Moody's 20191006

Please complete by October 18th, 2019. Provide code+output as part of your deliverable. Do not hesitate to reach out if you have questions.

This assignment is meant to be a first dive into the Moody's corpora. The goal is to get you familiar with the data structure at this point (it is a moving animal), and with the challenges having to do with working in Summit and, more importantly, working with OCR output.

We are going to work with one single manual, the one from 1930 having to do with industrial firms. You should have access to the Summit directory `/scratch/summit/diga9728/Moodys/Industrials/OCRrun1930`.[1] There is all kinds of random code/output there, for the purposes of this exercise focus only on the files contained in the directories that are named with three digits (the full list of relevant files is in `listallday1930.csv`).[2]

The structure of the data is as follows. The name of the files is of the form `OCRoutputIndustrial19300039-003213.day`. The first four digits in the file name refer to the year of the manual (1930). The following four digits are the microfiche that corresponds to the file. The following four digits (after the hyphen) are the image within the microfiche (note that the microfiche-image combination refers to a unique page in the manual). The last two digits refer to the brightness used by the OCR engine.

Regarding the content of the `.day` files: they are csv files created from scraping the XML output that the OCR engine creates. Each row in the .day file refers to a word (rather a token). The first six numbers in each row give details as to the location of each word. For now focus on entries 2–5, which give the x_1, x_2, y_1 and y_2 locations of each word in the page. These are measured in "sub-pixels" (I am making that name up): I typically work in pixel space, just multiply the coordinates by 400/1440 to move into pixel space. The other two columns we will work with in this exercise are the last two (columns 15 and 16), which contain the words (with punctuation in 15, without punctuation in 16).

The last important file you need to use is `mastercolumns1930.csv`. This is a master file that has some key metadata on the 1930 manual. Each row is an image (same naming convention, without the brightness data). The first entry is the image name, the second is the "section of the book" it belongs to (loosely named), the last column is the key one: it has 0s for pages we do not care about, 1s for "one column pages," 2s for "two column pages," 3s for "three column pages." For this exercise we only care about pages in the book with 1–3 in the last entry of this master file.[3]

(a) Let's work with one single page first. Let's pick `321/OCRoutputIndustrial19300035-002374.day`. Figure out how to "paint" the image: plot all words as rectangles, color coded regarding their content (all capitals versus capitalized versus all lower case versus numbers). Note that for plotting purposes you will need to "invert" the y coordinates.

(b) Take the set of pages that are labelled as one-column pages in `mastercolumns1930.csv`. The company name for these pages appears in the books as capitalized (all characters as capitals) centered in the image. Write a piece of code that grabs word sequences that "look like" a company name for the OCR renderings at brightness 70.

Hint: for this exercise ordering words from top to bottom is quite useful. Not sure you need to define "lines," but it is something that could help.

(c) Document how your algorithm performs as you change the brightness setting.

[1] In case you want to look at the original images, which is something I often do, they are located in `/scratch/summit/diga9728/Moodys/Industrials/Industrial1930`.

[2] Feel free to look at the .sh files to see how I batch jobs. The rest of the code read at your own risk!

[3] I do not remember now if this manual has "mixed" pages, i.e. transitions from a section that has one-column to a section that has two. I coded these as 12, 21, 13, 31, 23, 32. You can ignore those for now.

Background on this task: (a) “painting the image” is a must for many reasons, just give it a go (I’ll share some R code, call me on it if I forget); (b) this generates a bunch of false positives (i.e. CAPITAL STRUCTURE, LIABILITIES), but there are easy ways to get rid of them (start thinking how!); (c) this is a challenging question, mostly due to the pages that have really bad OCR/the engine goes ballistic (it happens, we have documented double digits bugs in the OCR code, but it is still pretty good!).