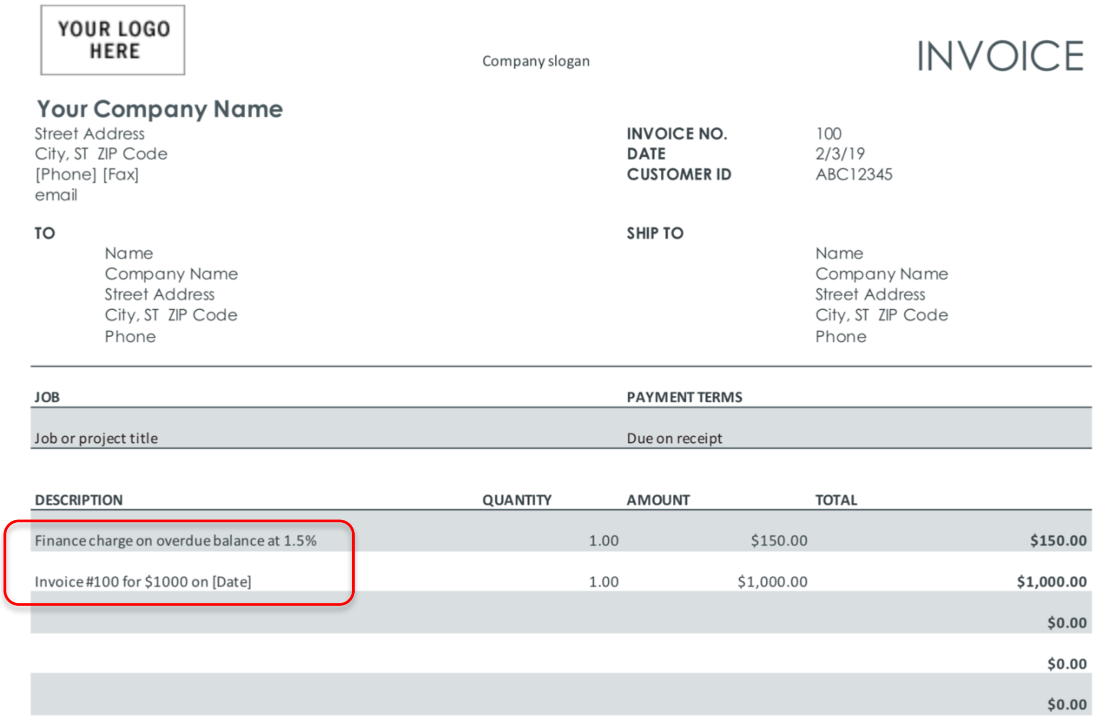
Thank you for considering the Data Scientist position at Hypatos. This take-home test is designed to gather more information about your experience and get a sense of how you approach problems we care about. Feel free to send questions directly to he.zhang@hypatos.ai if anything is unclear. Please also acknowledge the receipt of the email. The test should take you no more than 4 hours (model training time not included) since we want to be respectful of your time. Our intention of the test is to also use this problem for further discussions on the on-site interview.

## Problem description

You are presented with dataset that contains small subset of Hypatos labelled invoices in the past two years. Those invoices have been processed via our Optical Character Recognition (OCR) system and outputted in json format. Our goal is to build a classification model that extracts accounting information from an invoice. To this end, the first step is to find the line items from the dataset. See below an example invoice and the line item.



YOUR LOGO HERE

Company slogan

INVOICE

**Your Company Name**
Street Address
City, ST ZIP Code
[Phone] [Fax]
email

| | |
|---|---|
| **INVOICE NO.** | 100 |
| **DATE** | 2/3/19 |
| **CUSTOMER ID** | ABC12345 |

**TO**
Name
Company Name
Street Address
City, ST ZIP Code
Phone

**SHIP TO**
Name
Company Name
Street Address
City, ST ZIP Code
Phone

| JOB | PAYMENT TERMS |
|---|---|
| Job or project title | Due on receipt |

| DESCRIPTION | QUANTITY | AMOUNT | TOTAL |
|---|---|---|---|
| Finance charge on overdue balance at 1.5% | 1.00 | $150.00 | **$150.00** |
| Invoice #100 for $1000 on [Date] | 1.00 | $1,000.00 | **$1,000.00** |
| | | | **$0.00** |
| | | | **$0.00** |
| | | | **$0.00** |

## About dataset

Dataset is contained in two json files: hypatos-ds-train.json and hypatos-ds-test.json, both included in the assignment. Each file contains a set of invoice data in json format, e.g.

```
{⊟
  "id": 456561110,
  "words": [⊟
  {⊟
    "value": "gqeUrQ==",
    "region": {⊟
      "left": 0.27226892,
      "top": 0.032104637,
      "width": 0.03529412,
      "height": 0.0083234245,
      "page": "1"
    }
  },
  {⊟
    "value": "dKGlmYCFXw==",
    "region": {⊟
      "left": 0.27226892,
      "top": 0.058263972,
      "width": 0.0605042,
      "height": 0.007134364,
      "page": "1"
    }
  },
  {⊟
    "value": "eqCW",
    "region": {⊟
      "left": 0.33613446,
      "top": 0.058263972,
      "width": 0.020168068,
      "height": 0.007134364,
      "page": "1"
    }
  }
],
"entities": [⊟
  {⊟
    "metaData": {⊟
      "region": {⊟
        "page": 1
      }
    },
    "label": "item",
    "indices": [⊟
      0,
      1
    ]
    }
  ]
}
```

Here “id” denotes the invoice ID, and “words” is a list of words (generated by OCR) in current invoice. Each word contains the “value” (text) data (for confidentiality purpose, the characteristic is anonymized, which might slightly affect the prediction accuracy). The bounding box information of the text, i.e. “region” is included. In addition, “page” (page number) can be found below the bounding box data, as an invoice may contain multiple pages. We labelled words that are items in “entities”. For example, in the above sample, words "gqeUrQ==" and "dKGlmYCFXw==" are items, whereas "eqCW" is not.

You can download the dataset in the email attachment.

**Your task**

Your main task is to implement an approach to detect items from an input invoice. There are no constraints on programing language/framework. Please provide:

- programming code for the solution and (if not obvious) instruction how to run it
- your reasoning about the problem and the solution you have tried
- predictions for the test dataset and performance analysis

For first two parts you can pick your own format: can be code+separate document, commented code, IPython notebook etc.
Note: since expected time for this assignment is ~ 4 hours, we do not expect optimal solution but something that yields meaningful predictions.

**Questions to think about:**

1) What is good measure for classification accuracy?
2) What are possible shortcomings and extensions of your implementation?
3) How would you design a real-time performance system that responds to a high volume of prediction requests efficiently.