Tratamento de dados - sequência correta

✅ Etapas do tratamento de dados textuais (com exemplos):

Suponha o texto original:

```
text
CopiarEditar
"Os CLIENTES estavam RECLAMANDO do atendimento no banco, mas o B
ANCO não respondeu!!!"
```

📌 Etapa 1: Lowercase (minúsculas)

👉 Por quê?

Para normalizar palavras que são semanticamente iguais, mas escritas com maiúsculas.

```
python
CopiarEditar
"Os CLIENTES estavam RECLAMANDO do atendimento no banco, mas o B
ANCO não respondeu!!!"
→ "os clientes estavam reclamando do atendimento no banco, mas o banc
o não respondeu!!!"
```

📌 Etapa 2: Tokenização

👉 Por quê?

Para separar o texto em unidades (palavras), facilitando filtros e transformações.

```
python
CopiarEditar
→ ['os', 'clientes', 'estavam', 'reclamando', 'do', 'atendimento', 'no', 'banco',
```

',', 'mas', 'o', 'banco', 'não', 'respondeu', '!!!']

Obs: spaCy já faz isso ao usar `nlp(text)`.

📌 Etapa 3: Remoção de pontuação e símbolos

👉 Por quê?

Pontos, vírgulas, exclamações etc. geralmente **não carregam significado semântico** útil para embedding.

python
CopiarEditar
→ ['os', 'clientes', 'estavam', 'reclamando', 'do', 'atendimento', 'no', 'banco', 'mas', 'o', 'banco', 'não', 'respondeu']

📌 Etapa 4: Remoção de stopwords

👉 Por quê?

Stopwords como “os”, “do”, “no”, “mas”, “o” aparecem com muita frequência e **não ajudam a identificar o tema** do texto.

python
CopiarEditar
→ ['clientes', 'estavam', 'reclamando', 'atendimento', 'banco', 'banco', 'não', 'respondeu']

📌 Etapa 5: Lematização

👉 Por quê?

Para reduzir palavras à sua forma base, mantendo o **significado mais geral**, diferente de stemização (que pode cortar demais).

python
CopiarEditar

→ ['cliente', 'estar', 'reclamar', 'atendimento', 'banco', 'banco', 'não', 'responder']

🔄 Por que seguir nessa ordem?

Ordem	Etapa	Consequência se inverter
1	Lowercase	Se deixar pro final, “Banco” ≠ “banco”
2	Tokenização	Sem tokenizar, não conseguimos remover stopwords direito
3	Remover pontuação	Ponto e vírgula quebram tokens inúteis
4	Remover stopwords	Melhor após tokenizar para garantir acerto
5	Lematização	Precisa dos tokens já limpos e sem stopwords para ser mais relevante

---

🧪 Resultado final ideal:

python
CopiarEditar
['cliente', 'estar', 'reclamar', 'atendimento', 'banco', 'banco', 'não', 'responder']

Isso representa **a essência do texto**, pronto para:

- Geração de embeddings
- Agrupamento por similaridade
- Classificação etc.