Louis
Atallah
06/17/2022

Data Extraction and Audio Translation

Summary

Louis
Atallah
06/17/2022

Part 1: Put on your engineer hat

After reading the publication “State of the Developer Nation”, I have tried to imagine how people with visual impairments could better access to its content. It turns out that people suffering from vision troubles are more likely to struggle to get the information from the text and graphs. I had to think about solution to translate the content by another way rather than by reading.

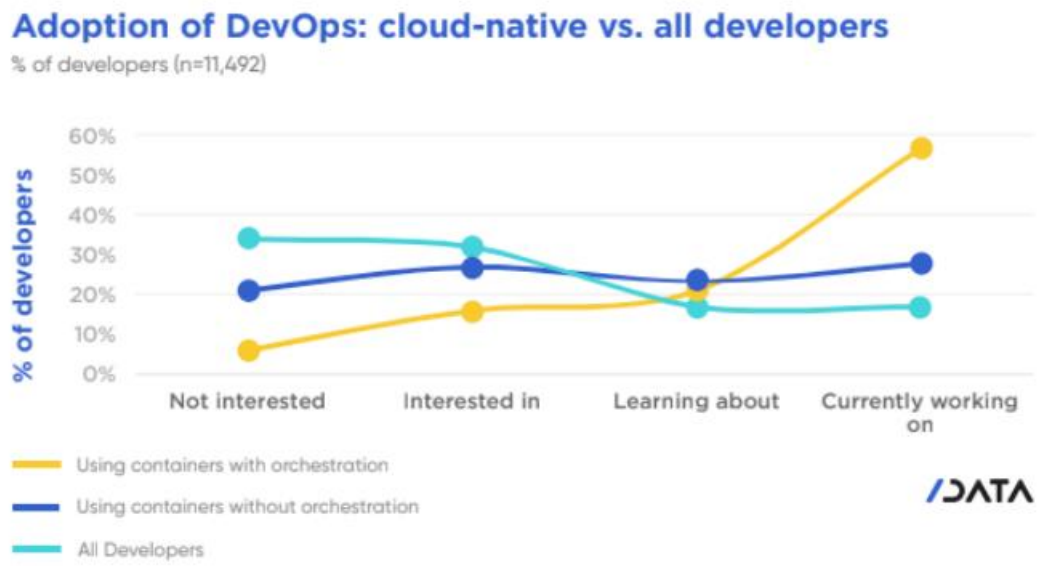
After having a deep reflection and carrying out extensive searches on the challenge, I decided to opt for an audio translation over than other means such as haptic or kinesthetic. To do so, I use Python programing and its external libraries to perform the tasks. As far as the text is concerned, it consists of using the playsound package that takes an mp3 file as input and read it loudly as output. To generate the audio file, I set the language corresponding to the text (here English) and instantiate a gtts object from the gTTS package. I pass the language and the text as parameters. The gtts object enables to create the mp3 file that is saved and read by the playsound function.

```
mytext = pytesseract.image_to_string(Image.open('Capture.png'))
myobj=gTTS(text=mytext,lang=language,slow=True)
```

For the data from images and graphs, I still use Python with the pytesseract library. It is a package that allows to get the data from any image. In the study my file extensions are .PNG opened with the PIL package and more precisely the “Image” sub package. The text in an image can be pulled with the *image_to_string()* function from the pytesseract, while the other content types are pulled with *image_to_data().* This function gathers all the data from the image including the text. Sometimes with some graphs, the data extracted could be null because of the format or the quality of the image. To have a better accuracy, I make the image grayscale using cv2. It highlights elements of the images that could not be detected previously. The data is stored in a data frame and can be accessed and converted to a string type to generate the mp3 files that will be played loudly.

Louis
Atallah
06/17/2022

Part 2: Showoff your ideas



For the second part of the study, I implement the process described in the first part. I use the *image_to_data()* function from pytesseract to get the data and store it in a data frame.

	level	page_num	block_num	par_num	line_num	word_num	left	top	width	height	conf	text
0	1	1	0	0	0	0	0	0	691	370	-1.000000	NaN
1	2	1	1	0	0	0	17	15	547	40	-1.000000	NaN
2	3	1	1	1	0	0	17	15	547	40	-1.000000	NaN
3	4	1	1	1	1	0	17	15	547	19	-1.000000	NaN
4	5	1	1	1	1	1	17	15	97	19	95.972511	Adoption
..	...	...	...	...	...	...	...	...	...	...	...	...
66	3	1	8	1	0	0	17	346	119	11	-1.000000	NaN
67	4	1	8	1	1	0	17	346	119	11	-1.000000	NaN
68	5	1	8	1	1	1	17	348	31	4	43.812553	—
69	5	1	8	1	1	2	59	346	11	8	51.089382	All
70	5	1	8	1	1	3	75	346	61	11	51.089382	Developers

[71 rows x 12 columns]

There are 12 features for each element including their positions and the confidence. To limit and reduce the noise in the table, I perform some modifications on the dataset such as dropping NaN, null values and setting a threshold at 40 in order to have reliable contents. I selected a threshold of 40 after observing that the elements are not correct when the confidence is inferior to 40.

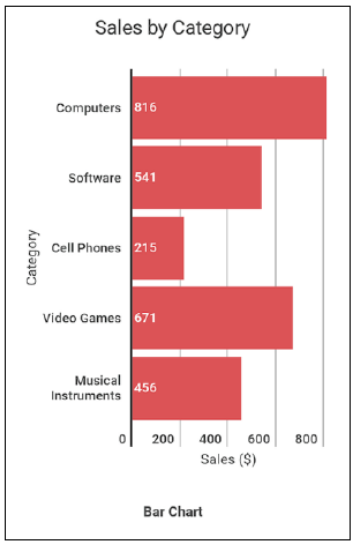
Louis
Atallah
06/17/2022

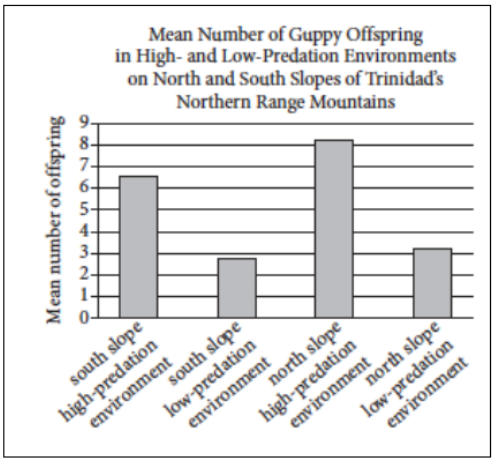
```
data = pytesseract.image_to_data(Image.open('Capture.png'), output_type='data.frame')
data = data.dropna()                # we drop the NaN
data = data[data['text'] != " "]    # we don't select elements containing only a blank
data = data[data['conf'] > 40]      # threshold of 40 for the confidence
```

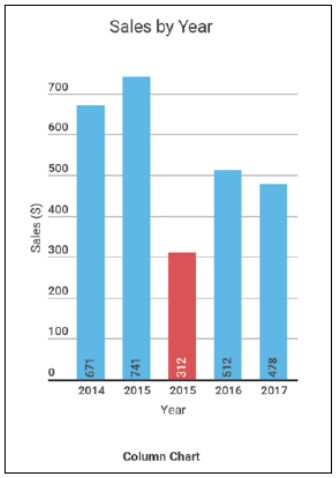
I obtain a clean dataset as stated below:

	level	page_num	block_num	par_num	line_num	word_num	left	top	width	height	conf	text
4	5	1	1	1	1	1	17	15	97	19	95.972511	Adoption
5	5	1	1	1	1	2	123	15	20	16	93.302750	of
6	5	1	1	1	1	3	151	16	87	18	92.086472	DevOps:
7	5	1	1	1	1	4	246	15	130	16	92.270035	cloud-native
8	5	1	1	1	1	5	383	19	27	12	96.948494	vs.
9	5	1	1	1	1	6	418	15	23	16	96.760635	all
10	5	1	1	1	1	7	449	15	115	19	96.046234	developers
12	5	1	1	1	2	1	17	42	8	10	91.165405	%
13	5	1	1	1	2	2	30	43	11	9	96.472588	of
14	5	1	1	1	2	3	46	42	68	13	93.106949	developers
15	5	1	1	1	2	4	118	42	57	11	90.687180	(n=11,492)
23	5	1	2	1	3	1	17	150	13	14	92.019943	2
31	5	1	2	1	7	1	17	233	13	13	52.129387	z
35	5	1	3	1	1	1	116	253	23	10	96.977623	Not
36	5	1	3	1	1	2	145	252	67	11	96.676399	interested
38	5	1	3	1	1	4	336	252	10	10	96.557564	in
39	5	1	3	1	1	5	393	252	58	13	94.563560	Learning
40	5	1	3	1	1	6	457	252	39	11	74.421730	about
42	5	1	3	1	1	8	590	252	54	13	80.068863	working
48	5	1	4	1	1	1	92	294	118	8	93.858772	containers
49	5	1	4	1	1	2	153	283	22	28	96.911858	with
50	5	1	4	1	1	3	224	295	27	7	95.530678	orchestration

Part 3: Expand your approach to other graphical materials







To process these 3 graphs autonomously, I implement a for loop (one iteration for each graph). I first apply the code used in the part 2 to extract the information. If the data frame is empty after getting rid of the null and NaN values, I perform a grayscale. In fact, only the graph 2 returned an empty data frame.

```
if data.empty:
    img = Image.open('Capture'+str(i)+'.png').convert('L')
    ret,img = cv2.threshold(np.array(img), 125, 255, cv2.THRESH_BINARY)
    img = Image.fromarray(img.astype(np.uint8))
```

Once the modification done, the data frame contains elements and their positions. It is not empty anymore and is ready to be processed and be played loudly as coded below:

```
# graph described via audio mode
# assembling the text
mytext = "Graph "+ str(i)
for j in range(data.shape[0]) :
    mytext+= str(data.iloc[j,11])+" "

myobj=gTTS(text=mytext,lang=language,slow=True)
myobj.save('Graph'+str(i)+'.mp3')
play('Graph'+str(i)+'.mp3')
print(data)
```

While the text is being read, the graph is displayed for convenience reasons.

Conclusion

To conclude this study, adapting a document to a blind person is a challenge as well as for translating the information as for finding the best way to transmit it. I proposed a solution based on the audio, detailed as well as I could do in the time allowed. Although there is not all the information contained in the graphs such as points coordinates, the context the axis the legend and units are detected. It is not trivial to get the data from a data point, but I have some ideas to explore related to the computer vision, such as measuring the location of the points related to the axis.