Hypatos.ai Coding Challenge

Congratulations, you made it through the initial interviews. Now it's time to show off your coding skills. The following task should be self-explanatory, but if you have any questions don't hesitate to contact me at cem.dilmegani@hypatos.ai.

Your Task: Comparison Service

During our client work, we need to measure how accurate our invoice extraction models are for line item extraction. We built a Python script to achieve this and ask for you to do the same.

Please push the source code to a Github repository for review.

Terminology

Important terminology include:

- Extraction results: The data that our model automatically extracts from documents
- Ground truth: The actual data in the invoice
- Invoice: An image that contains an invoice
- File: The file that includes the invoice. Assume every file includes only 1 invoice.
- Header information: The text information that you find in the header or footer sections of the invoice
- Line item information: Table information in the invoice. Most invoices contain line items that describe the product or service that has been sold.
- Line items include these entities. Each entity can be NULL or contain values:
    - Description (String)
    - Quantity (Float)
    - Unit price (Float)
    - Total price (Float)

Spec

Accuracy measurement is trivial for header information. This is because these fields all have unique identifiers which are known before invoice extraction. For example, the sender of a specific file, is a data entity which has to exist in all invoices.

However, it is more complex to measure accuracy for line items. This is because a document could include 0 or any positive integer number of line items. We need to measure our accuracy for line items as accurately as possible.

Please include the Python code to do this comparison. We will test the code with 2 data sets following the same format as the attached data sets.

Output will include

- a score for overall accuracy of the model. 0 indicates no errors and higher scores indicate higher error levels
- an evaluation of each data entity (cell) in the ground truth in CSV format. Assume that all cells are of equal value to us. Please see the output format for more details

Technical spec

We believe there is no one-size-fits-all technology. Good engineering is about using the right tool for the right job, and constantly learning about them. Therefore, feel free to mention in your README how much experience you have with the technical stack you choose, we will take note of that when reviewing your challenge.

Here are some technologies we are more familiar with:

- Python
- JavaScript
- PHP
- Go
- Java

How we review

We will take into consideration your experience level.

We value quality over feature-completeness. It is fine to leave things aside provided you call them out in your project's README. The goal of this code sample is to help us identify what you consider production-ready code. You should consider this code ready for final review with your colleague, i.e. this would be the last step before deploying to production.

The aspects of your code we will assess include:

- Business understanding: This assessment is about comparing output of an automated document extraction with the actual data on the document. We are interested in evaluating accuracy like how our clients would evaluate it.
- Clarity: Does the README clearly and concisely explains the problem and solution? Are technical tradeoffs explained?
- Correctness: does the application do what was asked? If there is anything missing, does the README explain why it is missing?

- Code quality: is the code simple, easy to understand, and maintainable? Are there any code smells or other red flags? Is the coding style consistent with the language's guidelines? Is it consistent throughout the codebase?
- Testing: how thorough are the automated tests? Will they be difficult to change if the requirements of the application were to change? Are there some unit and some integration tests?
- We're not looking for full coverage (given time constraint) but just trying to get a feel for your testing skills.
- Technical choices: do choices of libraries, databases, architecture etc. seem appropriate for the chosen application?
- Does your README contain information on how to run it? Bonus point (those items are optional):
- Scalability: will technical choices scale well? If not, is there a discussion of those choices in the README?
- Production-readiness: does the code include monitoring? Logging? proper error handling?