Progress report For february 5th:

1.What I did:
Ensured dataset formatting,and data labelling(need to finish this faster).Along with having each scenario along with documentation as its own record.

Formatted dataset to match the format as shown in pretraining data.

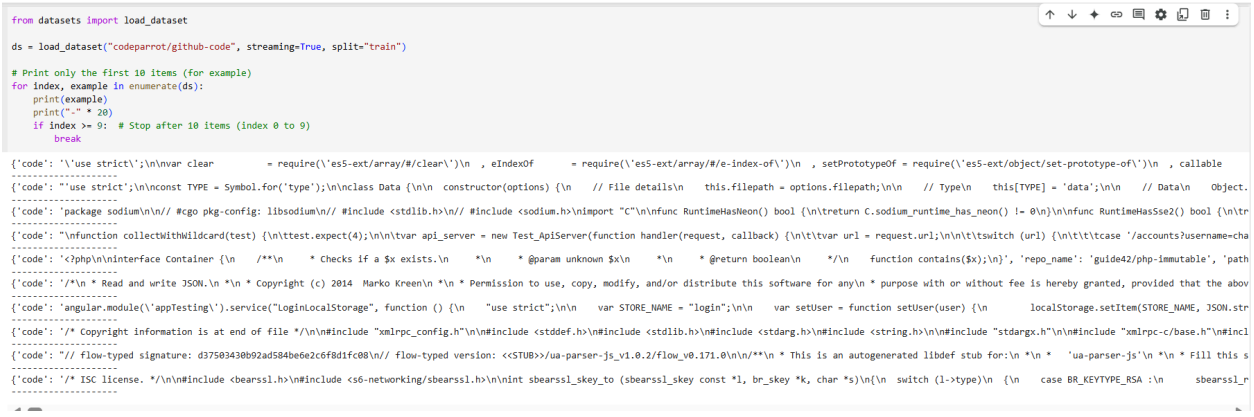
2.Process:

2.1First looked into existing datasets found on huggingface.com.

Looked at the pretraining dataset-> codeparrot/github-code · Datasets at Hugging Face .

Further looked into datasets related to the codet5p model->subbu264/codeT5p-19k · Datasets at Hugging Face and stojchet/python-codet5p-770m-empty · Datasets at Hugging Face dataset.

2.2 Outputted datasets from hugging face.com to ensure data formatting(More complete outputs can be seen in this directory dataFormat.txt):



```
from datasets import load_dataset

ds = load_dataset("codeparrot/github-code", streaming=True, split="train")

# Print only the first 10 items (for example)
for index, example in enumerate(ds):
    print(example)
    print("-" * 20)
    if index >= 9:  # Stop after 10 items (index 0 to 9)
        break
```

```
{'code': '\'use strict\';\n\nvar clear          = require(\'es5-ext/array/#/clear\')\n  , eIndexOf       = require(\'es5-ext/array/#/e-index-of\')\n  , setPrototypeOf = require(\'es5-ext/object/set-prototype-of\')\n  , callable
--------------------
{'code': "'use strict';\n\nconst TYPE = Symbol.for('type');\n\nclass Data {\n\n  constructor(options) {\n    // File details\n    this.filepath = options.filepath;\n\n    // Type\n    this[TYPE] = 'data';\n\n    // Data\n    Object.
--------------------
{'code': 'package sodium\n\n// #cgo pkg-config: libsodium\n// #include <stdlib.h>\n// #include <sodium.h>\nimport "C"\n\nfunc RuntimeHasNeon() bool {\n\treturn C.sodium_runtime_has_neon() != 0\n}\n\nfunc RuntimeHasSse2() bool {\n\tr
--------------------
{'code': "\nfunction collectWithWildcard(test) {\n\ttest.expect(4);\n\n\tvar api_server = new Test_ApiServer(function handler(request, callback) {\n\t\tvar url = request.url;\n\n\t\tswitch (url) {\n\t\t\tcase '/accounts?username=cha
--------------------
{'code': '<?php\n\ninterface Container {\n    /**\n     * Checks if a $x exists.\n     *\n     * @param unknown $x\n     *\n     * @return boolean\n     */\n    function contains($x);\n}', 'repo_name': 'guide42/php-immutable', 'path
--------------------
{'code': '/*\n * Read and write JSON.\n *\n * Copyright (c) 2014  Marko Kreen\n *\n * Permission to use, copy, modify, and/or distribute this software for any\n * purpose with or without fee is hereby granted, provided that the abov
--------------------
{'code': 'angular.module(\'appTesting\').service("LoginLocalStorage", function () {\n    "use strict";\n\n    var STORE_NAME = "login";\n\n    var setUser = function setUser(user) {\n        localStorage.setItem(STORE_NAME, JSON.str
--------------------
{'code': '/* Copyright information is at end of file */\n\n#include "xmlrpc_config.h"\n\n#include <stddef.h>\n#include <stdlib.h>\n#include <stdarg.h>\n#include <string.h>\n\n#include "stdargx.h"\n\n#include "xmlrpc-c/base.h"\n#incl
--------------------
{'code': "// flow-typed signature: d37503430b92ad584be6e2c6f8d1fc08\n// flow-typed version: <<STUB>>/ua-parser-js_v1.0.2/flow_v0.171.0\n\n/**\n * This is an autogenerated libdef stub for:\n *\n *   'ua-parser-js'\n *\n * Fill this s
--------------------
{'code': '/* ISC license. */\n\n#include <bearssl.h>\n#include <s6-networking/sbearssl.h>\n\nint sbearssl_skey_to (sbearssl_skey const *l, br_skey *k, char *s)\n{\n  switch (l->type)\n  {\n    case BR_KEYTYPE_RSA :\n      sbearssl_r
--------------------
```

```
import pandas as pd

df = pd.read_parquet("hf://datasets/stojchet/python-codet5p-770m-empty/humaneval/train-00000-of-00001.parquet")
for index, row in df.iterrows():
      for col_name, value in row.items():
          print(f"{col_name}: {value}")
      print("-" * 20)  # #seperating data instances
```

```
    fruit_distribution("0 apples and 1 oranges",3) -> 3 - 0 - 1 = 2
    fruit_distribution("2 apples and 3 oranges", 100) -> 100 - 2 - 3 = 95
    fruit_distribution("100 apples and 1 oranges",120) -> 120 - 100 - 1 = 19
    """

--------------------
task_id: HumanEval/68
language: python
completion: "
    assert type(arr) is list, arr
    assert arr[0] >= 0
    assert arr[-1] >= 0
    arr[0] = arr[0] % 2
    arr[1] = arr[1] % 2
    return arr

pluck([-2,3])

pluck([-3,4,5])

pluck([-2,2,3])
pluck([-2,+2,-1,5])
pluck([1, -3,+1, +2, -3, +2, +3, +4, +5])
pluck(['foo', 'bar', '+1', 'baz'])
pluck(['foo', 'bar', '+5', 'baz', '-1'])
pluck(['fooo', '+', '+'])
pluck(['fooo', '+5', '+2', '+3, '+4, +5, '+6])          # [3,5,5,6]
pluck(['foo', '+7', 'bar,baz'])
pluck(['foo', '+7', 'bar,baz', '+2', '+3, +4, +5, +6, +f])          # [7,4,5,6,5,6]
pluck(['foo', '+5', '-baz,+6,-8,-x,-y,-z,+1,+2,+3,+4,+1,3,-5,-4,+2,-7,-8,-9,+1,+2,+9,1,0,+1,+3,+5,+0,+7,-8,-9,-2,-6,+9,4,-8])
```

```
from datasets import load_dataset

ds = load_dataset("subbu264/codeT5p-19k")
for example in iter(ds["train"]):  # Assuming 'train' is the split name
      print(example)
      print("-" * 20) # Separator between examples
```

```
Streaming output truncated to the last 5000 lines.
{'instruction': "Question: The only difference between easy and hard versions is a number of elements in the array.\n\nYou are given an array $a$ consisting of $n$ integers. The value of the $i$-th element of the array is $a_i$.\r
--------------------
{'instruction': 'Question: Implement the StreamChecker class as follows:\n\nStreamChecker(words): Constructor, init the data structure with the given words.\nquery(letter): returns true if and only if for some k >= 1, the last k\>
--------------------
{'instruction': "Question: Given 2 integers N and K. Convert N(given in Decimal) to base K.\n \nExample 1:\nInput:\nN = 10, K = 2\nOutput:\n1010\nExplanation:\n(10)_{10} = (1010)_{2}\nExample 2:\nInput:\nN = 4, K = 8\nOutput:\n4\r
--------------------
{'instruction': "Question: There is a sequence of colorful stones. The color of each stone is one of red, green, or blue. You are given a string s. The i-th (1-based) character of s represents the color of the i-th stone. If the c
--------------------
{'instruction': "Question: In this Kata, you will be given an array of arrays and your task will be to return the number of unique arrays that can be formed by picking exactly one element from each subarray. \n\nFor example: `sol\
--------------------
{'instruction': 'Question: Polycarpus is a system administrator. There are two servers under his strict guidance — a and b. To stay informed about the servers\' performance, Polycarpus executes commands "ping a" and "ping b". Each
--------------------
{'instruction': 'Question: *This kata is inspired by [Project Euler Problem #387](https://projecteuler.net/problem=387)*\n\n---\n\nA [Harshad number](https://en.wikipedia.org/wiki/Harshad_number) (or Niven number) is a number that
--------------------
{'instruction': "Question: You are given two positive integers A and B. You need to construct two different binary strings (i.e, they are strings which consist of only 0s and 1s), which satisfy these two conditions:\n\nBoth the st
--------------------
{'instruction': 'Question: DZY has a hash table with p buckets, numbered from 0 to p - 1. He wants to insert n numbers, in the order they are given, into the hash table. For the i-th number x_{i}, DZY will put it into the bucket r
--------------------
{'instruction': 'Question: Problem\n\nA graph is given in which N vertices, each numbered from 1 to N, are connected by N-1 undirected edges. For each vertex, output the shortest number of steps to start from that vertex and visit
--------------------
{'instruction': 'Question: # Write this function\n\n![](http://i.imgur.com/mlbRlEm.png)\n\n`for i from 1 to n`, do `i % m` and return the `sum`\n\n    f(n=10, m=5) // returns 20 (1+2+3+4+0 + 1+2+3+4+0)\n\n*You'll need to get a lit
--------------------
{'instruction': "Question: Given an array, return the reversed version of the array (a different kind of reverse though), you reverse portions of the array, you'll be given a length argument which represents the length of each por
```

Overall, the results feel like the data is dealing with key valued data structure and the tagging can be the keys referring to relevant parts of the data.

3.Issues

Currently having issues with prompt engineering but hoping will become easier with current dataset and tagging.

4.Data may not be enough for the tagging even though its relevant also unsure about if code should be included within the tag for now sticking with text.