sequência-para-sequência (*sequence-to-sequence*): mapeia uma sequência de entrada para uma sequência de saída.

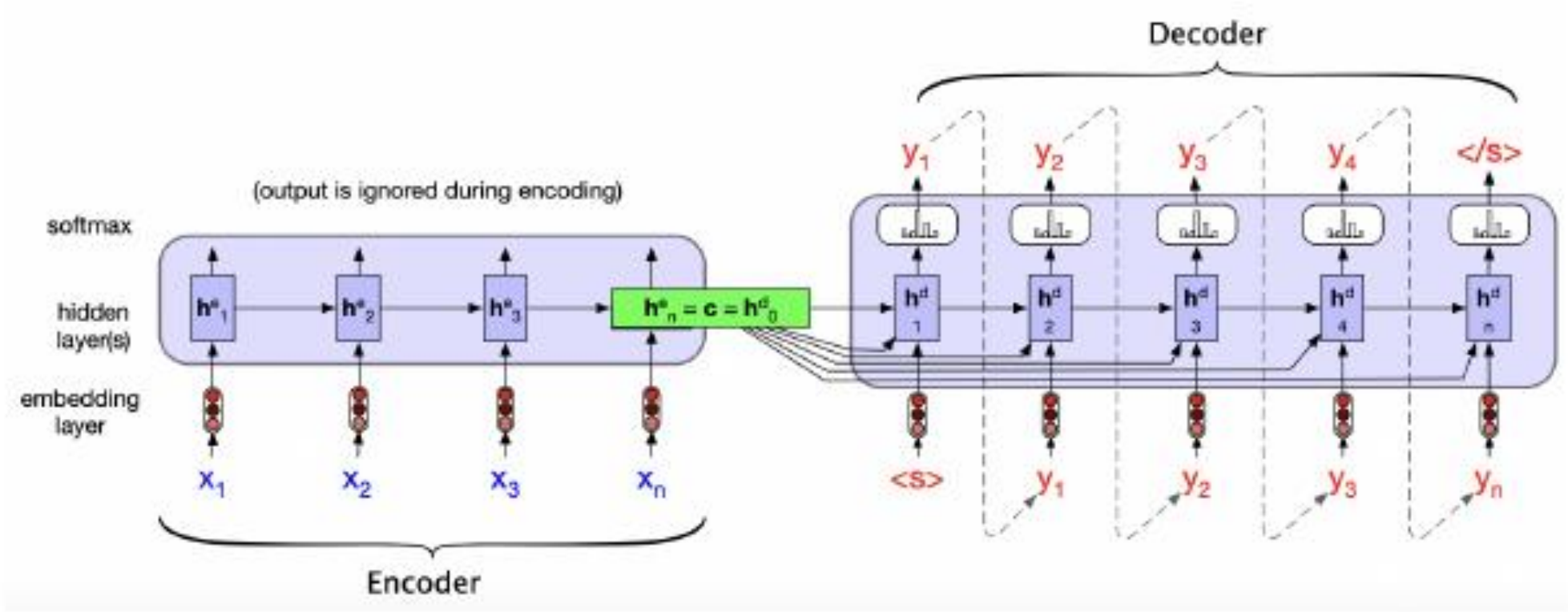
Componentes:

Codificador: processa uma sequência de entrada **(x)** e gera uma representação contextualizada **(h)** desta sequência.

Contexto: o último estado do codificador (**h = c**).

Decodificador: utiliza o contexto **(c)** como estado inicial e gera a sequência de saída, de forma autorregressiva, até a marcação final da sequência.

Modelo codificador-decodificador



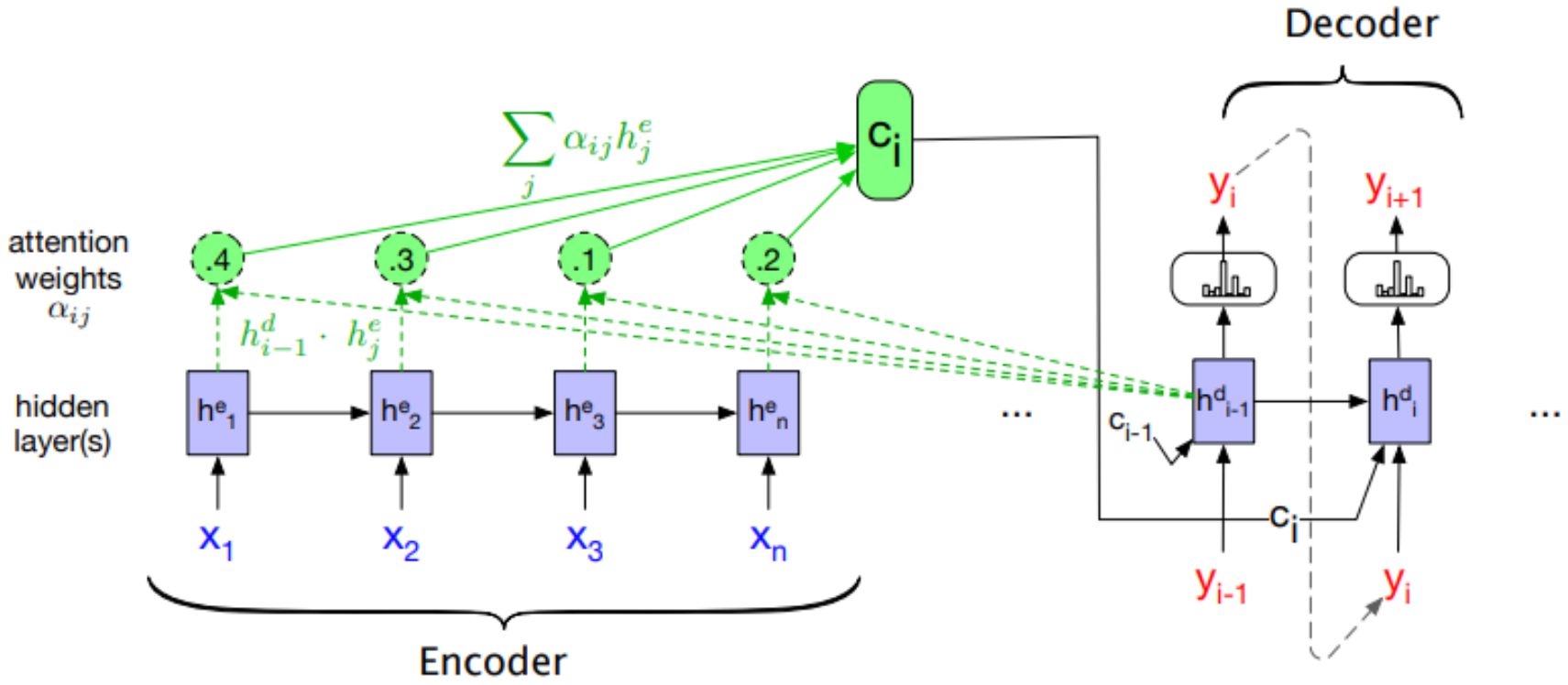
Fonte – (JURAFSKY; MARTIN, 2023)

Mecanismo de atenção

No modelo codificador-decodificador o contexto é o último estado oculto do codificador: ele engloba “toda” a informação sobre a sequência, se tornando um gargalo e perdendo detalhes da informação original.

Solução: mecanismos de atenção - todos os estados ocultos serão utilizados para compor o contexto **c**.

Mecanismo de atenção



Fonte – (JURAFSKY; MARTIN, 2023)

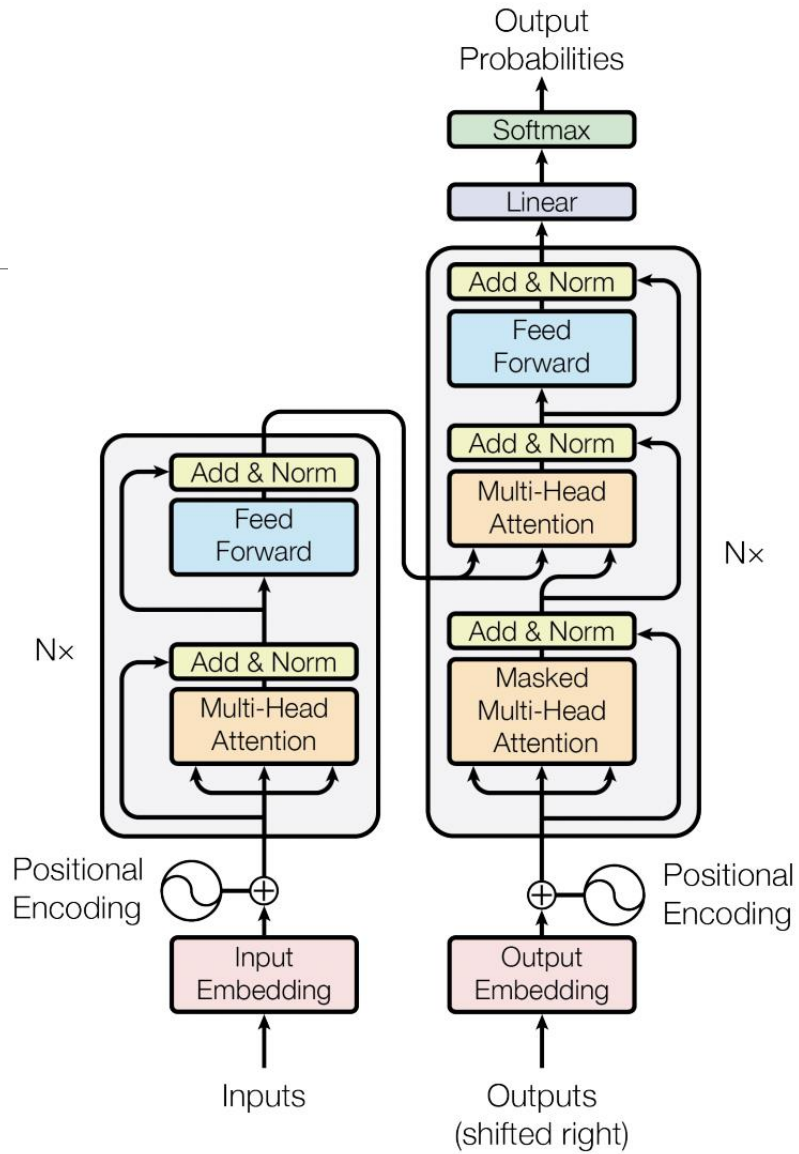
Transformers (original)

Arquitetura que **NÃO utiliza recorrência** na obtenção de representações, portanto permite paralelização do processamento.

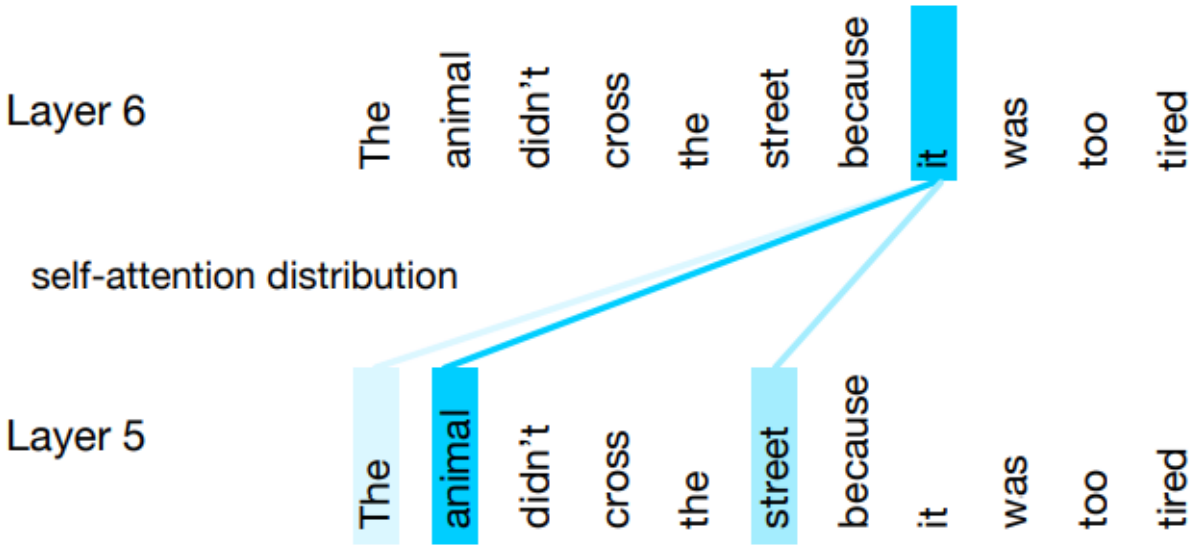
Também utiliza o modelo codificador-decodificador e tem no mecanismo de (auto) atenção o seu principal bloco.

Os blocos ou camadas da arquitetura são chamados de Transformers.

A entrada para um transformer, no processamento da língua natural, é uma sequência de palavras. A saída é um predição sobre qual é a próxima palavra.



Transformers - auto atenção (self attention)



Cálculo de um **score** que mede a similaridade entre os vetores "*embedding* das palavras + *embedding* posicional" "na sequência", até a palavra de interesse.

Sucessivas operações implementam a hierarquia de conceitos (a representação enriquecida do contexto).

Transformers

Codificador: cada bloco é dividido em duas subcamadas – *multihead attention* (scores calculados com pesos diferentes) e rede neural *feedforward*. Entre as camadas existe:

- uma camada residual: que soma a entrada à saída da camada posterior para ajudar a prevenir o problema da dissipação do gradiente.
- uma camada de normalização: para estabilizar a saída da rede após a aplicação do resíduo.

Decodificador: a subcamada *multihead attention* trabalha sobre a saída do codificador. Além disso, há uma camada de *multihead attention* com máscara para não permitir a verificação da posição futura (a que quer ser predita) da sequência em processamento.

Transformers

Multi-head attention: implementa a auto-atenção, inovando no fato que cada posição i de uma sequência de entrada receberá uma “atenção” diferente (matriz de pesos diferentes).

Representação de posição em uma sequência: diferentemente da rede neural recorrente, na qual a representação de ordem para uma sequência está implícita na estrutura do modelo, na arquitetura Transformers a informação não existe. A informação precisa ser fornecida ou aprendida.

Transformes – Rodrigo Nogueira

https://www.youtube.com/watch?v=CP2B-OWvtF8

Grandes modelos de linguagem

Modelos de aprendizado indutivo treinados em grandes quantidades de dados textuais para aprender representações de linguagem. Podem processar e gerar textos de maneira sofisticada e que capture nuances da linguagem (humana, técnica, ...).

Do treino ao uso

Fase 1: Treinamento auto-supervisionado

- O que conhecemos como pré-treinamento
- Constitui o “modelo fundacional”
- Treinamento para prever a próxima palavra
- Congelamento dos pesos para uso na inferência ou no fine-tuning

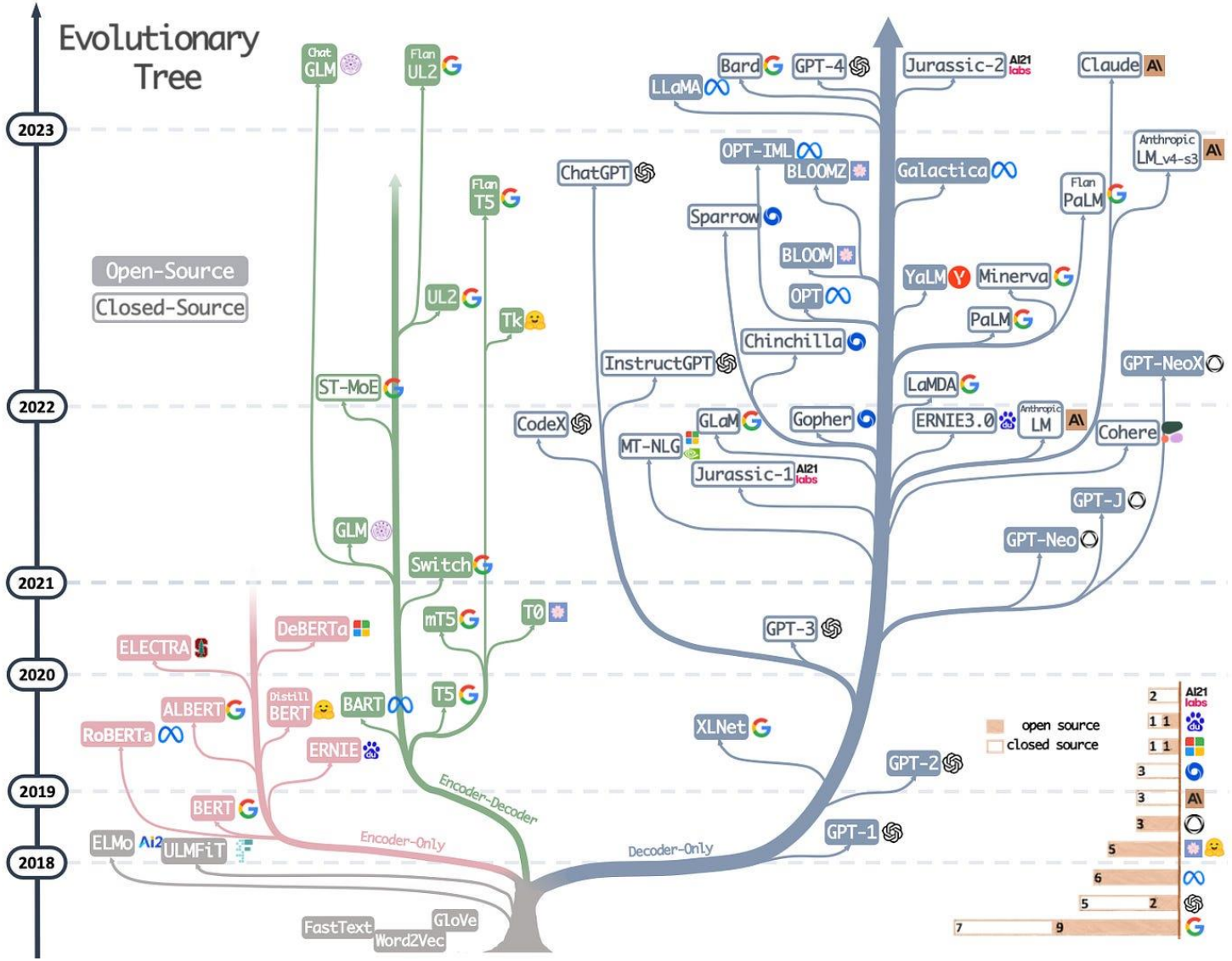
Fase 2: Ajuste fino (fine-tuning) supervisionado

- Continuação do treinamento, porém orientado a tarefas: outras línguas, especialização de domínio, perguntas e respostas, agentes conversacionais, análise de sentimentos …..

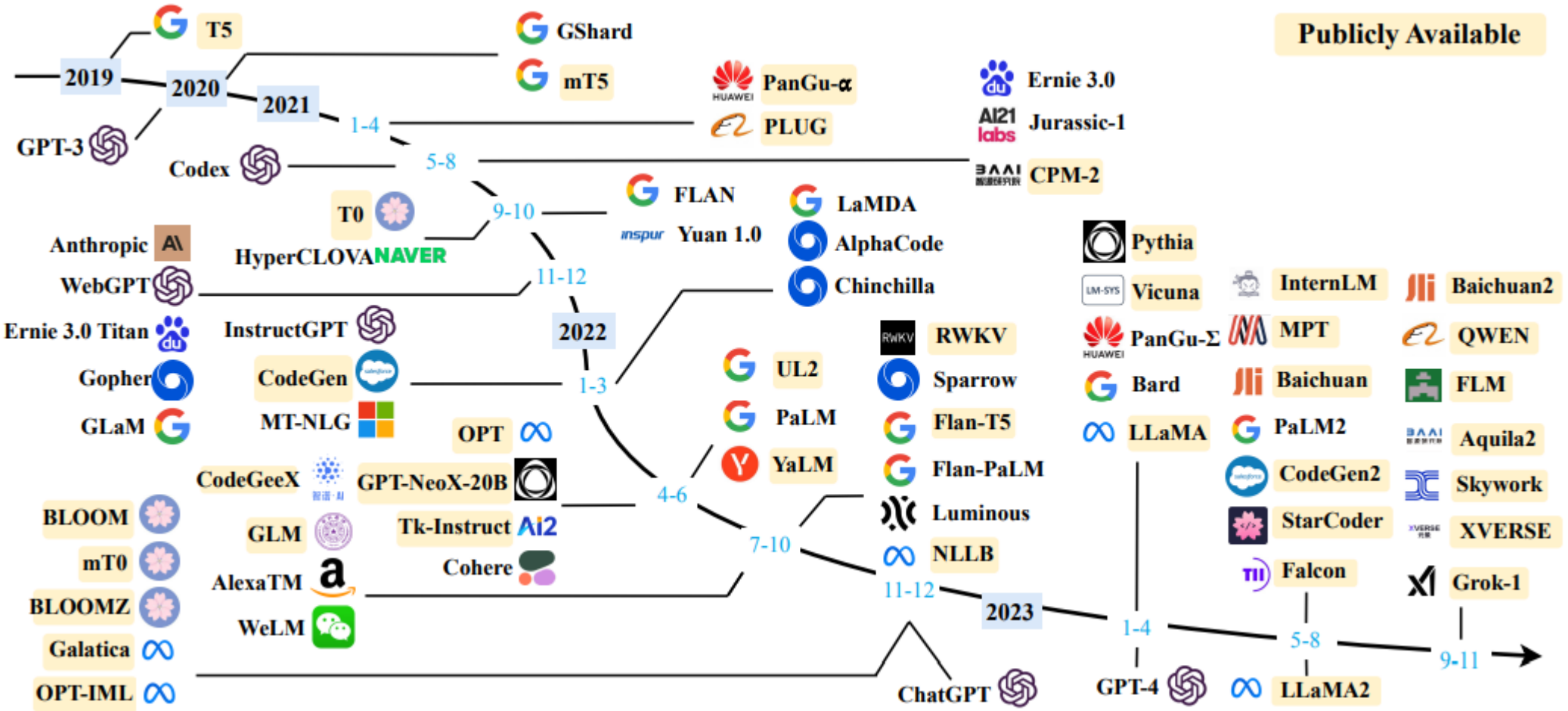
Fase 3: Inferência

- Uso do modelo.
- Admite a engenharia de prompt

Grandes Modelos de Linguagem



https://arxiv.org/pdf/2304.13712



https://arxiv.org/pdf/2303.18223.pdf

https://www.medtextpert.com/prompt-engineering-and-beyond/

Engenharia de prompt

É uma nova disciplina, uma nova forma de usar um sistema (neste caso, um sistema de inteligência artificial generativa).

Em um “prompt” nós escrevemos um texto, em linguagem natural, que descreve uma tarefa a ser realizada pelo modelo de linguagem.

A ideia é “projetar entradas” para o modelo de linguagem (ou outra ferramenta que admita o mesmo tipo de ideia) com o intuito de otimizar a saída recebida (a saída do modelo).

Técnicas mais comuns: zero-shot; one-shot; few-shots; chain-of-thougths; step-by-step ...

- https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results
- https://www.promptingguide.ai/pt/techniques

Exemplo

```
messages=[
        {"role": "assistant", "content": variaveis[2]},
        {"role": "user", "content": variaveis[1]},
        {"role": "system", "content": "Você é um robô chamado Blabinha e está conversando com uma criança." +
                                      "Evite gerar perguntas. Use no máximo 100 palavras."},
        {"role": "system", "content": "Para responder primeiro demonstre contentamento em conhecer a pessoa e por último pergunte se ela já ouviu fala sobre a Amazônia Azul" },
    ]
```

Exemplo

```
messages=[
        {"role": "system",
         "content": "Você é um robô chamado Blabinha e está conversando com uma criança. Use no máximo 150 palavras."},
        {"role": "system", "content": "Siga somente os passos para gerar o texto:" +
                                      "Passo 1- Se a pessoa já souber sobre amazônia azul parabenize ela, se não diga algo reconfortante." +
                                      "Passo 2- Explique brevemente que o desafio consiste em criar um super-herói e para isso vai ser" +
                                      " preciso aprender sobre a amazônia azul" +
                                      "Passo 3 - CONVIDE ela para participar do desafio"},
        {"role": "assistant", "content": variaveis[2]},
        {"role": "user", "content": variaveis[1]},
    ]
```

Exemplo

```
messages=[
        # Responda, Sim, se a pessoa tiver
        {"role": "system", "content": "Responda TRUE se a pessoa pediu dica e FALSE se não pediu"},
        {"role": "user", "content": variaveis[1]},
    ]
.....
messages=[
            {"role": "system",
             "content": "Amazônia Azul é a região que compreende a superfície do mar, águas sobrejacentes ao leito do mar, solo e subsolo marinhos contidos na extensão atlântica que se projeta a partir do litoral até o limite exterior da Plataforma Continental brasileira"},
            {"role": "system", "content": "Primeiro diga que vai dar dica de assuntos sobre a Amazônia Azul e termine enumerando 4 possiveis assuntos que estejam relacionados a Amazônia Azul."},
        ]
```