ENTER↵

ENTER AI FELLOWSHIP

TAKE HOME PROJECT

PRAZO: 07/NOV, MEIO DIA

OBJETIVO

Criar uma solução capaz de extrair informações estruturadas de arquivos PDF de forma síncrona e com o melhor custo-benefício e acurácia possível.

DESCRIÇÃO GERAL

Você deverá construir uma solução que será utilizada para **extrair dados estruturados de milhares de documentos**, devendo:

- Responder cada requisição em menos de 10 segundos;
- Minimizar o custo monetário de execução ao máximo;
- Garantir consistência e precisão das entidades extraídas (todas as respostas são únicas para cada documento).

Para cada documento, existem três parâmetros de entrada:

1. *label* → identifica o tipo de documento (ex.: "fatura", "contrato", "formulário").
2. *extraction_schema* → um dicionário em que as chaves são os nomes dos campos a serem extraídos e os valores são descrições dos campos. (i.e “cpf” : “número de identificação de pessoa física em formato XXX.XXX.XXX-X”)
3. *pdf* → o arquivo PDF

O objetivo é extrair as informações do PDF no formato definido pelo schema. O seu sistema pode acumular conhecimento a cada solicitação.

Restrições do problema:

- Cada PDF possui apenas uma página.
- O PDF já vem com OCR feito, ou seja, todo o texto já está embutido na página.
- Você não conhece antecipadamente os labels e nem seus respectivos schemas completos.
- Para cada label, o seu schema completo correspondente é fixo.
- Os campos do schema especificado como parâmetro de um documento *extraction_schema* pertencem ao schema completo do label.

  - Exemplo: considere um label fictício “RG” com schema completo *{nome, nome_mae, nome_pai, numero_rg, data_nascimento}*. Lembre-se: você não conhece o schema completo.
  - Numa primeira requisição, você pode receber um documento com label “RG” e *extraction_schema = {nome, nome_mae, data_nascimento}*
  - Numa segunda requisição, você pode você pode receber um documento com label “RG” e *extraction_schema = {nome, nome_pai, numero_rg, data_nascimento}*
- Os documentos de um mesmo label seguem um template padrão, mas:
  - Alguns (como formulários) têm estrutura idêntica entre arquivos;
  - Outros (como contratos ou faturas) mantêm o layout geral, mas as informações podem variar de posição.

DADOS PARA DESENVOLVIMENTO

Este repositório possui alguns exemplos de documentos com seus labels e *extraction_schemas* correspondentes. Você encontrará exemplos de três labels diferentes, podendo utilizá-los livremente para treinar, testar e ajustar sua solução.

ENTREGÁVEL

- Todo o código no Github ou equivalente
- Um arquivo README.md explicando:
  - Quais desafios você mapeou, quais decidiu endereçar e qual a sua solução proposta para cada um deles - aqui é o espaço para você colocar em palavras toda a criatividade que você aplicou.
  - Como utilizar a sua solução.
- Ter uma solução que recebe uma requisição com (label, schema, pdf) e retorna os dados extraídos estruturados.

Você pode entregar sua solução em qualquer formato funcional, como:

- Um endpoint de API (por exemplo, /extract)
- Uma interface web simples (UI)
- Um script executável

O QUE NOS INTERESSA

Extrair dados com LLMs virou commodity. Queremos ver a **criatividade técnica** para desenvolver estratégias eficientes que:

1. Reduzam chamadas ao LLM (cache, pattern matching, OCR parsing local)
2. Mantenham 80%+ de acurácia mesmo com layouts variáveis

3. Respondam em <10s por requisição
4. Minimizem custo monetário

Exemplos de desafios interessantes que você pode querer endereçar:

- Balancear precisão vs custo: trade-off entre usar heurísticas rápidas e LLM. Exemplos:
    - Soluções criativas para reduzir custo (e.g., pré-processamento, cache, heurísticas etc.)
    - Como usar LLMs e outras técnicas de NLP
    - Otimizar o contexto.
- Lidar com variabilidade: documentos do mesmo label podem ter layouts diferentes

CRITÉRIOS DE AVALIAÇÃO

Desenvolvemos um conjunto de dados interno contendo milhares de exemplos que serão usados para avaliar a sua solução. Para cada item, avaliaremos a sua solução e mediremos:

- Tempo de resposta: as extrações/respostas devem ser retornadas em média, em menos de 10 segundos
- Precisão: Pelo menos 80% dos campos devem ser extraídos corretamente:
    - 1 caractere errado = campo errado
    - A avaliação não é case sensitive (ENTER = enter = EnTer != inter)
- Custo: Qual foi o custo médio para realizar a extração de cada documento.

Diferenciais

Além dos fatores acima, nós também vamos levar em consideração qualquer tentativa de ir além do escopo óbvio previsto, como por exemplo:

- Uma UI funcional e intuitiva
- Velocidade excepcional (extrações em menos de 0.1 segundo)

Infraestrutura e Restrições

- Você receberá uma API key da OpenAI com limite de budget.
- Os créditos são suficientes para o desafio, mas, se necessário, você pode solicitar mais.
- Para chamadas de LLM, use exclusivamente o modelo gpt-5 mini (link)

Dica: o custo das chamadas ao LLM é o upper bound do custo total do seu sistema — otimizar o uso do modelo é parte essencial do desafio.

ENTER↵

FAQ

1. Preciso treinar um modelo?
→ Não. O foco é em construir a solução de ponta a ponta de forma criativa, não em treinamento de modelos.

2. Preciso me preocupar com extrair o texto/fazer OCR dos documentos?
→ Não.

3. Posso usar quaisquer linguagens ou bibliotecas externas?
→ Sim - qualquer linguagem, biblioteca ou framework é permitida.

4. Posso usar embeddings ou outras técnicas de NLP?
→ Sim, desde que a execução permaneça dentro do limite de custo.

5. E se o extraction_schema pedir um campo que não existe no PDF?
→ Retorne null para esse campo.

6. Posso fazer cache de resultados de PDFs já processados?
→ Sim, desde que seja apenas entre requisições na mesma sessão.

7. Se eu chamar um LLM, vai contar no tempo de 10s?
→ O tempo a ser considerado será o tempo decorrido desde a ação considerada como o envio até a recepção da resposta. Ou seja, o que acontece assincronamente e em background é irrelevante.

8. Os labels na avaliação serão os mesmos que os de desenvolvimento?
→ Não necessariamente. Seu sistema deve ser adaptativo. Lembre-se: você não conhece os labels nem os *extraction_schema* de antemão.

9. Como devo entregar a solução?
→ Repositório GitHub com README detalhando sua abordagem e instruções de execução.

10. Onde estão os dados para eu testar minha solução?
→ Eles estão no seguinte repositório do GitHub.

10. Preciso criar uma API?
Qualquer formato é aceitável, desde que seja fácil de usar. Por exemplo:

1. Aplicação nativa: Deve permitir a passagem de parâmetros em lote, especificando o caminho da pasta com os arquivos.
2. Aplicação web + API: Deve permitir a passagem de parâmetros em lote, especificando o caminho da pasta com os arquivos.
3. Script / CLI: Deve permitir a passagem de parâmetros em lote, especificando o caminho da pasta com os arquivos.

É importante que o processamento em lote seja feito em série, ou seja, cada linha deve ser processada independentemente das seguintes. O primeiro input de um lote deve ser retornado em menos de 10 segundos.

ENTER↵

EXEMPLO 1

→ Label: carteira_oab

→ PDF:

→ Schema:

```
JSON
{
  "nome": "Nome do profissional, normalmente no canto superior esquerdo da
imagem",
  "inscricao": "Número de inscrição do profissional",
  "seccional": "Seccional do profissional",
  "subsecao": "Subseção à qual o profissional faz parte",
  "categoria": "Categoria, pode ser ADVOGADO, ADVOGADA, SUPLEMENTAR,
ESTAGIARIO, ESTAGIARIA",
  "telefone_profissional": "Telefone do profissional",
  "situacao": "Situação do profissional, normalmente no canto inferior
direito."
}
```

→ Output:

```JSON
{
  "nome": "SON GOKU",
  "inscricao": "101943",
  "seccional": "PR",
  "subsecao": "Conselho Seccional - Paraná",
  "categoria": "Suplementar",
  "telefone_profissional": null,
  "situacao": "Situação Regular"
}
```

EXEMPLO 2

→ Label: carteira_oab

→ Schema:

```
JSON
{
  "nome": "Nome do profissional, normalmente no canto superior esquerdo da imagem",
  "inscricao": "Número de inscrição do profissional",
  "seccional": "Seccional do profissional",
  "subsecao": "Subseção à qual o profissional faz parte",
  "categoria": "Categoria, pode ser ADVOGADO, ADVOGADA, SUPLEMENTAR, ESTAGIARIO, ESTAGIARIA",
  "endereco_profissional": "Endereço profissional completo",
  "situacao": "Situação do profissional, normalmente no canto inferior direito."
}
```

→ PDF:

→ Output:

```JSON
{
  "nome": "JOANA D'ARC",
  "inscricao": "101943",
  "seccional": "PR",
  "subsecao": "Conselho Seccional - Paraná",
  "categoria": "Suplementar",
  "endereco_profissional": "Avenida Paulista, Nº 2300, andar Pilotis, Bela Vista, São Paulo - SP, 01310300",
  "situacao": "Situação Regular"
}
```