**📌— Pesquisa de APIs Open Source para o Projeto**

**Tema: *Querido Diário – Radar das Tecnologias de Investimento de IA nos Municípios*  
Foco: estudo das APIs/bibliotecas open source de IA que podem ajudar a filtrar publicações no Diário Oficial sobre investimentos em educação tecnológica.**

**APIs/Bibliotecas analisadas**

**Estudo de APIs Open Source para o Projeto**

**🔹 SpaCy**

* **O que é:** Uma biblioteca de processamento de linguagem natural (NLP).
* **Por que usar:** É ótima **para extrair informações específicas de textos longos**, como nomes de órgãos públicos, valores em reais e termos como *“educação tecnológica”*.
* **No projeto:** Podemos usá-la para **filtrar automaticamente os trechos do Diário Oficial** que falam sobre investimento e ignorar o que não é relevante.

**🔹 Hugging Face Transformers**

* **O que é:** Um conjunto de modelos de inteligência artificial já treinados para entender textos em várias línguas, inclusive português.
* **Por que usar:** Permite classificar se um texto é relevante ou não e até gerar resumos curtos dos atos normativos.
* **No projeto:** Seria útil para criar um “filtro inteligente” que decide se um ato fala de educação tecnológica e ainda produzir resumos diários prontos para leitura.

**🔹 Haystack**

* **O que é:** Um framework para busca semântica em documentos.
* **Por que usar:** Com ele, é possível fazer perguntas em linguagem natural como: *“Quais atos falam sobre investimento em educação tecnológica em 2025?”*
* **No projeto:** Poderia virar a base de uma ferramenta de busca interna, ajudando a consultar rapidamente os diários já analisados.

**🔹 NLTK**

* **O que é:** Uma biblioteca mais tradicional de NLP.
* **Por que usar**: Boa para tarefas básicas de limpeza de texto e protótipos rápidos.
* **No projeto:** Pode ser útil como apoio no pré-processamento antes de aplicar ferramentas mais avançadas.

**🔹 LLMs Open Source (**ex.: GPT4All, LLaMA 2, Mistral**)**

* **O que são:** Modelos de linguagem avançados que podem rodar localmente.
* **Por que usar**: Conseguem entender contexto e produzir resumos mais naturais dos textos.
* **No projeto:** Poderiam ser usados futuramente para automatizar ainda mais a geração dos diários, deixando os resumos em linguagem simples e direta.

**Conclusão**

* Primeira prioridade (fáceis e aplicáveis): **SpaCy + Hugging Face Transformers.**
* Segunda prioridade (avançado**): Haystack (busca semântica/QA).**
* Exploração futura: **LLMs open source (resumos mais “naturais”).**
* Baixa prioridade: **NLTK (mais didático do que prático).**

**Estrutura da Primeira API**

**🔹 O que vamos usar**

* **FastAPI** → para criar a API (leve e fácil de usar).
* **spaCy** → para análise de texto (open source).

**🔹 Como funciona**

1. **Entrada:** texto de um ato do Diário Oficial.
2. **Processamento:**
   * Regras simples de palavras-chave (ex.: *educação tecnológica, capacitação digital*).
   * spaCy identifica órgãos, valores e datas.
3. **Saída:** JSON estruturado com os resultados.

**Fluxo básico de funcionamento:**

1. O usuário envia o texto de um ato (por exemplo, uma publicação do Diário Oficial) para a API.
2. A API aplica duas camadas de análise:
   * **Regras simples de palavras-chave** (como “educação tecnológica”, “capacitação digital”, “laboratório de informática”).
   * **Processamento com spaCy**, que identifica automaticamente entidades como órgãos públicos, valores em reais e datas.
3. A API combina essas informações e decide se o texto é **relevante ou não**, atribuindo uma pontuação.
4. Por fim, a API gera uma resposta em **JSON**, padronizada, com os seguintes campos:
   * Relevante (sim/não)
   * Score (grau de confiança)
   * Órgão
   * Valor
   * Data
   * Palavras-chave encontradas
   * Resumo em poucas linhas