Data Annotations Engineer

Veryfi is a YC-funded Silicon Valley startup that uses AI to understand documents like receipts and invoices. As a Data Annotations Engineer at Veryfi, you'll contribute to the evolution of our training data infrastructure and the development of new features and projects. You'll gather, process, and analyze diverse datasets to generate high-quality training data for our machine-learning models. Furthermore, by delving deep into our system, you'll have the autonomy to identify challenges and opportunities, taking ownership of developing solutions to refine existing tools and algorithms.

Keywords: NLP, Patterns Detection, Data Labeling, Software Development, Data Engineering.

Key Responsibilities:

- Gather, process, and analyze diverse datasets to generate training data that fuels the development of our ML projects.
- Expand and optimize the training data pipelines to improve the speed and accuracy of our processes.
- Collaborate with a cross-functional team to define requirements and prioritize development efforts.

Essential Skills:

- Proficient in Python programming for data handling and processing, with experience in utilizing data science tools such as Pandas, NumPy, SciPy, and others.
- Strong analytical thinking with a focus on delivering results.
- Meticulous attention to detail, ensuring accuracy and precision in all data handling and processing tasks.
- Enthusiastic about learning and adapting to new technologies and methodologies, particularly in the realm of Machine Learning (ML).
- Innovation mindset, adept at challenging existing processes and driving positive change.

Preferred Qualifications:

- Familiarity with regex development, software engineering principles, and Linux command line tools.
- Experience with Natural Language Processing (NLP) techniques and libraries, including the use of Large Language Models (LLMs) and supervised learning for document data extraction.
- Effective organizational abilities, capable of managing projects independently from inception to completion.
- Exceptional verbal and written communication skills, effectively communicating problems, proposed solutions, and results to stakeholders in a multicultural environment.
- A Bachelor's degree in computer science, engineering, or a related field. Postgraduate studies are a plus but not required.

Technical test

We need you to complete an important step in training supervised machine learning models: labeling information. We have provided you with a new set of documents (attached), and your task is to accurately extract the following information in a JSON format:

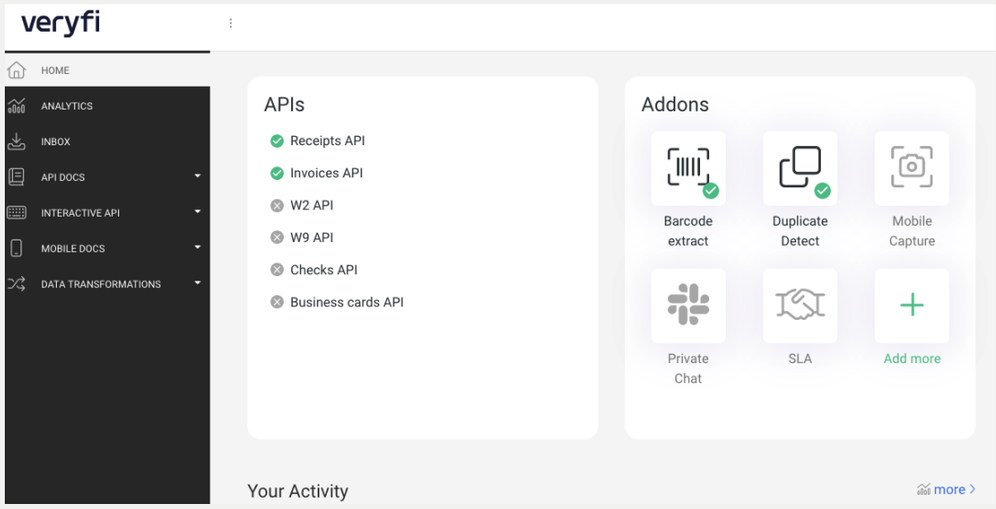
- vendor name
- vendor address
- bill to name
- invoice number
- date
- For each line item, capture:
  - sku
  - description
  - quantity
  - tax_rate
  - price
  - total

If you need clarification about the fields requested, please visit: https://faq.veryfi.com/en/articles/5571268-document-data-extraction-fields-explained

> 💡 *"An invoice line item is a single entry on an invoice. For example, an invoice for 10 red books at $1.00 each, and 20 blue books at $3.00 each, would be considered to have two invoice line items."*
>
> *Source*.

1. Create an `OCR API` Veryfi account at https://hub.veryfi.com/signup/

> ⚠️ Please note that creating a Receipts OCR and Expenses App account will not allow you to use our APIs. To ensure access to our APIs make sure that the left pane in your hub account is gray. If the pane is green or pink, then you created the wrong account type.



> 💡 Be aware that the free-trial account comes with certain restrictions. Please send us the email address you used to create your account, and we will increase your account's limitations accordingly. This will enable you to test everything you need without any roadblocks.

2. Create a Python-based system to extract the required information.
    a. Use Veryfi's Python API to get the OCR output for each document.

       https://github.com/veryfi/veryfi-python

       - The OCR output should be under `ocr_text` within the API response. You can ignore everything else because that's the information we are trying to improve with your annotations.

    b. Develop an automatic solution that extracts the requested information in a JSON format out of the `ocr_text`. Be thorough describing how the solution extracts the data in the provided documents. Also, the solution should support any document with the same format while excluding any other documents. Test the exclusion processing a document of your own.
3. Include in your solution a file describing your approach, assumptions, and the coding best practices you implemented in detail.
    - Code paradigm
    - Unit tests
    - etc...
4. Send the link to the repository of your solution using the same form you used to access this document.