

Mestrado em Engenharia e Gestão Industrial
Inteligência Artificial aplicada na Indústria

Natural Language Processing (NLP)

Daniel Nogueira



dnogueira@ipca.pt



<https://www.linkedin.com/in/danielfnogueira/>



Natural Language Processing



Linguagem



Curiosidade



Espanto/indignação



Desconforto



Tristeza



Desconfiança



Indicação e aviso



Foco e pesquisa



Alerta silencioso/Receio e prudência



Alerta sonoro/Aviso para perigo



Ameaça e possível ataque



Medo e receio submissivo



Pavor e stress extremos



Descontracção e pedido de mimos



Expectativa



Submissão total



Ansiedade e desconforto



Pedido de atenção e interacção

Natural Language Processing

Linguagem



▷ Linguagem Natural

- ✓ Refere-se ao modo como os seres humanos se comunicam entre si, utilizando um conjunto de regras e códigos predeterminados (sintaxe, semântica, fonética, etc.).
- ✓ Em termos linguísticos, a língua natural é uma expressão que apenas se aplica a uma linguagem que evoluiu naturalmente, como a fala nativa (primeira língua) de um indivíduo.
- ✓ É formada por unidades menores (palavras) que possuem significados, e essas unidades, por sua vez, são formadas por unidades ainda menores (como vogais e consoantes).
- ✓ É caracterizada por sua **complexidade, ambiguidades, variabilidade e nuances contextuais**.
- ✓ O estudo das línguas permite identificar muito sobre seu funcionamento (regras e códigos) e sobre como a mente e o cérebro humanos processam a linguagem.

Natural Language Processing

Linguagem

▷ Linguagem Natural

“Sistema de **símbolos** de um **vocabulário** que, quando colocados em uma determinada **ordem** e expressos em um determinado **contexto**, transmitem um **significado.**”

Exemplos:

- Os idiomas como o português, o inglês, etc.



Natural Language Processing

Linguagem

▷ Linguagem Artificial

- ✓ Refere-se a linguagem criada artificialmente para algum propósito (objetivos específicos).
- ✓ São projetadas para serem precisas e sem ambiguidades

Exemplos:

- Linguagens de Programação: utilizadas para escrever programas de computador (Python, Java, C++, etc).
- Linguagens Formais: utilizadas em matemática e lógica (notação matemática, lógica proposicional, etc).
- Linguagens Construídas: criadas para facilitar a comunicação humana, como línguas planejadas (Lingua de sinais, Klingon - da série "Star Trek", etc).



Natural Language Processing

Definição

- ▷ Conjunto de métodos para tornar a linguagem humana acessível a máquinas
- ▷ Subcampo da linguística e ciência de dados
- ▷ Inclui técnicas eficientes para representação de dados textuais
- ▷ Analisa e produz insights de dados de áudio e texto



Natural Language Processing



Definição

Linguística

Modelar a linguagem



Computação

Implementar os modelos



Tornar as máquinas aptas a **processarem** a linguagem natural.

- Entender, gerar e extrair informações úteis;
- Comunicar



Natural Language Processing

Aplicações

▷ Sistemas de Diálogo

- Assistentes virtuais: Alexa (Amazon), Siri (Apple), Google Assistant (Google) e Cortana (Microsoft);
- Chat-bots existentes em portais de atendimento, de lojas virtuais, bancos, órgãos públicos, etc.;
- Geração e análise de diálogos (sistemas de segurança preventivos – prevenção de crimes, por exemplo)

▷ Extração e Recuperação de Informações

- Construção automática (ou “preenchimento”) de uma infobox a partir de texto;
- Localização, em grandes coleções, de material (geralmente documentos) de uma natureza não estruturada (geralmente texto) que satisfaz uma necessidade de informação.



Natural Language Processing

Aplicações



▷ Perguntas e resposta (geração e solução)

- Sobre assuntos de domínio genérico ou restrito;
- Relativas a comunidades, como StackOverflow, redes sociais etc;
- Na resolução de questões de múltipla escolha (a partir da "leitura" de textos sobre o assunto) ou raciocínio lógico ou relativas a imagens;
- Identificação de similaridade e/ou reformulação de questões;
- Classificação e categorização de questões.

▷ Classificação de texto, discurso e imagem

- Classificação de documentos, sentenças, etc.;
- Classificação de sentimentos, emoções, intenções ou predileções (em serviços como Youtube e Netflix);
- Modelagem abstrata de tópicos, para a descoberta de estruturas semânticas ocultas em textos;
- Classificação de reclamações;
- Classificação de imagens.



Natural Language Processing

Aplicações

- ▷ Reconhecimento de padrões
- ▷ Resumos automatizados de texto e de discursos
- ▷ Máquina de tradução
- ▷ Processamento de imagens



Generative AI

IA Generativa

- A IA generativa é um tipo de inteligência artificial capaz de criar conteúdo original, como linguagem natural, imagens, áudio e código.
- A saída de uma IA generativa é baseada nas entradas fornecidas pelo usuário.
- As aplicações de IA generativa que usam linguagem natural como entrada são alimentadas por grandes modelos de linguagem (LLM) para executar processamento de linguagem natural (NLP).

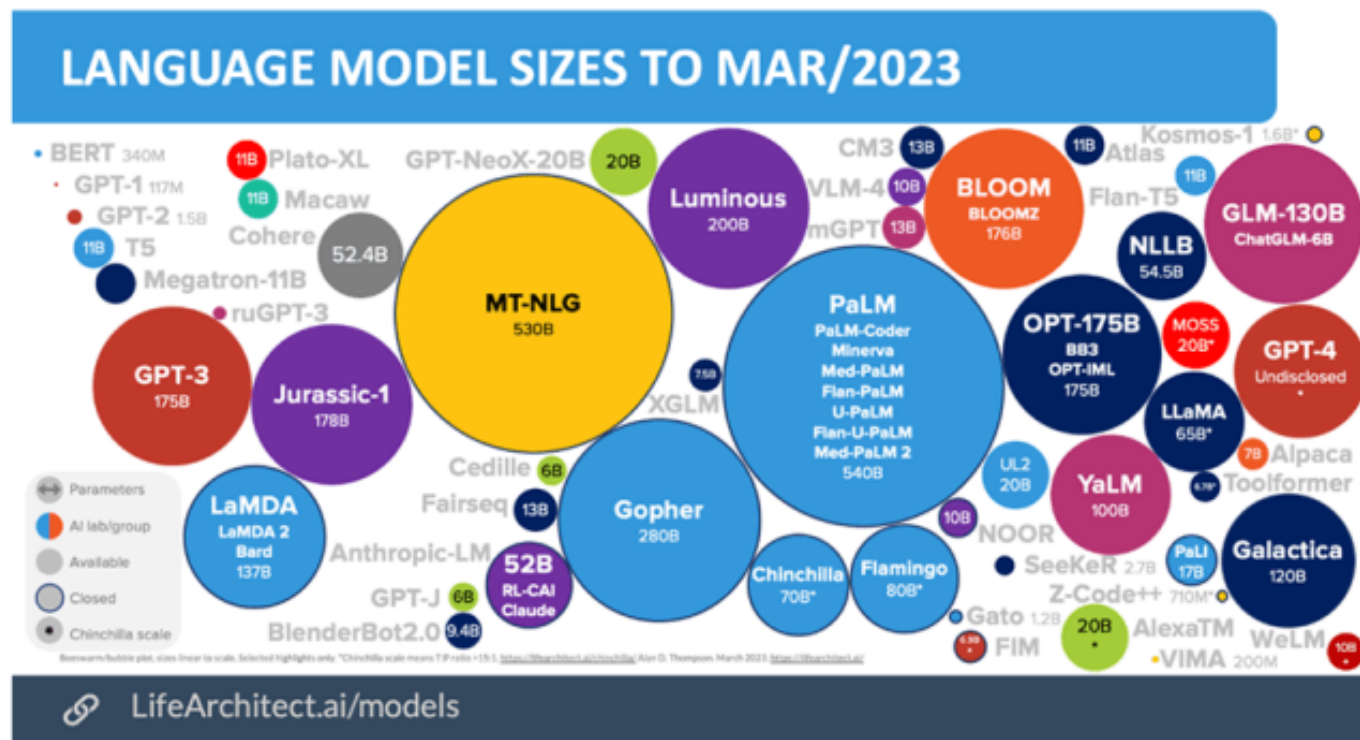


Generative AI

IA Generativa

LLM

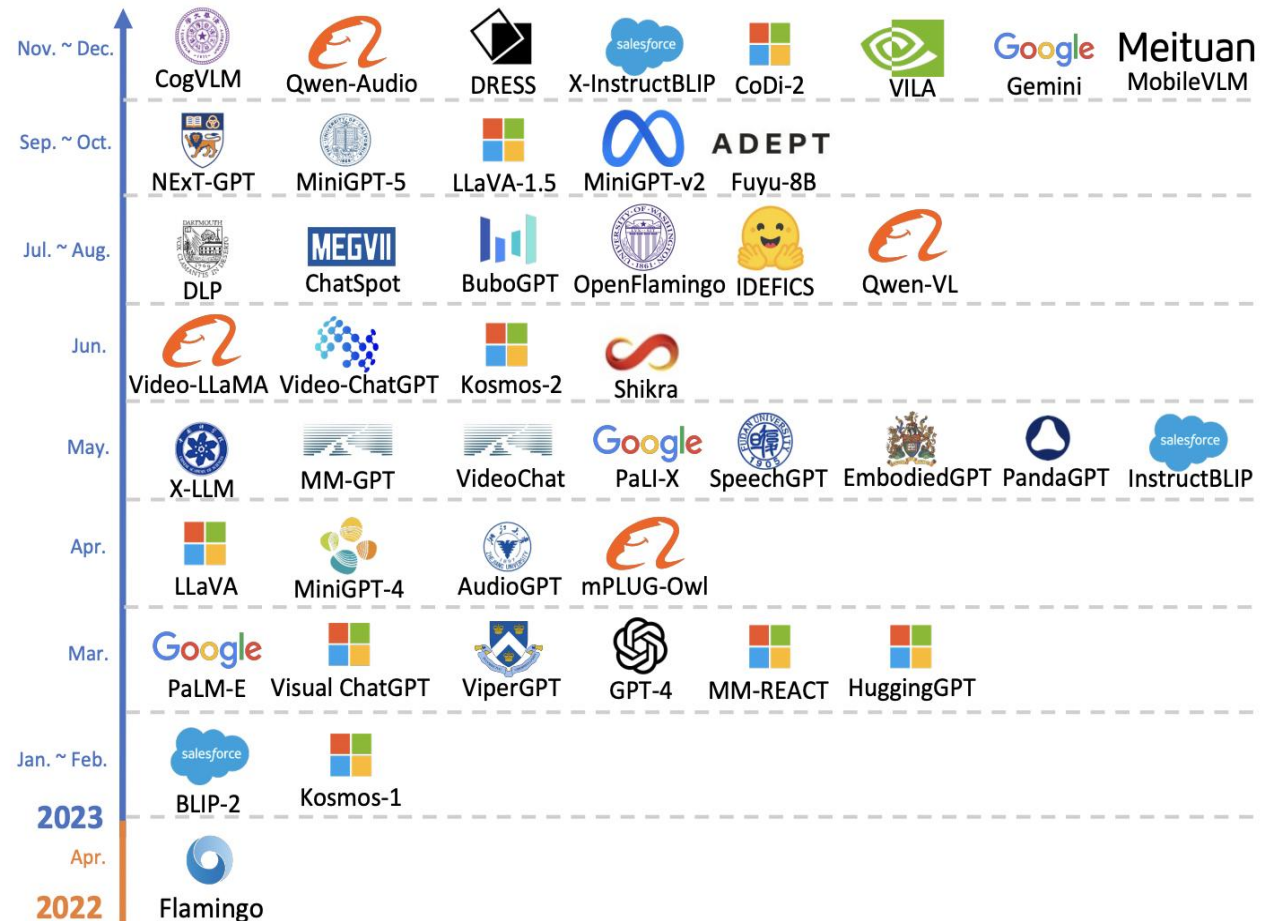
Large Language Models (LLMs) são **modelos** treinados em **grandes volumes de dados** e usam muitos parâmetros para gerar resultados para, por exemplo, responder a perguntas, traduzir idiomas e gerar texto.



Generative AI

IA Generativa

LLM



SOURCE:

Zhang, D., Yu, Y., Li, C., Dong, J., Su, D., Chu, C., & Yu, D. (2024). Mm-lms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Generative AI

Processos

- ▷ Separar o texto em unidades menores (*“tokens”*)
- ▷ Remover elementos (palavras, por exemplo) sem “significado” semântico (stopwords)
- ▷ Etiquetagem morfossintática (*“part-of-speech tagging”*)
- ▷ Remover acentos e pontuações
- ▷ Padronizar a forma de escrita (colocar verbos no infinitivo, por exemplo)

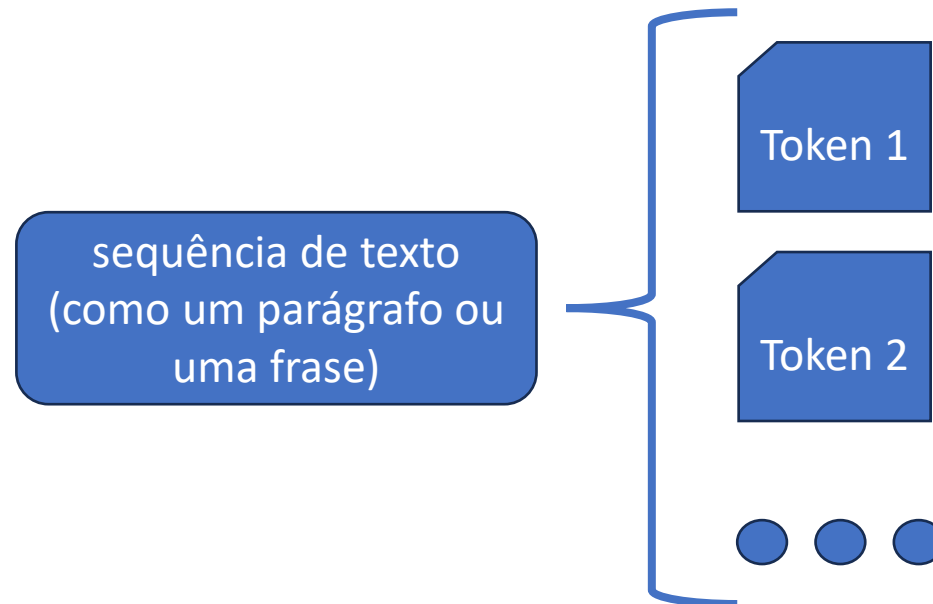
Stemming e Lemmatization



Generative AI

Processos

- ▷ Separar o texto em unidades menores (“tokens”)



Generative AI

Processos

- ▷ Separar o texto em unidades menores (*“tokens”*)

Tokenização de Palavras: Divide o texto em palavras individuais.

“O estudante foi para a escola de carro.”

O	estudante	foi	para	a	escola	de	carro	.
---	-----------	-----	------	---	--------	----	-------	---



Generative AI

Processos

- ▷ Separar o texto em unidades menores (“tokens”)

Tokenização de Caracteres: Divide o texto em caracteres individuais.

“O estudante foi para a escola de carro.”

O	e	s	t	u	d	a	n	t	e	...
---	---	---	---	---	---	---	---	---	---	-----



Generative AI

Processos

- ▷ Separar o texto em unidades menores (“tokens”)

Tokenização de Sentenças: Divide o texto em sentenças.

“Tokenização é importante. Ela facilita o processamento.”

Tokenização é importante.

Ela facilita o processamento.



Generative AI

Processos

▷ Separar o texto em unidades menores (“tokens”)

Desafios:

- **Pontuação:** Decidir se a pontuação deve ser separada das palavras ou tratada como parte das palavras.

"Olá, mundo!"

Separar pontuação das palavras:

✓ Tokens: "Olá" ", " "mundo" "!"

Manter a pontuação junto às palavras:

✓ Tokens: "Olá," "mundo!"



Generative AI

Processos



▷ Separar o texto em unidades menores (“tokens”)

Desafios:

- **Pontuação:** Decidir se a pontuação deve ser separada das palavras ou tratada como parte das palavras.
- **Aglutinação e Separação de Palavras:** Em algumas línguas, palavras compostas podem ser aglutinadas ou separadas de maneiras que podem variar contextualmente.

“O secretário-geral falou aos membros do partido.”

Tokens: "O" "secretário" "-" "geral" "falou" "aos" "membros" "do" "partido" "."

"secretário-geral" é uma palavra composta que se refere a uma posição de liderança em uma organização

Generative AI

Processos

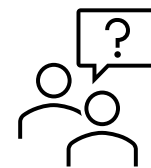


▷ Separar o texto em unidades menores (“tokens”)

Desafios:

- **Pontuação:** Decidir se a pontuação deve ser separada das palavras ou tratada como parte das palavras.
- **Aglutinação e Separação de Palavras:** Em algumas línguas, palavras compostas podem ser aglutinadas ou separadas de maneiras que podem variar contextualmente.
- **Ambiguidade Linguística:** Palavras que têm múltiplos significados podem ser difíceis de tokenizar corretamente sem um contexto adequado.

“Ele foi ao banco.”



Generative AI

Processos

▷ Separar o texto em unidades menores (“tokens”)

Desafios:

- **Pontuação:** Decidir se a pontuação deve ser separada das palavras ou tratada como parte das palavras.
- **Aglutinação e Separação de Palavras:** Em algumas línguas, palavras compostas podem ser aglutinadas ou separadas de maneiras que podem variar contextualmente.
- **Ambiguidade Linguística:** Palavras que têm múltiplos significados podem ser difíceis de tokenizar corretamente sem um contexto adequado.

“Ele foi ao banco para pedir um empréstimo.”



Generative AI

Processos



▷ Separar o texto em unidades menores (“tokens”)

Desafios:

- **Pontuação:** Decidir se a pontuação deve ser separada das palavras ou tratada como parte das palavras.
- **Aglutinação e Separação de Palavras:** Em algumas línguas, palavras compostas podem ser aglutinadas ou separadas de maneiras que podem variar contextualmente.
- **Ambiguidade Linguística:** Palavras que têm múltiplos significados podem ser difíceis de tokenizar corretamente sem um contexto adequado.
- **Contratos e Abreviações:** Lidar com contrações, abreviações e diferentes formas de escrita.

“Vou à casa da Maria.”

Tokens: “Vou” “à” “casa” “da” “Maria”

“à” é uma contração da preposição "a" com o artigo "a".

Generative AI

Processos



▷ Separar o texto em unidades menores (“tokens”)

Desafios:

- **Pontuação:** Decidir se a pontuação deve ser separada das palavras ou tratada como parte das palavras.
- **Aglutinação e Separação de Palavras:** Em algumas línguas, palavras compostas podem ser aglutinadas ou separadas de maneiras que podem variar contextualmente.
- **Ambiguidade Linguística:** Palavras que têm múltiplos significados podem ser difíceis de tokenizar corretamente sem um contexto adequado.
- **Contratos e Abreviações:** Lidar com contrações, abreviações e diferentes formas de escrita.

“Dr. Silva é um bom médico”

Tokens: “Dr.” “Silva” “é” “um” “bom” “médico”

“Dr.” é uma abreviação de “Doutor”.

Generative AI

Processos



- ▷ Remover elementos (palavras, por exemplo) sem “significado” semântico (**stopwords**)

Stopwords são palavras que ocorrem com alta frequência em um idioma, mas que carregam pouco ou nenhum significado semântico próprio.

Exemplos:

Em português: "de", "a", "o", "é", "em", "um", "e", etc.

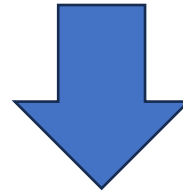
Na análise de texto, estas palavras podem ser removidas para focar nos termos que carregam mais informação semântica.

Generative AI

Processos

- ▷ Remover elementos (palavras, por exemplo) sem “significado” semântico (**stopwords**)

“gosto conversar processamento linguagem natural amigos!”



“Eu gosto de conversar sobre processamento de linguagem natural com meus amigos!”



Generative AI

Processos



- ▷ Remover elementos (palavras, por exemplo) sem “significado” semântico (**stopwords**)

Processo de Remoção de Stopwords

- 1. Identificação de Stopwords:** Primeiro, é necessário ter uma lista de stopwords. Esta lista pode ser específica para cada idioma e pode variar dependendo da aplicação.
- 2. Tokenização:** A frase é dividida em tokens.
- 3. Remoção de Stopwords:** Cada token é comparado com a lista de stopwords e, se um token estiver na lista, ele é removido.
- 4. Recomposição da Frase (opcional):** Os tokens restantes podem ser recombinaados para formar a frase filtrada.

Generative AI

Processos



- ▷ Remover elementos (palavras, por exemplo) sem “significado” semântico (**stopwords**)

“Eu gosto de conversar sobre processamento de linguagem natural com meus amigos!”

Lista de Stopwords (exemplo): "eu", "de", "sobre", "com", "meus"

- ✓ Tokenização:

"Eu" "gosto" "de" "conversar" "sobre" "processamento" "de" "linguagem" "natural" "com" "meus" "amigos!"

- ✓ Remoção de Stopwords:

"gosto" "conversar" "processamento" "linguagem" "natural" "amigos!"

- ✓ Recomposição da Frase (opcional):

“gosto conversar processamento linguagem natural amigos!”

Generative AI

Processos



- ▷ Remover elementos (palavras, por exemplo) sem “significado” semântico (***stopwords***)

“Eu gosto de conversar sobre processamento de linguagem natural com meus amigos!”



“gosto conversar processamento linguagem natural amigos!”

Generative AI

Processos



▷ Etiquetação morfossintática (“*part-of-speech tagging*”)

POS tagging: refere-se à atribuição de **rótulos gramaticais** a cada palavra de um texto, indicando sua **categoria sintática** (substantivo, verbo, adjetivo, etc.).

- ✓ Ajuda a entender a função gramatical de cada palavra dentro de uma frase;
- ✓ Fundamental para a análise sintática e semântica do texto.

Generative AI

Processos

▷ Etiquetação morfossintática (*"part-of-speech tagging"*)

"The quick brown fox jumps over the lazy dog"

- **"The"** é marcado como "DT" (Determiner - determinante).
- **"quick"** é marcado como "JJ" (Adjective - adjetivo).
- **"brown"** é marcado como "JJ" (Adjective - adjetivo).
- **"fox"** é marcado como "NN" (Noun - substantivo).
- **"jumps"** é marcado como "VBZ" (verbo na 3ª pessoa do singular no presente).
- **"over"** é marcado como "IN" (Preposition - preposição).
- **"the"** é marcado como "DT" (Determiner - determinante).
- **"lazy"** é marcado como "JJ" (Adjective - adjetivo).
- **"dog"** é marcado como "NN" (Noun - substantivo).



Generative AI

Processos



▷ Etiquetação morfossintática (*“part-of-speech tagging”*)

Importância do POS Tagging

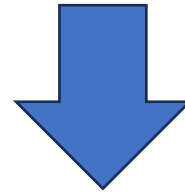
- ✓ **Desambiguação:** Ajuda a desambiguar palavras que podem ter múltiplas funções gramaticais. Por exemplo, "book" pode ser um substantivo ("I read a book") ou um verbo ("I will book a ticket").
- ✓ **Análise Sintática:** Facilita a construção de árvores sintáticas, que são representações hierárquicas da estrutura de uma frase.
- ✓ **Extração de Informação:** Identificação de nomes próprios, datas, locais, etc.
- ✓ **Tradução Automática:** Melhora a precisão de sistemas de tradução automática ao fornecer informações sobre a estrutura gramatical das frases.
- ✓ **Análise de Sentimentos:** Contribui para a análise de sentimentos ao ajudar a identificar adjetivos e outros elementos que carregam carga emocional.

Generative AI

Processos

▷ Remover acentos e pontuações

“Eu gosto de conversar sobre processamento de linguagem natural com meus amigos!”



“gosto conversar processamento linguagem natural amigos”



Generative AI

Processos

▷ Padronizar a forma de escrita (Stemming e Lemmatization)

Stemming:

- ✓ Processo de redução de palavras flexionadas (ou às vezes derivadas) ao seu radical ou raiz.
- ✓ O radical é parte da palavra que contém o significado principal, desconsiderando a flexão gramatical.
- ✓ Opera de maneira heurística, aplicando regras simples como remoção de sufixos comuns (como "s", "es", "ed", "ing" etc.)
- ✓ Embora seja rápido e fácil de implementar, pode resultar em raízes não reconhecíveis ou não válidas em alguns casos.



Generative AI

Processos

▷ Padronizar a forma de escrita (Stemming e Lemmatization)

“gosto conversar processamento linguagem natural amigos”

ORIGINAL	STEM
Gosto	Gost
Conversar	Convers
Processamento	Processament
Linguagem	Linguag
Natural	Natur
Amigos	Amig



Generative AI

Processos

▷ Padronizar a forma de escrita (Stemming e Lemmatization)

Lemmatization:

- ✓ Processo mais refinado que envolve a análise morfológica das palavras para determinar a forma básica, ou lema.
- ✓ O lema é a forma canônica de uma palavra (que você encontra em um dicionário).
- ✓ Leva em consideração o contexto e a classe gramatical da palavra.
- ✓ Isso é alcançado através de regras linguísticas complexas e de um dicionário que mapeia palavras flexionadas para seus lemas.



Generative AI

Processos

▷ Padronizar a forma de escrita (Stemming e Lemmatization)

“gosto conversar processamento linguagem natural amigos”

ORIGINAL	Lemma
Gosto	GostAR
Conversar	ConversAR
Processamento	ProcessamentENTO
Linguagem	LinguagEM
Natural	NaturAL
Amigos	AmigOS



Generative AI

Processos

▷ Padronizar a forma de escrita (Stemming e Lemmatization)

“~~gost~~o conversar processamento linguagem natural amigos”
 “gostar conversar processamento linguagem natural amigos”

ORIGINAL	Lemma
Gosto	GostAR
Conversar	ConversAR
Processamento	ProcessamentENTO
Linguagem	LinguagEM
Natural	NaturAL
Amigos	AmigOS



Generative AI

Processos



▷ Padronizar a forma de escrita (Stemming e Lemmatization)

Diferenças principais:

- ✓ **Precisão:** {
 - Lemmatization é mais precisa porque utiliza conhecimento linguístico detalhado sobre as palavras.
 - Stemming pode gerar resultados não válidos em alguns casos.

- ✓ **Complexidade:** {
 - Lemmatization é mais complexa computacionalmente, pois envolve análise morfológica e uso de dicionários
 - Stemming é mais simples e baseado em regras heurísticas.

- ✓ **Aplicações:** {
 - Lemmatization é preferido em aplicações onde a precisão é crucial, como em sistemas de questionamento ou análise de sentimentos.
 - Stemming é frequentemente usado em casos onde a velocidade é mais importante que a precisão, como em motores de busca.

Generative AI

Embeddings

Definição:

- ✓ São representações vetoriais de palavras ou frases em um espaço de alta dimensionalidade;
- ✓ Esses vetores são gerados de forma que palavras ou frases com significados semelhantes tenham representações próximas umas das outras no espaço vetorial.

“gostar conversar processamento linguagem natural amigos”



[0.21, -1.43, 0.87,]

ou

[(0.21, 0.33, 0.55), (-2.43, 1.0, 0.55),]



Generative AI

Embeddings

Definição:

- ✓ São representações vetoriais de palavras ou frases em um espaço de alta dimensionalidade.
- ✓ Esses vetores são gerados de forma que palavras ou frases com significados semelhantes tenham representações próximas umas das outras no espaço vetorial.

Objetivo:

- ✓ Capturar relações semânticas e contextuais entre palavras.
- ✓ Eles permitem que algoritmos de Machine Learning processem texto de maneira eficiente, transformando dados textuais em uma forma numérica que pode ser usada por modelos.



Generative AI

Embeddings

“rei”

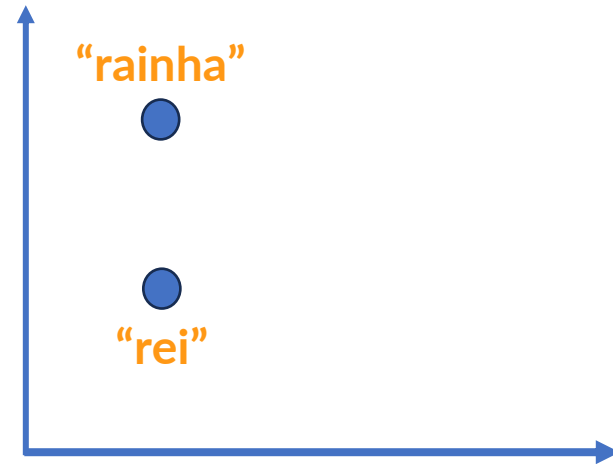


[0.21, 0.87]

“rainha”



[0.21, 1.35]



Generative AI

Tokens

O arquiteto não reiniciou o servidor
porque ele estava com preguiça.

“Tokenização”: processo de quebrar longas quantidades de texto em unidades menores. Unidades estas que podem ser mapeadas para se tornarem números.

Tokens por caracter

LETRA	Indice	LETRA	Indice	LETRA	Indice
A	1	I	9	Q	17
B	2	J	10	R	18
C	3	K	11	S	19
D	4	L	12	T	20
E	5	M	13	U	21
F	6	N	14	V	22
G	7	O	15	X	23
H	8	P	16	Z	24



Generative AI

Tokens

15 1 18 17 21

O arquiteto não reiniciou o servidor
porque ele estava com preguiça.



[15, 0, 1, 18, 17, 21, 9, 20, 5, 20, 15, 0, 14, 1, 15, 0, 18, 5, 9,
14, 9, 3, 9, 15, 21, 0, 15, 0, 19, 5, 18, 22, 9, 4, 15, 18, 0, 16,
15, 18, 17, 21, 5, 0, 5, 12, 5, 0, 5, 19, 20, 1, 22, 1, 0, 3, 15,
13, 0, 16, 18, 5, 7, 21, 9, 3, 1]

TOTAL: 68 Tokens

Tokens por caracter

LETRA	Indice	LETRA	Indice	LETRA	Indice
A	1	I	9	Q	17
B	2	J	10	R	18
C	3	K	11	S	19
D	4	L	12	T	20
E	5	M	13	U	21
F	6	N	14	V	22
G	7	O	15	X	23
H	8	P	16	Z	24



A quantidade de tokens a serem processados pode ser muito grande!!!!

Generative AI

Tokens

<https://github.com/pythonprobr/palavras>



Tokens por palavra

O arquiteto não reiniciou o servidor
porque ele estava com preguiça.

219195 26887 213002 267358 219195 281596
O arquiteto não reiniciou o servidor
252197 104165 119840 76912 254276
porque ele estava com preguiça.

[219195, 26887, 213002, 267358, 219195, 281596,
252197, 104265, 119840, 76912, 254276]

TOTAL: 11 Tokens

PALAVRA	Indice	PALAVRA	Indice	PALAVRA	Indice
A	1	ELE	104165	PORQUE	252197
ABAIXO	2
ABALADO	3	ESTAVA	119840	PREGUIÇA	254276
...
ARQUITETO	26887	NÃO	213002	REINICIOU	267358
...
COM	76912	O	219195	SERVIDOR	281593
...	ZUMBIR	320094

Aumentamos muito a complexidade do nosso dicionário!!!!

Generative AI

Tokens

<https://github.com/pythonprobr/palavras>

Tokens por prefixo, radical e sufixo

O arquiteto não reiniciou o servidor
porque ele estava com preguiça.

Bebendo



Generative AI

Tokens

<https://github.com/pythonprobr/palavras>

Tokens por prefixo, radical e sufixo

O arquiteto não reiniciou o servidor
porque ele estava com preguiça.

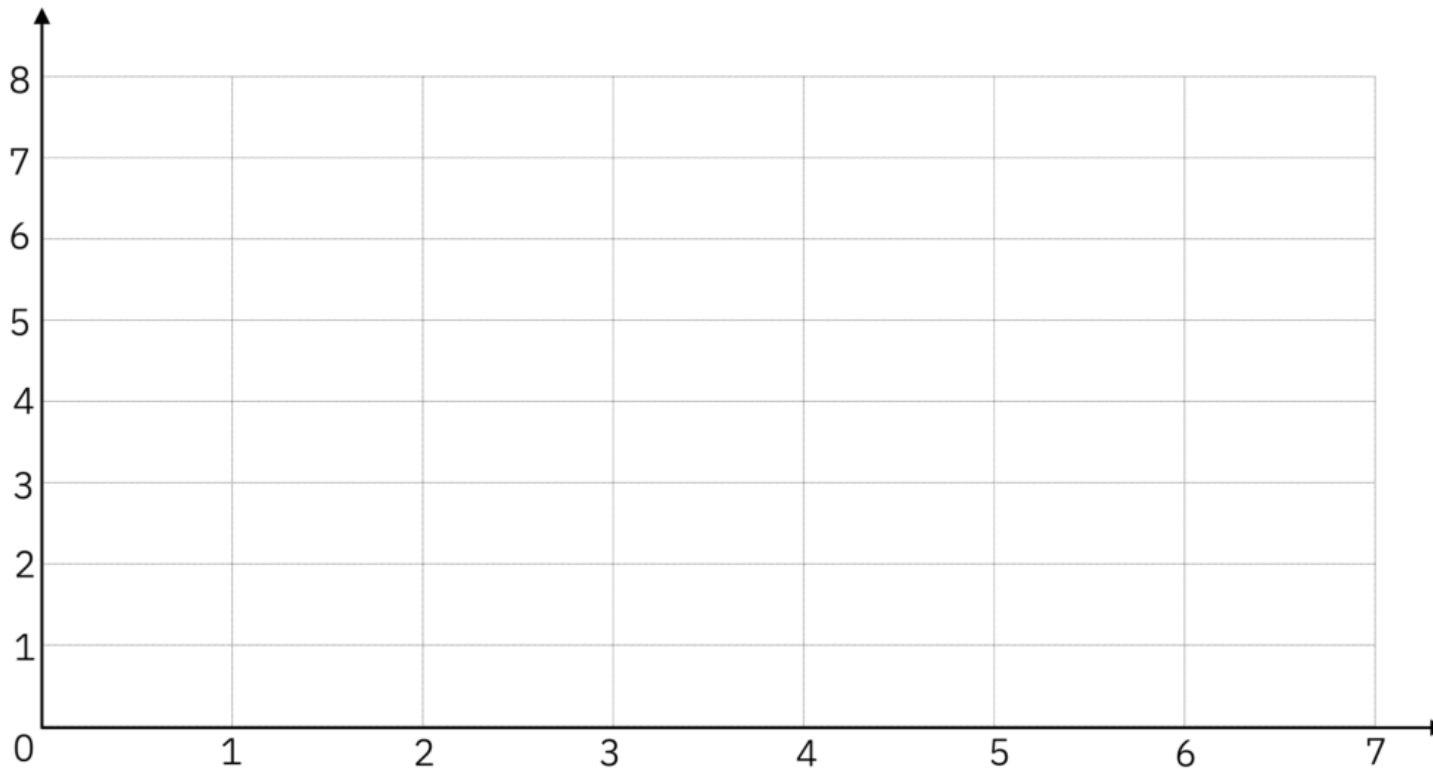
O arquiteto não reiniciou o servidor porque ele estava com preguiça.

TOTAL: 28 Tokens



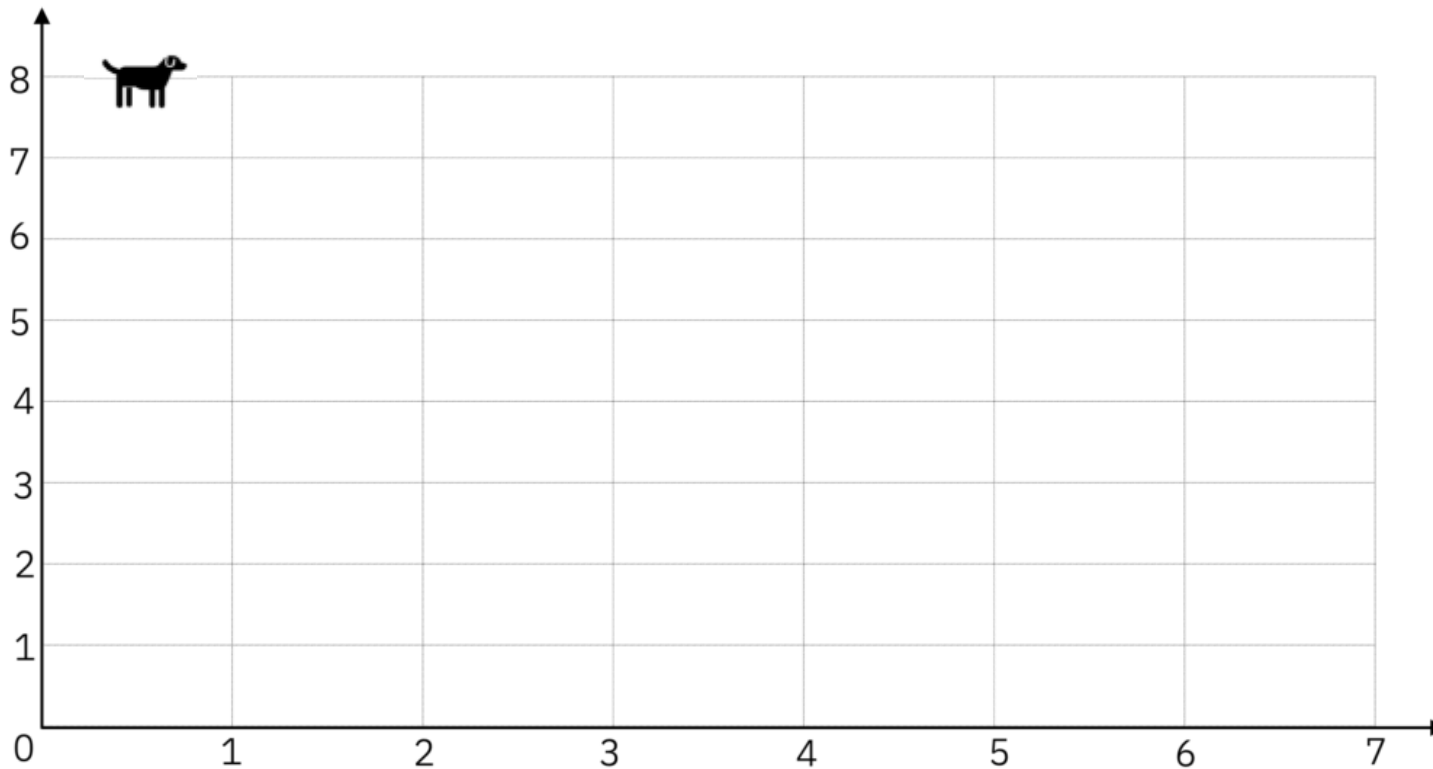
Generative AI

Embeddings



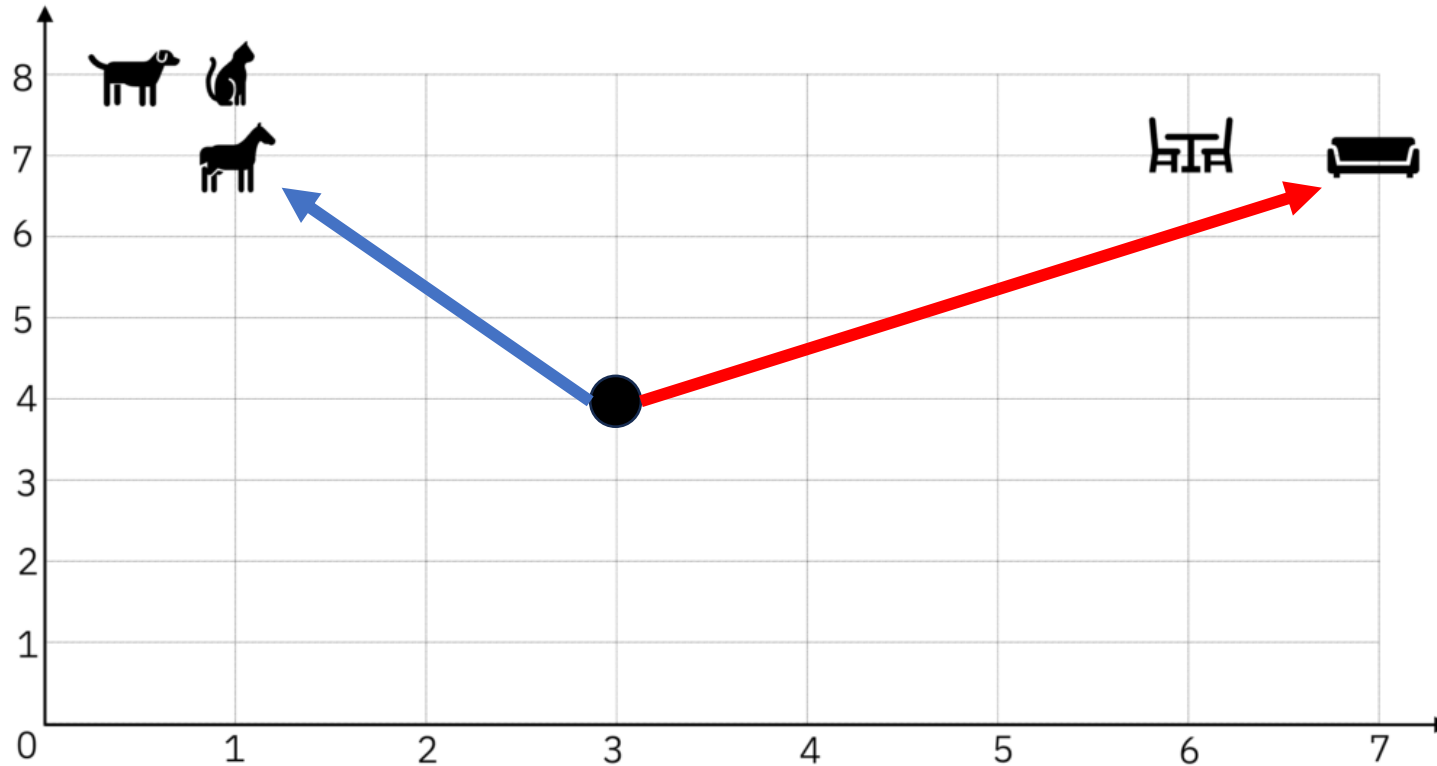
Generative AI

Embeddings



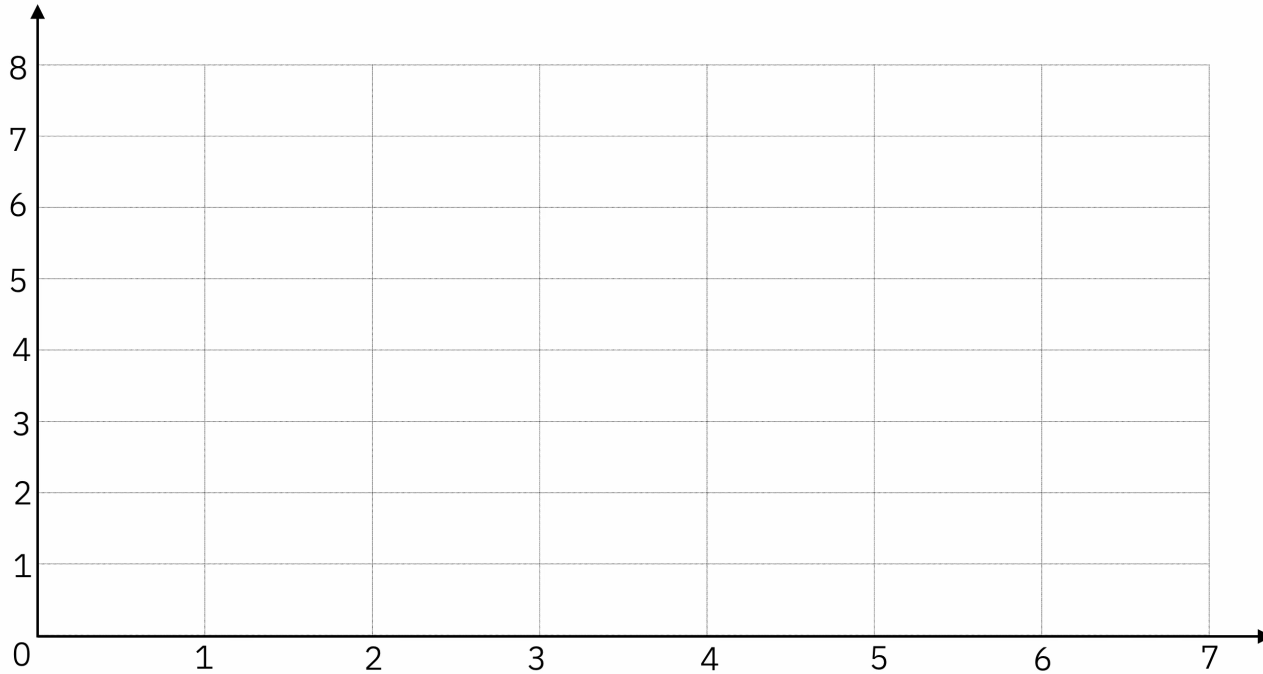
Generative AI

Embeddings



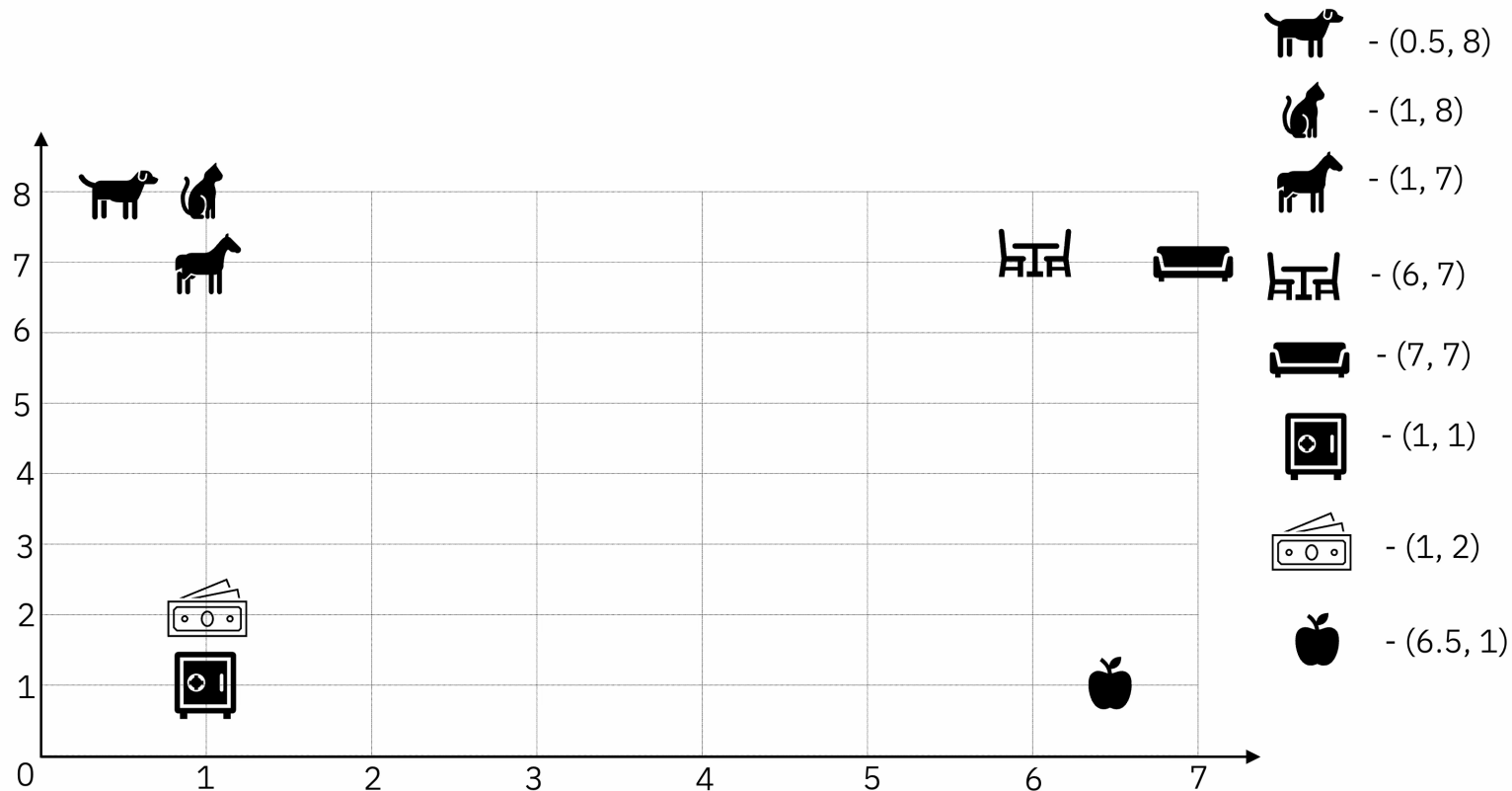
Generative AI

Embeddings



Generative AI

Embeddings



Generative AI



Embeddings



[-0.06113929, -0.0012407, 0.06087311, 0.01699911, 0.05108206, ..., 0.03732946, -0.00689885]



[-0.01101368, -0.04874269, -0.05087062, -0.02283244 0.01541347, ..., 0.06616838, 0.0045159]



[-0.05816573, -0.03017926, 0.05343566, -0.06409686, 0.0160787, ..., -0.0134629, -0.00547542]



[0.04290543, 0.04314668, 0.06709401, -0.02074, -0.0637757, ..., -0.01543431, -0.03469143]



[0.02085212, -0.04604341, -0.0511762, -0.05042295, -0.03493, 0.047325, ..., -0.06708, 0.01174]

512 ~ 4096 dimensões

Generative AI

Chunking

Definição:

- ✓ É um processo de dividir grandes partes de texto em segmentos menores.
- ✓ É essencial para a otimização da relevância do conteúdo que recebemos de um banco de dados vetorial quando usamos o LLM para incorporar embeddings ao conteúdo

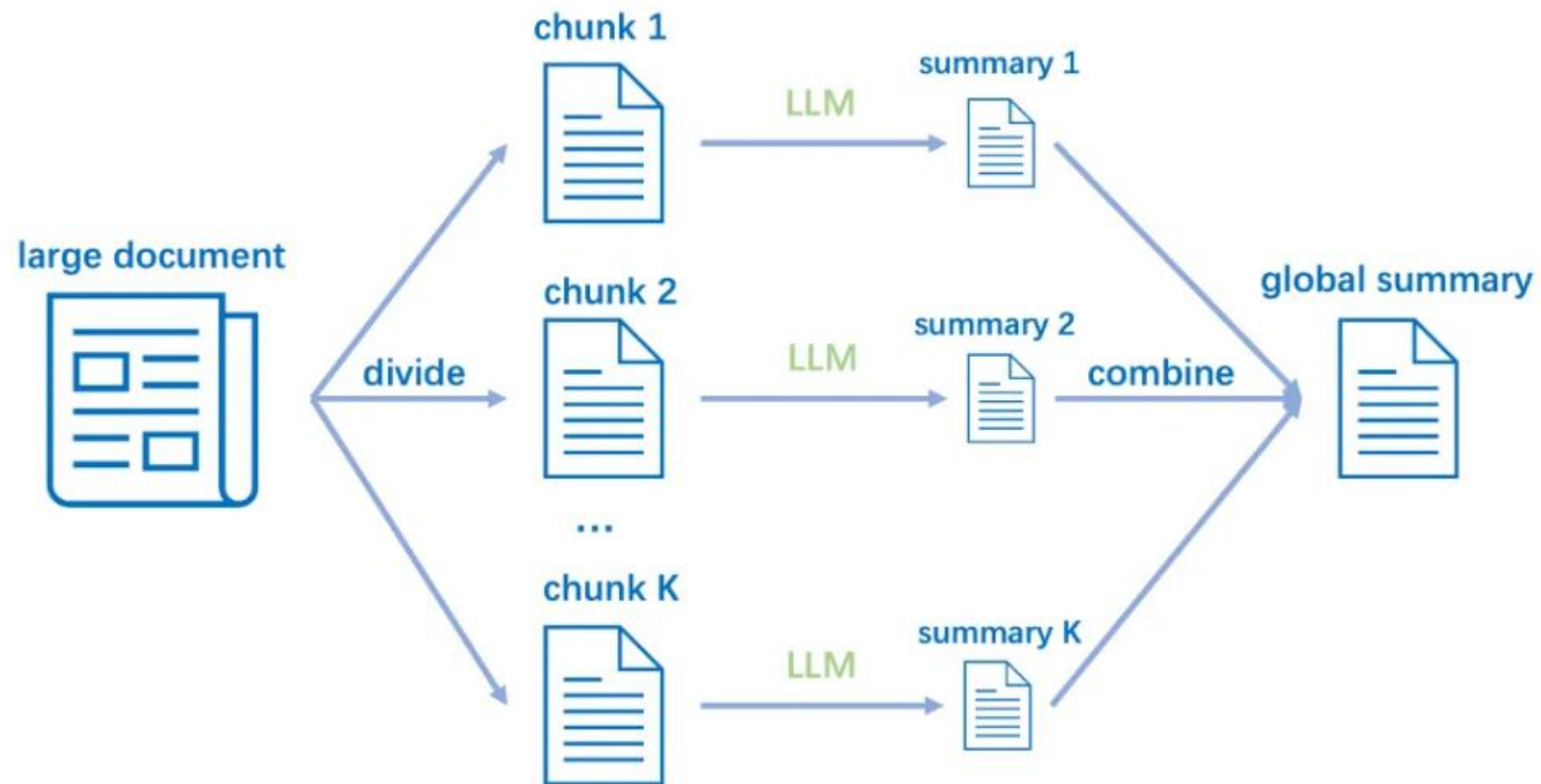
Objetivo:

- ✓ O objetivo do chunking é determinar se o contexto é realmente relevante para nosso prompt.
- ✓ Análise de sentenças para determinar sua utilidade para a análise sintática e semântica do prompt.
- ✓ É frequentemente usado para melhorar a compreensão da estrutura da sentença.



Generative AI

Chuncking



Generative AI

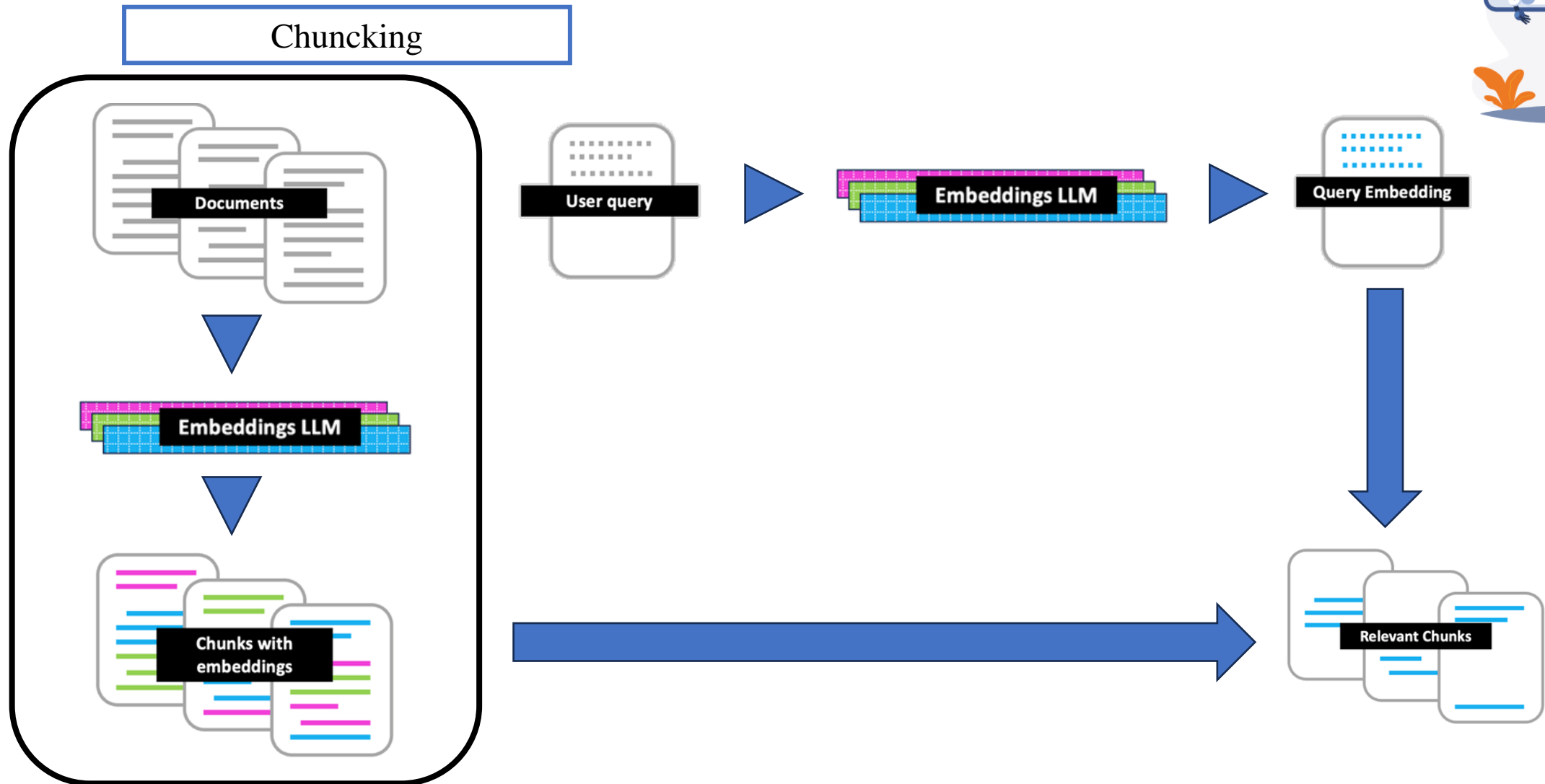
Chunking

Componentes Principais:

- ✓ **Técnica de divisão:** determina onde os limites dos blocos serão colocados — com base nos limites dos parágrafos, separadores específicos da linguagem de programação, tokens ou até mesmo limites semânticos
- ✓ **Tamanho do bloco:** o número máximo de caracteres ou tokens permitidos para cada bloco
- ✓ **Sobreposição de blocos:** número de caracteres ou tokens sobrepostos entre blocos; blocos sobrepostos podem ajudar a preservar o contexto entre blocos; o grau de sobreposição é normalmente especificado como uma porcentagem do tamanho do bloco



Generative AI



Generative AI

Chunking

✓ Token Fixo sem Sobreposição:

- Nesta técnica, os documentos são divididos em blocos de um número fixo de tokens, sem sobreposição.
- Essa abordagem funciona melhor quando há limites contextuais claros entre os chunks.

Exemplo:

Texto:

"A história começou. Era uma vez um dragão. Ele vivia em uma caverna."

Chunking:

["A história começou.", "Era uma vez um dragão.", "Ele vivia em uma caverna."]

Essa técnica pode resultar em perda de contexto, pois a transição entre os chunks é abrupta.



Generative AI

Chunking

✓ Token Fixo com Sobreposição

- Nesta técnica, os documentos são divididos em partes com um número fixo de tokens, mas com alguma sobreposição entre os chunks.
- Ajuda a manter o contexto.

Exemplo:

Texto:

"A história começou. Era uma vez um dragão. Ele vivia em uma caverna."

Chunking:

["A história começou. Era uma vez", "uma vez um dragão. Ele vivia em uma caverna."]

A sobreposição garante que informações importantes na transição não sejam perdidas.



Generative AI

Chunking

✓ Recursivo com Sobreposição

- Esta técnica divide os documentos usando delimitadores (como quebras de linha) e depois mescla recursivamente em chunks fixos.
- Isso mantém partes semanticamente relacionadas unidas.

Exemplo:

Texto:

"A história começou. Era uma vez um dragão. Ele vivia em uma caverna."

Chunking:

["A história começou. Era uma vez um dragão.", "Era uma vez um dragão. Ele vivia em uma caverna."]

Dessa forma, as frases que fazem sentido são mantidas juntas, melhorando a coerência.



Generative AI

Chunking

Texto:

"A história começou. Era uma vez um dragão. Ele vivia em uma caverna."

✓ **Token Fixo sem Sobreposição:**

["A história começou.", "Era uma vez um dragão.", "Ele vivia em uma caverna."]

✓ **Token Fixo com Sobreposição**

["A história começou. Era uma vez", "uma vez um dragão. Ele vivia em uma caverna."]

✓ **Recursivo com Sobreposição**

["A história começou. Era uma vez um dragão.", "Era uma vez um dragão. Ele vivia em uma caverna."]



Generative AI

Retrieval-Augmented Generation

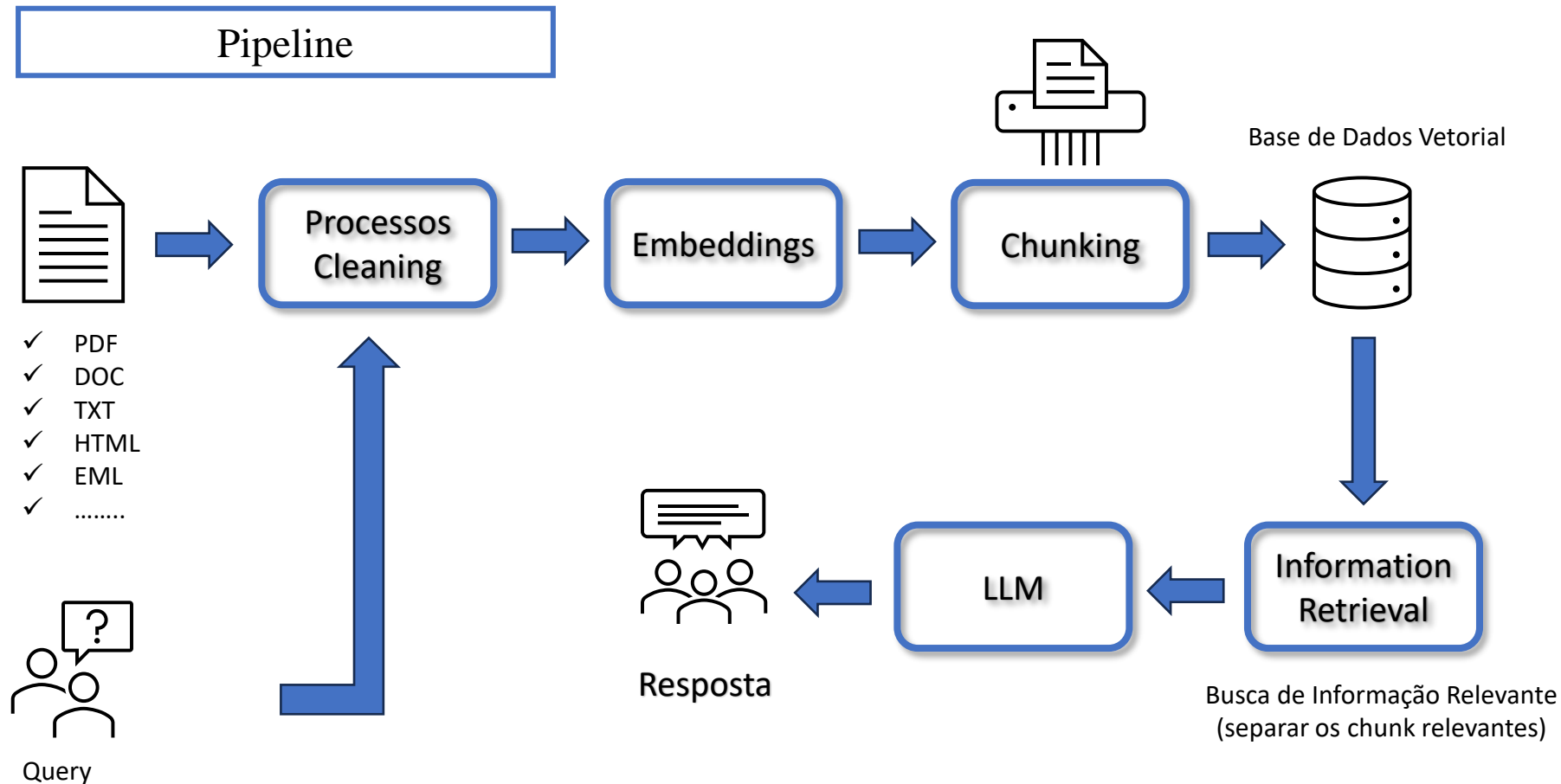
RAG



Retrieval-Augmented Generation (RAG) é o processo de otimizar a saída de um LLM, de forma que ele faça referência a uma **base de conhecimento** fora das suas fontes de dados de treinamento antes de gerar uma **resposta**.

- ✓ Implementação Econômica
- ✓ Informações Atualizadas
- ✓ Maior Confiança de Utilizadores
- ✓ Maior Controle na Etapa de Desenvolvimento

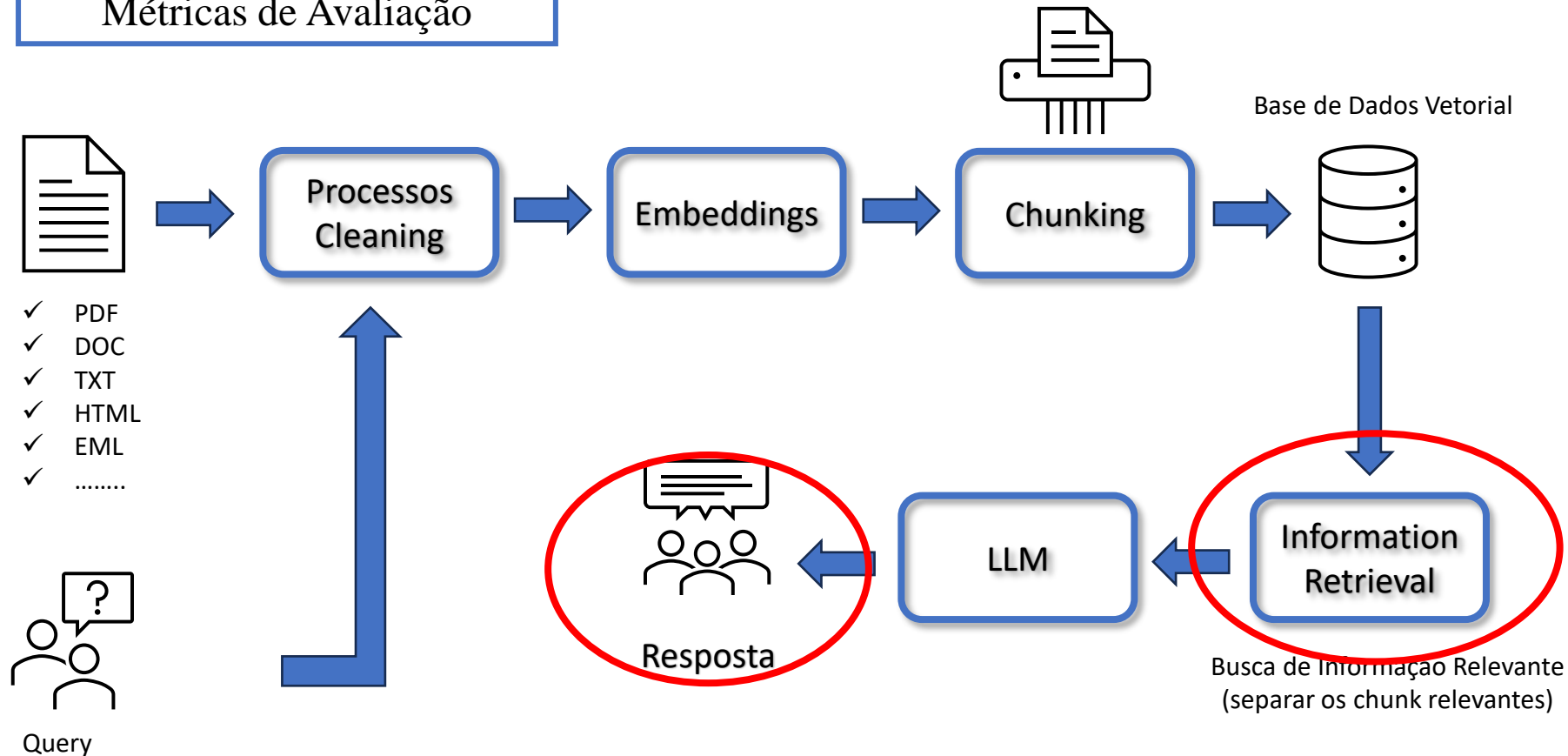
Retrieval-Augmented Generation



Retrieval-Augmented Generation



Métricas de Avaliação



Retrieval-Augmented Generation



Métricas de Avaliação

Information
Retrieval

- ✓ context_relevancy
- ✓ context_recall

LLM

- ✓ faithfulness
- ✓ answer_relevance

Retrieval-Augmented Generation



Métricas de Avaliação

LLM

Information
Retrieval

- ✓ context_relevancy
- ✓ context_recall

LLM

- ✓ faithfulness
- ✓ answer_relevance

Mede o grau de precisão em que a resposta gerada reflete as informações presentes no contexto fornecido (documento).

$$FF = \frac{\text{Nº de afirmações na resposta gerada que podem ser inferidas do contexto dado}}{\text{Nº total de afirmações na resposta gerada}}$$

Retrieval-Augmented Generation



Métricas de Avaliação

LLM

Information
Retrieval

- ✓ context_relevancy
- ✓ context_recall

Mede o grau de precisão em que a resposta gerada reflete as informações presentes no contexto fornecido (documento).

LLM

- ✓ faithfulness
- ✓ answer_relevance

Pergunta: Onde e quando Einstein nasceu?

Einstein nasceu na Alemanha.

Einstein nasceu em 14 de março de 1879.

Contexto: Albert Einstein (nascido em 14 de março de 1879) foi um físico teórico nascido na Alemanha, amplamente considerado um dos maiores e mais influentes cientistas de todos os tempos.

Alto FF: Einstein nasceu na Alemanha em 14 de março de 1879.

$$FF = \frac{1+1}{2} = 1$$

Baixo FF: Einstein nasceu na Alemanha em 20 de março de 1879.

$$FF = \frac{1+0}{2} = 0.5$$

Retrieval-Augmented Generation



Métricas de Avaliação

LLM

Information
Retrieval

- ✓ context_relevancy
- ✓ context_recall

LLM

- ✓ faithfulness
- ✓ answer_relevance

Mede o quanto a resposta gerada é pertinente para o contexto fornecido.

- Gera perguntas com base nas respostas (no mínimo 3)
- Compara as perguntas geradas com a pergunta apresentada

$$AR = \frac{1}{N} \sum_{i=1}^N \cos(E_{gerado}, E_{apresentada})$$

Retrieval-Augmented Generation



Métricas de Avaliação

LLM

Information
Retrieval

- ✓ context_relevancy
- ✓ context_recall

LLM

- ✓ faithfulness
- ✓ answer_relevance

Mede o quanto a resposta gerada é pertinente para o contexto fornecido.

Pergunta Fornecida: Onde fica a França e qual a sua capital?

Alto AR: França é na Europa Ocidental e sua capital é Paris.

Baixo AR: França é na Europa Ocidental.

Perguntas Geradas:

- **Pergunta 1:** "Em qual parte da Europa a França está localizada?"
- **Pergunta 2:** "Paris é a capital de que país da Europa?"
- **Pergunta 3:** "Você pode identificar a região da Europa onde a França está situada?"

Mestrado em Engenharia e Gestão Industrial

Inteligência Artificial aplicada na Indústria

Daniel Nogueira



dnogueira@ipca.pt



<https://www.linkedin.com/in/danielfnogueira/>

