Andrei Mihnea-Stefan

Wikipedia parser pentru imagini de persoane

Introducere

Acest parser downloadeaza imagini de persoane de pe wikipedia. Codurile sunt numite *download_wiki_images_final.ipynb* si *download_wiki_images_final.py*. Aceste imagini downloadate de pe Wikipedia sunt introduse in MTCNN, care extrage numai fata din imagine. Dupa aceea, aceste fete sunt introduce in resnet, care invata numele fetei persoanei. Este necesar sa se treaca print MTCNN deoarece resnet asteapta numai fete si pozele trebuie sa fie toate de aceeasi dimensiune: 16- de pixeli pe 160 de pixeli.

Ideile din acest parser de Wikipedia sunt de fapt simple, o data ce stii anumite lucruri despre paginile Wikipedia:

- Wikipedia tine si un dictionar cu toate cuvintele si numele (de personae, locatii geografice, cladiri istorice, etc.) care apar in toate paginile lor. Acest dictionar este accesat la linkuri de tipul urmator:

```
prefix="Q"+str(count_links)
flag=1
wiki_link="https://www.wikidata.org/wiki/%s"%(prefix)
```

Aici, *%s* din link este ceva de forma Q23 sau Q1234 sau Q540000 sau etc.

- O data accesat acest link exista o modalitate de a vedea jsonul paginii. Din acest json este o modalitate de a verifica daca pagina este despre o persoana.
- Daca pagina este despre o persoana, se poate extrage numele persoanei.
- De asemenea, se poate extrage numele imaginii care apare pe pagina Wikipedia.
- Linkuri-le in care imaginile sunt tinute au o farmatare standard:

  https://www.wikidata.org/wiki/%s#/media/File:%s

  Aici primul *%s* este in formatul explicat mai devreme (Q urmat de un numar: Q23, etc), iar al doiles *%s* este numele imaginii obtinut la bullet pointul anterior, in care spatiile sunt inlocuite cu "_". Aceste formatari standard ale likurilor este descoperit destul de repede daca te joci oleaca apasand pur si simplu pe linkurile din Wikipedia.

Cod

Cu aceste informatii, codul devine simplu. Pur si simplu creez un loop infinit care incearca tot felul de prefixe de forma Q1,Q2,….,Q1000,….., accesez pagina din dictionarul Wikipedia si ma uit la json (pentru acesta devine un dictionary in python cu care se poate lucre extrem de repede):

```
while True:
    prefix="Q"+str(count_links)
    flag=1
    wiki_link="https://www.wikidata.org/wiki/%s"%(prefix)
    json_link="https://www.wikidata.org/wiki/Special:EntityData/%s.json"%(prefix)
    try:
        response=requests.get(json_link)
    except requests.exceptions.Timeout:
        print("Too much time to request wikipedia link %s"%(image_link))
        flag=0
    except requests.exceptions.TooManyRedirects:
        print("Bad wikipedia link %s"%(image_link))
        flag=0
    except requests.exceptions.RequestException as e:
        print("Catastrophic error for wikipaedia link %s"%(image_link))
        flag=0
    if flag==1:
        try:
            person=json.loads(response.content)
        except ValueError as err:
            flag=0
```

Restul de *try* si *except* sunt acolo pentru ca incerc sa prind tot felul de erori. Altfel codul se opreste.

Dupa aceea verific daca pagina este despre o persoana:

```
if flag==1:
    if prefix in person["entities"].keys():
        if "P31" in person["entities"][prefix]["claims"].keys():
            if "mainsnak" in person["entities"][prefix]["claims"]["P31"][0].keys():
                if "datavalue" in person["entities"][prefix]["claims"]["P31"][0]["mainsnak"].keys():
                    if "value" in person["entities"][prefix]["claims"]["P31"][0]["mainsnak"]["datavalue"].keys():
                        if "id" in person["entities"][prefix]["claims"]["P31"][0]["mainsnak"]["datavalue"]["value"].keys():
                            if person["entities"][prefix]["claims"]["P31"][0]["mainsnak"]["datavalue"]["value"]["id"]=="Q5":
```

Daca este despre o persoana, obtin numele persoanei si il curat de anumite caractere:

```
person_name=""
if "entities" in person.keys():
    if prefix in person["entities"].keys():
        if "labels" in person["entities"][prefix].keys():
            if "en" in person["entities"][prefix]["labels"].keys():
                person_name=person["entities"][prefix]["labels"]["en"]["value"]
                #remove quotes since it gives an error when the image file name when saved
                if '"' in person_name:
                    person_name=person_name.replace('"',"")
                if "/" in person_name:
                    person_name=person_name.replace('/',"")
                if " "in person_name:
                    person_name=person_name.replace(' ',"")
                if "\\" in person_name:
                    person_name=person_name.replace(' ',"")
                if "." in person_name:
                    person_name=person_name.replace('.',"")
                if "?" in person_name:
                    person_name=person_name.replace("?","")
                if "!" in person_name:
                    person_name=person_name.replace("!","")
                #remove accents, diacritics
                person_name=unidecode.unidecode(person_name)
```

Dupa aceea obtin numele imaginii si accesez linkul cu formatarea mentionata in introducere pentru a downloada imaginea:

```
if person_name!="":
    if "claims" in person["entities"][prefix].keys():
        if "P18" in person["entities"][prefix]["claims"].keys():
            if "mainsnak" in person["entities"][prefix]["claims"]["P18"][0].keys():
                if "datavalue" in person["entities"][prefix]["claims"]["P18"][0]["mainsnak"].keys():
                    if "value" in person["entities"][prefix]["claims"]["P18"][0]["mainsnak"]["datavalue"].keys():
                        image_name=person["entities"][prefix]["claims"]["P18"][0]["mainsnak"]["datavalue"]["value"]
                        if '"' in image_name:
                            image_name=image_name.replace('"','')
                        if "\\" in image_name:
                            image_name=image_name.replace("\\","")
                        image_suffix="_".join(image_name.split(" "))
                        image_link="https://www.wikidata.org/wiki/%s#/media/File:%s"%(prefix,image_suffix)
                        #image_link = urllib.parse.quote(link,safe=':/')
                        try:
                            r=requests.get(image_link)
                        except requests.exceptions.Timeout:
                            print("Too much time to request image link %s"%(image_link))
                            flag=0
                        except requests.exceptions.TooManyRedirects:
                            print("Bad image link %s"%(image_link))
                            flag=0
                        except requests.exceptions.RequestException as e:
                            print("Catastrophic error for image link %s"%(image_link))
                            flag=0
                        if flag==1:
                            #parse HTML code
                            soup=BeautifulSoup(r.text,"html.parser")
                            #find all images in url (should only be 1 here)
                            images=soup.findAll('img')
                            if len(images)!=0:
                                image=images[0]
                                try:
                                    image_link=image["src"]
                                except:
                                    flag=0
                                if flag==1:
                                    image_link="https:"+image_link
                                    try:
                                        urllib.request.urlretrieve(image_link,save_dir+"\\"+person_name+".jpg")
                                    except Exception as e:
                                        print("%s for image has error %s, link:%s"%(image_name,e,image_link))
                                        flag=0
                                    if flag==1:
                                        #print("Success for %s with link %s"%(image_name,image_link))
                                        count_people+=1
```

Restul de *try* si *except* sunt acolo doar ca sa prind anuite erori pentru ca altfel codul ar iesi din while loop.

Ca sa iti dai seama ce chei din dictionary sa accesezi pentru a obtine numele imaginii, sau a persoanei, pur si simplu printezi jsonul in python si cu ctrl+F cauti in el unde este numele persoanei si prin ce chei trebuie sa teci ca sa ajungi acolo.