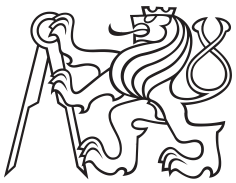
Bachelor Project



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Wikipeople

Milan Welser

Supervisor: Ing. Vojtěch Franc, Ph.D.
Field of study: Open Informatics
Subfield: Computer and Information Science
August 2021

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Welser Milan** Personal ID number: **474656**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer and Information Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Wikipeople

Bachelor's thesis title in Czech:

Anotovaná databáze obrázků tváří lidí z Wikipedie

Guidelines:

The goal of the project is to develop scripts for automated downloading and processing of personal homepages from Wikipedia. The processing script will find a representative face and attributes associated to the person. The output will be a database of facial images annotated by attributes like the subject's name, birth date, age, gender, occupation and so on. Part of the project will be a statistical evaluation of the precision of the automatically created annotation.

Bibliography / sources:

[1] R.Rothe, R.Timofte and L. Van Gool. DEX: Deep Expectation of apparent age from a singe image. IEEE ICCV Workshops 2015
[2] V.Franc, J.Čech. Learning CNN from weakly annotated facial images. Image and Vision Computing. 2018.
[3] S.Moschouglou et al. AgeDB: the first manually collected, in-the-wild age database. IEEE CVPR Workshops 2017.

Name and workplace of bachelor's thesis supervisor:

Ing. Vojtěch Franc, Ph.D., Machine Learning, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **25.02.2021** Deadline for bachelor thesis submission: **13.08.2021**

Assignment valid until: **30.09.2022**

Ing. Vojtěch Franc, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____ Date of assignment receipt

_____ Student's signature

Acknowledgements

I would like to thank Ing. Vojtěch Franc, Ph.D. for supervising my thesis and for providing lots of important feedback and suggestions. I would also like to thank my friend Matěj Suchánek, who advised me on use of Pywikibot.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 13 August 2021

Abstract

In this thesis we present the method and scripts intended for automatic downloading and processing of personal homepages from Wikipedia. With the fields of machine learning and computer vision constantly developing, the demand for annotated databases of facial images rises. The goal of this work is the development of tools for creating such databases. The image annotation is a set of additional information describing specific attributes in said image. The reader is introduced to principals and function of Wikipedia, Wikidata and other relevant projects together with their mutual relationship. The thesis explains used method and implementation details of created scripts. Last but not least, the scripts are tested on a set of pages, which have been manually annotated in order to determine the result precision.

Keywords: Annotated database of facial images, dataset creation, dataset, Wikipedia

Supervisor: Ing. Vojtěch Franc, Ph.D.

Abstrakt

V této práci představujeme metodu a skripty určené k automatickému stahování a zpracování osobních stránek z Wikipedie. S neustále se rozvíjející oblastí strojového učení a počítačového vidění stoupá zájem o anotované databáze obrázků tváří. Účelem této práce je vývoj nástrojů pro tvorbu přesně takové databáze. Anotace obrázků spočívá v dodatečných informacích k obrázku, které popisují jeho určité atributy. Čtenáři jsou vysvětleny principy fungování Wikipedie, Wikidata a dalších souvisejících projektů spolu s jejich vzájemnými vztahy. Práce vysvětluje použitou metodu i implementační detaily vytvořených skriptů. V neposlední řádě jsou skripty testovány na množině stránek, která byla rovněž manuálně anotována za účelem zjišení přesnosti výsledků.

Klíčová slova: Anotovaná databáze obrázků tváří, tvorba datových sad, datová sada, Wikipedia

Překlad názvu: Anotovaná databáze obrázků tváří lidí z Wikipedie

Contents

Figures

Tables

Chapter 1

Introduction

The goal of this project is to develop scripts for downloading and processing personal homepages from Wikipedia, specifically the English version, serving as tools for creating a database of facial images annotated by attributes such as name, birth date, age, gender, occupation and possibly even the date of death.

Wikipedia is the largest multilingual online encyclopedia and its contents are freely available in over 300 languages. The English Wikipedia consists of more than 6.3 million articles, including hundreds of thousands pages about specific people (personal homepages). [6]

With rapid progression of machine learning, the demand for collections of data (called datasets) is increasing. One of the most sought after datasets in the field of computer vision are those with annotated facial images. An annotated facial image is accompanied by a set of labels, such as age, gender, occupation or other useful traits of said facial image.

This work presents the tools intended to convert Wikipedia's personal homepages into annotated facial images for use in machine learning.

Chapter 2

Wikipedia and related projects

2.1 Wikipedia

The word "Wikipedia" is a blend of words "wiki" and "encyclopedia". In contrast to the latter word, probably known to most people, the former is more interesting. The word "wiki" is a Hawaiian word meaning "quick", but nowadays it is known much more for being a type of a website. A website, whose contents can be edited from the web browser (often by any visitor), and which usually keeps a version history for each editable page. [7]

Anyone can edit or create articles on Wikipedia, as long as they abide by the rules [8]. This has many effects, some positive, such as increasing the site growth rate or overall popularity increase, but also a few negatives. Most obvious being vandalism, meaning users could edit any page to anything they wanted, f.e. fill it with false information or insults. However, the Wikipedia community is large and it does not take long for the page to be reverted to its previous state thanks to aforementioned version history. The more severe negative effect is brought by users who have good intentions. In their effort to better Wikipedia, they introduce minor inaccuracies or even falsities, which are often much harder to detect. Another common problem consists of users placing information in the wrong section of the page. This would not have to cause any problems for a human user, but quite many for a program extracting information from said page.

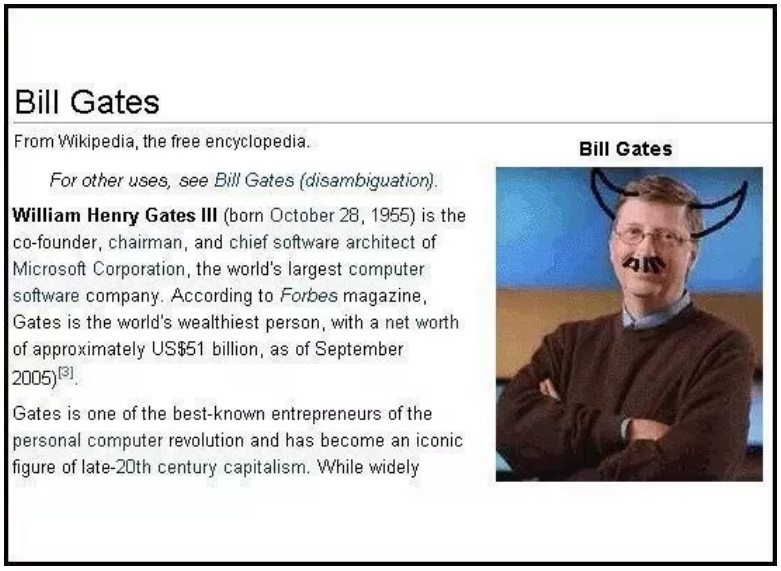


Figure 2.1: Example of vandalism on Wikipedia in 2005 [1]

2.2 Wikimedia

Wikimedia movement (or just Wikimedia) was created following Wikipedia's success around its community. The movement revolves around a group of many related projects [9], including but not limited to:

- Wikipedia, The free encyclopedia
- Wiktionary, Free dictionary and thesaurus
- Wikimedia Commons, Free media repository
- Wikidata, Free knowledge base
- Wikiquote, Collection of quotations
- MediaWiki, freely available wiki engine, used by Wikipedia and other Wikimedia projects

2.3 Wikidata

"Wikidata is a free, collaborative, multilingual, secondary database, collecting structured data to provide support for Wikipedia, Wikimedia Commons, the

other wikis of the Wikimedia movement, and to anyone in the world." [10]

Wikidata, another Wikimedia project, works on the same principle as Wikipedia and is very important for this project. In contrast to Wikipedia, Wikidata stores *structured data* instead of any data, making them easily readable for both humans and machines. Many Wikipedia articles have their Wikidata page equivalent. It should be noted that Wikidata only serves as a secondary database that provides support for Wikipedia and other projects. This means information between Wikipedia and Wikidata could differ. Information present on Wikipedia could be completely absent from Wikidata or vice versa. This could be considered beneficial as we can utilize Wikidata as another information source.

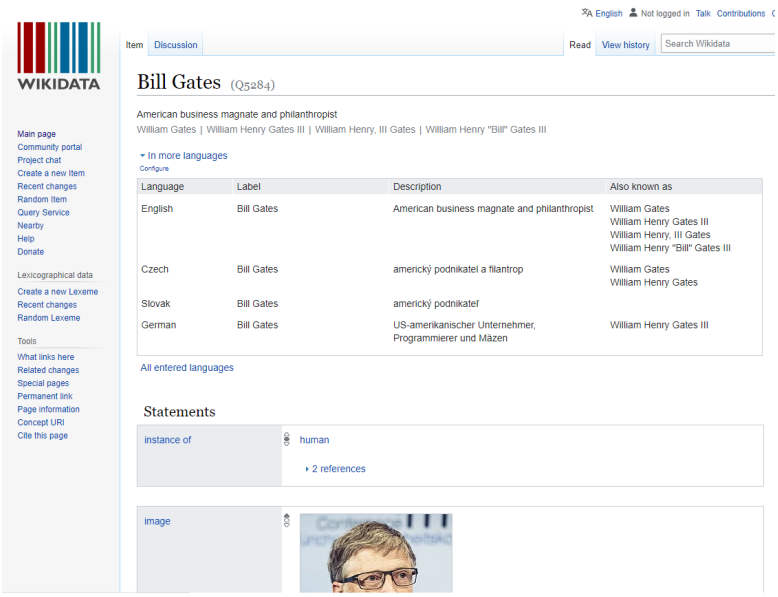


Figure 2.2: Example of a personal page on wikidata (Bill Gates) [2]

2.4 Wikimedia Commons

Wikimedia Commons (or just Commons) is a free media file repository. Available media files include sound, images or video clips. This repository functions on the same principles as Wikipedia and other Wikimedia projects. Wikimedia Commons servers as a common repository for many Wikimedia projects, including Wikipedia and Wikidata. Not all media files from Wikipedia are hosted here, as Wikipedia predates Wikimedia Commons and not all editors choose it as a repository, but it's still very significant. Especially because along with the media files, the site sometimes stores extra data describing said file, which could prove useful for the cause of this project.

Chapter 3

Retrieving data from Wikipedia

3.1 Purpose and readability

Wikipedia was created for human users and their need for information. Even today, this is the site's main purpose:

"Wikipedia's purpose is to benefit readers by acting as a widely accessible and free encyclopedia; a comprehensive written compendium that contains information on all branches of knowledge. The goal of a Wikipedia article is to present a neutrally written summary of existing mainstream knowledge in a fair and accurate manner with a straightforward, "just-the-facts style". Articles should have an encyclopedic style with a formal tone instead of essay-like, argumentative, promotional or opinionated writing." [11]

In order to accomplish it's goal, Wikipedia presents the information in a format, which is most easily comprehended by a human reader. This poses a problem for any machines intending to extract information from Wikipedia, since human readable format may not be easily processed by machines and vice versa.

The MediaWiki software, on which Wikipedia is based, still offers ways to access their sites programmatically. And although many articles on Wikipedia contain at least some partially structured data, any programmer attempting to extract data from articles programmatically will inevitably find themselves

trying to parse through significant amounts of unstructured data. Some third party services attempt to structure and provide data from Wikipedia with varying degrees of success and coverage, however these services do not have the interconnection with other Wikimedia projects such as Wikidata and Commons, f.e. the feature of accessing the page equivalent at Wikidata and the ability to access additional data for images on Commons.

3.2 Web scraping

Web scraping would be considered the brute force method of solving the problem at hand. Web scraping Wikipedia and other MediaWiki wikis is often advised against for multiple reasons. Not only does Wikipedia already offer other, easier methods of downloading an article, the programmer intending to webscrape Wikipedia is going around set limits for number of requests and could affect Wikipedia servers performance. Site administrators could punish such behaviour with slowing or blocking future connections and in most extreme cases, an IP address ban could be issued.

3.3 Wikipedia API

An API, or application programming interface, is one of the most common ways for programs to interact with websites. Thanks to the MediaWiki software, the Wikipedia offers an API in accordance with the MediaWiki specification. [12] The API offers not only options for reading Wikipedia, but for editing it as well.

Using the API instead of web scraping comes with many benefits. The most obvious is the absence of parsing through HTML. The API also allows users to specify the output format and query the wiki for pages from specific wiki categories or pages that meet pre-set requirements. The most important benefit for the site is the ability to regulate the number of answered requests, virtually staying in control and making the site run smoothly. Anyone using the API should abide by the API etiquette.[13]

The API of course has it's limitations and negatives. The most apparent negative is the orthodox, newcomer-unfriendly documentation. Any users without previous experience using APIs could have a very hard time using it

to it's full potential. The wiki does not provide many examples of API usage either.

Most limitations are put in place in order to keep the site running smoothly, specifically the maxlag parameter, which could be understood as an aggressivity rating (lower being less aggressive). When Wikipedia servers get busy, the maxlag threshold is raised and all tasks with lower maxlag are stopped, until the servers are able to process them. [14] Abusing this parameter to get faster responses is against guidelines and may result in a ban. The number of pages per request are also limited to 50 and the request contains information on how to request additional pages, allowing for recursive requests should anyone need higher volumes of pages. The reasoning for such limitations are clear - Wikipedia's primary interest lies within the human users and their work should not be delayed by any programs.

3.4 Pywikibot

"Pywikibot is a Python library and collection of scripts that automate work on MediaWiki sites. Originally designed for Wikipedia, it is now used throughout the Wikimedia Foundation's projects and on many other wikis." [15]

Pywikibot is the next level after the API to work with Wikipedia or any wiki running on MediaWiki software. It is built on the MediaWiki API, but it is friendlier towards newcomers. With Python being the prevalent language in the Machine learning field, Pywikibot becomes a very attractive choice for traversing wikis.

Though Pywikibot is designed towards the part of community that edits and contributes to wikis, it's features for reading Wikipedia are extensive and cover almost all of the API functionality.

Pywikibot consists of two main components. Pywikibot itself and already written an functioning scripts. The list of scripts is decent in size, though mostly focused on editing and contributing to wikis. [16] The Pywikibot part consists of all the source files to interact with wiki websites. It is used for the scripts shipped with the package as well as for use in custom scripts. Although it may appear so, the user isn't required to have an account to read and traverse a wiki with Pywikibot.

As the Pywikibot is built on the MediaWiki API, the maxlag parameter limitations apply here as well. The package default is maxlag=5 and it is recommended to keep it that way. The package contains mechanisms that delay other requests to the wiki if a request was declined. The only negatives of this package that became apparent during this project were some minor missing features from the API and the lack of examples for using pywikibot as a library. MediaWiki provides just barely enough examples to get a user going, but for slightly intermediate tasks the user must go to the documentation, which once again, is very orthodox and newcomer unfriendly. A possible better option is to simply look at the source code of the package and learn about functions from their docstrings. Combining this approach with listing all member functions in python and trial and error eventually yields result, albeit a very inconvenient way of learning to operate the package.

Still, Pywikibot qualities overweight the negatives and thus it seems to be the ideal tool for the project.

3.5 Downloading the entire Wikipedia

An option which was explored during this project was downloading the entirety of Wikipedia and process it offline. This seemed like a very promising solution as the user would be able to create snapshots of Wikipedia and have different versions of the database. Unfortunately, working with offline materials without pywikibot, easy access to wikidata and commons, together with low amount of resources on the topic, proved to be too complicated and the solution with pywikibot was chosen instead.

Chapter 4

The method

The purpose of this chapter is to explain the used method for achieving the goals of this project without going into implementation details. The project consists of two parts (scripts), the first one collects specific data from Wikipedia, Wikidata and Commons, and the second one processes the collected data, creating the final database.

4.1 Data collection

The sole purpose of the data collection script is to harvest necessary data from Wikipedia, Wikidata and Commons and save them in a reasonable format.

In order to achieve this, we need to traverse Wikipedia. For this, we use Wikipedia's categories. The main category for this is the Living people category, which contains over million pages.[17] The secondary categories are the deaths categories. They are split based on the year of subject's death. For example, Stephen Hawking's page belongs to a "2018 deaths" category. The number of pages in these categories can be seen in fig.4.1.

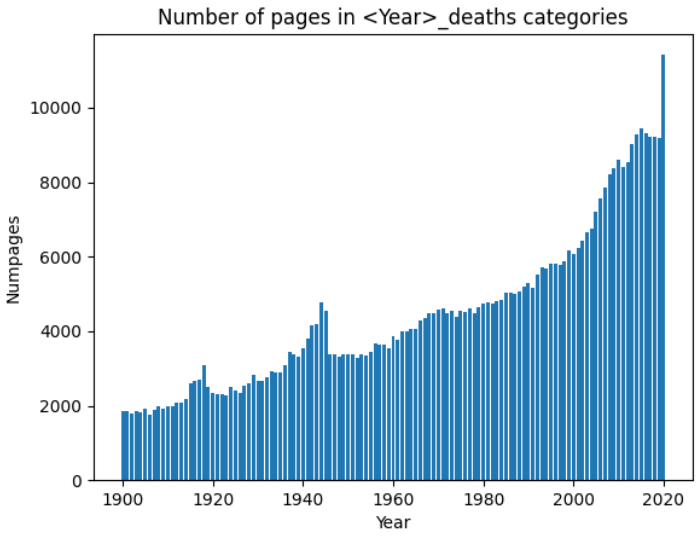


Figure 4.1: Page counts in respective death categories

Furthermore, we attempt to traverse the categories in a sort of uniform manner, meaning the same number of pages are processed for every letter. For example, we would like to process 10 pages for every latter, totalling 270 pages. 10 starting with special symbols, 10 starting with A and so on.

We require every entry in the final database contains the following:

- Name
- Facial image
- Birth date
- Age on the facial image
- Sex/Gender
- Occupation
- Death date (only for deceased subjects)

Some of the required fields could be found within the page's infobox. An infobox (example in fig.4.2 is a box filled with information about the subject, usually located on the right side of the page.

Figure 4.2: Example of an infobox (František Křižík) [3]

4.1.1 Name

Retrieving the subject's name is trivial in the vast majority of cases, as it is located in the URL or name of the page. In fig.4.3 we can see a rare case of a page with a name that isn't according to the requirements. The title of the page should either contain only the subject's name, or the page should not be categorised as "2020 deaths".

Figure 4.3: Page with an incorrect title [4]

4.1.2 Facial image

The facial image is the single most important trait which could not be omitted. The main image of the article, also called thumbnail or page image, is located within the infobox. It would be pointless to collect data on pages with no image, therefore we could skip them in this part of the process. Based on testing and experience during the project, the estimated percentage of pages with a page image was around 20 to 30 % in the Deaths categories and around 30-40 % in the Living people category, meaning 70-80 % and 60-70 % of pages were skipped from the Deaths and Living people categories respectively based on missing imagery. More precise numbers can be seen in fig.4.4.

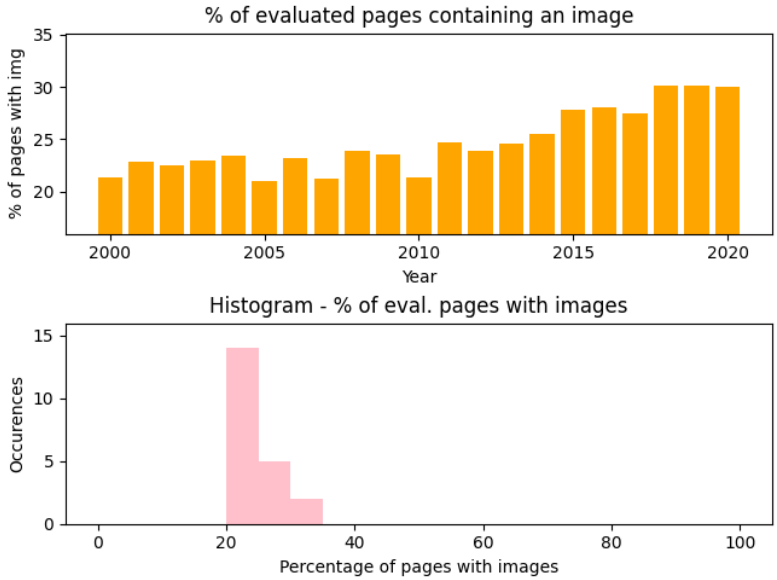


Figure 4.4: Percentages of pages with page image in Deaths categories

4.1.3 Birth date

The birth date (specifically birth year) is required in order to determine the subject's age in the photograph. On Wikipedia pages, the birth date is most commonly encountered again within the infobox. We also grab the birth date from the Wikidata page equivalent. In future, we could also try to parse out the birth date from the first paragraph of text on Wikipedia, however it is doubtful that the increase in precision would be significant without above average parsing methods.

4.1.4 Age on the facial image

One of the most interesting attributes of a facial image is the subject's age. Images annotated with age are already often sought after and the demand can be expected to increase over time because of potential uses in machine learning.

In order to obtain the age on the facial image, we need two attributes. Already obtained birth date and the date said photograph was taken. After subtracting former from the latter, we acquire the age on the image.

The most common place to find the image date is from the summary located below the image on it's repository site (usually either Wikipedia or Commons), as seen in fig.4.5. We collect the contents of rows "Description" and "Date". We also check the image metadata (EXIF), specifically the tag number 36867 - DateTimeOriginal, which contains the date and time when the original image data was generated.

Figure 4.5: The image summary containing the image date (Donald Trump)[5]

4.1.5 Sex/gender

It appears that the only reliable source in providing subject's sex is Wikidata. This field is very often missing in the Wikipedia infobox. When manually examining articles, the only apparent way of determining it was by used pronouns within the text or perhaps guessing from the photograph (which proved very difficult in many cases). But not even this was always possible and it required a level of understanding of the written text, therefore the sole source of subject's sex/gender is Wikidata.

4.1.6 Occupation

Although occupation may be one of the lower priority attributes, we still require it. The best place for retrieving information about occupation is once again Wikidata. It contains already structured lists of occupations in vast majority of cases. We also retrieve the contents of the occupation field within the infobox. However, this field is not nearly as common as could be expected. It should also be noted that sometimes the occupation is mentioned in the first sentence of text within the Wikipedia page, which we also collect.

4.1.7 Death date

We collect the death date for deceased subjects. It can be retrieved the same way as birth date, which is the way it's done in the project. There is also an alternate way to retrieve the death date. Because of the aforementioned way pages are categorised on Wikipedia (specific categories for deaths every year), we could retrieve it from the name of the category. Although this is only an observation, during the development of this project, after thousands of processed pages, there were zero observed inconsistencies in the death date and the page's category.

4.2 Data processing

The purpose of this part of the project (the second, processing script) is firstly to evaluate the collected data, and secondly to find faces within the collected images and determine whether they meet the requirements. The decision making in this part is very simple and could be improved upon in the future, but as we will see later, even such simple decision making can bring somewhat decent results. During this part, there are many places where an entry could be discarded, therefore roughly 40-50% of collected entries were discarded in this process. To put this in perspective, the percentages of collected pages relative to the all existing pages were around 20-30% in deaths categories and 30-40% in the living people category. With another 40-50% being discarded in the process, the final percentage of used pages is around 10-15% for deaths categories and 15-25% in the living people category. The death categories had consistently higher discard rates, which could probably be explained with the pages being older and not having as much relevant information available in contrast to the living people category, however, this is only a hypothesis.

Description	Deaths	Living people
After download	20-30%	30-40%
After processing	10-15%	15-25%

Table 4.1: Percentage of used pages

4.2.1 Facial recognition

After using facial recognition software (software which detects faces and facial features) on collected images, we get lots of useful data. For each image, we receive the number of found faces, their locations in the image, the score, showing how confident the software is that it's a face, and possibly a list of locations of facial landmarks (both eyes, nose and mouth corners). We eliminate all images that contain both less and more faces than 1 in order to ensure that the photograph features the subject. This could be improved upon in the future by using facial recognition on multiple images from the page and comparing the identity of found subjects. However, an argument could be made that implementing this would improve results only very slightly (if at all) due to the low percentage of pages with images along with the assumption that pages with multiple images are likely to contain better quality page images.

4.2.2 Sex/gender

Because we only collect sex/gender from one source, we assume it is correct and use it.

4.2.3 Birth date and death date

For both birth date and death date, we collect them from 2 separate sources, therefore we evaluate them in the same manner. If we have data from both sources, we compare them and if the years equal, we considers the correct year. If they differ, we discard this entry. If we only have data from one source, we consider this source trustworthy and assume the year is correct. In the case of no data, the entry is discarded.

4.2.4 Image date and age

We collect the possible image date from three places - two of those are from the image file page summary (description and date), and the third is from the image exif. The evaluation logic is somewhat simple. If all three years equal, we assume they are correct. If the "Date" field from summary and exif date years are equal, we assume they are correct, ignoring the "Description" field. Lastly, the following priority takes place if priory comparisons did not yield results. Highest priority field is the "Date" field from the summary, second priority is the exif date and the lowest priority is the description box. If no result is decided upon, the entry is discarded.

After successfully finding the image date, the age is calculated and a range check is performed to discover any inaccuracies. If the range check fails, the entry is discarded.

4.2.5 Occupation

Although we collect occupation data from 3 sources - Wikidata, Wikipedia infobox and the first sentence on the Wikipedia page - in the end we only use the occupation provided from Wikidata. Wikidata provides occupation for a vast majority of pages we look at. This data appeared very reliable even during the development of the project. On the other hand, the occupation field in infoboxes was missing in over 50% of the pages and parsing any useful information from the first sentence proved to not be worth the effort. To add to this, comparing occupation from multiple sources that do not use the same naming styles can also be difficult. Even though the words "singer", "rapper" and "musician" go together, computers aren't able to comprehend the association without more advanced parsing solutions. Therefore the choice was made to only use the occupation data from Wikidata. We still keep the collected data in case of an improvement in parsing in the future.

Chapter 5

Implementation

This chapter describes the implementation details, such as used platform, software and libraries. It also describes files and the overall file structure. Lastly, it addresses the important parts of code itself.

5.1 Platform and language

This project was developed using OS Windows 10 and Python 3.7.9. There shouldn't have been any OS specific libraries (besides os.path to separate filename and fileextension), commands or situations encountered, therefore the scripts could likely work on other operating systems, however, this could not be guaranteed. As for the Python version, all the used libraries *should* work with Python 3.6 and higher, still, no guarantees.

5.2 Used libraries

Here is the list of libraries with short description of what they were used for. The list does not contain dependencies as Python should install those automatically.

- pywikibot - the most important library, it allowed access to wikipedia, wikidata and commons programmatically
- mwparserfromhell - important parsing library used for parsing wikicode in order to extract information from wikipedia
- numpy - for reading and saving images with unicode names, because opencv isn't capable of that
- retinaface-pytorch - face recognition library used to detect faces in collected photos
- opencv - used for image manipulation and loading for retinaface-pytorch
- PIL.Image - used to access image EXIF data
- pandas - used for merging csv files
- wikipedia - used for extracting the first sentence of a wikipedia article
- matplotlib - although this library is not used in the current version of the scripts, it was used extensively for testing and manual annotation
- jupyter - this module is also not used in the scripts, still, it would have been almost impossible to develop the scripts without it

List of used modules from python standard library:

- csv - to manipulate csv files
- time - for measuring the program timing
- glob - for discovering all csv files
- re - for regular expressions
- os.path - for separating filename and extension
- pathlib - to create folders
- datetime - to handle and convert date formats
- webbrowser - for manual annotation of the sample set

5.3 Project structure

The example project structure can be seen in fig.5.1. The root directory (called project_directory) contains a single directory and three python files. Files *wikidownload.py* and *wikiprocess.py* are the scripts for downloading and processing. File *wikipeoplelib.py* is a module containing functions for the downloading script. Directory names in brackets with names in caps are placeholder names that depend on values of global variables used within scripts.
CSV_FOLDER - all csvs created by the download script are stored here, they are named with the respective category names, this folder also contains an info.txt file created by the downloading script if multiple years of death categories are processed
info.txt - text file containing information about the number of queried and downloaded wikipedia articles with the times the program took
IMG_FOLDER - all images downloaded by the download script are stored here
BIGGERFACE_FOLDER - if enabled, the processing script attempts to crop out the head and saves it into this folder
VIS_FOLDER - if enabled, the processing script saves original photos with visualised annotations from the face recognition software into this folder
PROC_FOLDER - all processed csvs are stored in this folder, their names are "processed"+[categoryname].csv, this folder also includes the file merged.csv, which is the final output of the processing script

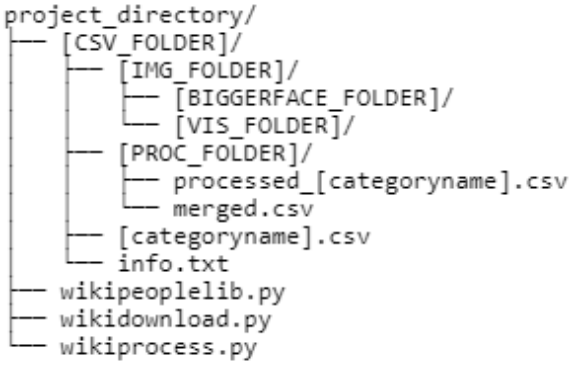


Figure 5.1: The project directory structure

The project pipeline is very straightforward. Both scripts check for existence of specified folders, creating them if necessary. The *wikidownload.py* script saves all downloaded data into csv files named by the respective categories and the images are downloaded into the IMG_FOLDER. Each row in these csv files contains the filename of a relevant image. These csv files are read by

the *wikiprocess.py* script, which does it's job, saves its output category-wise into the PROC_FOLDER and then creates a merged csv file as well.

5.4 Downloading script

The download script offers 3 helper functions and 1 macro-like function. Here are the function headers with the respective docstrings and extra explanations.

```
def livingppl(sampleSize=100,sortkey = "!ABCDEFGHIJKLMNOPQRSTUVWXYZ"):
    """
    Function to process the living people category from wikipedia
    :param sampleSize: Number of pages that will be processed for
                       each letter in sortkey
    :param sortkey: For every letter in sortkey, sampleSize pages
                    will be processed
    :return: Dictionary with stats for processed pages
    """
```

The livingppl function processes the Living people category from Wikipedia. The only important parameter is sampleSize. With the default value=100, the total amount of pages handled by the function is 100 * length of sortkey (27 in this case), therefore the function would handle up to 2700 pages on the default setting. The returned dictionary only contains numbers of processed and written pages as well as the time the whole process took.

```
def deaths(year,sampleSize=50,sortkey = "!ABCDEFGHIJKLMNOPQRSTUVWXYZ"):
    """
    Function to process the deaths of a specified year category
    from wikipedia
    :param year: the year specifying the deaths category
    :param sampleSize: Number of pages that will be processed
                       for each letter in sortkey
    :param sortkey: For every letter in sortkey, sampleSize
                    pages will be processed
    :return: Dictionary with stats for processed pages
    """
```

The deaths function process the "<year>_deaths" category on Wikipedia. The year parameter specifies the deaths category. The sampleSize parameter serves the same function as in the previous function. The returned dictionary has the same specifications as in the previous function.

```
def processYears(frm=2020,to=2000,sampleSize=50):
    """
    Function to process multiple years of death categories
    from wikipedia
    (going from present to past, so backwards)
    This function also creates an info.txt file with statistics
    about processed categories
    :param frm: The higher year from which the function starts
                (default 2020)
    :param to: The lower year from which the function starts
                (default 2000)
    :param sampleSize: Number of pages that will be processed
                       for each letter in the sortkey
    """
```

The processYears function calls the deaths() function in a loop, processing all categories in the specified year range. This function generates the info.txt file.

```
def downloadSampleData():
    """
    This function was used to download the sample data
    """
```

This is the macro-like function, which was used to create the sample set used in the next chapter.

All of these functions utilize the wikipeoplelib.py module, which contains functions for extracting specific data from the page and infobox objects. They all contain descriptive docstrings and anyone wishing to use them in their own scripts should have no problem doing so.

5.5 Processing script

Although the processing script is not separated like the downloading script, it still offers 1 helper function and 1 macro-like function. Here are the function headers with the respective docstrings and extra explanations.

```
def processCsv(filename,model,saveVis=True,
facecrop=True,startfrom=None,deaths=False):
    """
    Process a csv file of downloaded data
    :param filename: Name of the csv file
    :param model: the model to be used for face detection
    :param saveVis: Also save images with visualised rectangles
                    for faces
    :param facecrop: Also save images with just the face cropped
    :param startfrom: Start processing from a row with this name
    :param deaths: Require rows to contain the deathdate information
    """
```

The processCsv is the only helper function. It's purpose is to process a csv created by the downloading script. The mandatory parameters are the filename of the csv to be processed and the model from the pytorch-retinaface library, which will be used for face recognition. saveVis and facecrop parameters serve the purpose of saving visualised face annotations and cropped head images respectively. Startfrom is a string parameter which allows the user to start processing the csv from a specific row instead of start of the file. This is especially helpful on lower-end machines. The deaths parameter should be set to True if processing csvs for deaths categories.

```
def createSample():
    """
    Function to process the sample data downloaded by the sample
    function in the download script
    """
```

Once again a macro-like function, which processes the sample created by the downloading script and outputs the merged csv (as well as all the separate category processed csvs) used in the next chapter.

Chapter 6

Precision evaluation

In order to rate the performance of the project a comparison between the script output and ground truth has to be made. Ground truth refers to information which is known to be true. The simplest way to achieve this is to manually annotate a set of pages that has been processed by the scripts.

6.1 The sample set of pages

The sample set consists of the living people category with sampleSize=20 (at max 540 pages) and 2010-2020 deaths categories with sampleSize=2 (at max 54 pages per category for a total of 11 categories). The exact amounts are shown in tab.6.1.

-	Maxpages	After download	After processing
Living people	540 (100%)	192 (35%)	101 (18.7%)
Deaths	594 (100%)	164 (27.6%)	79 (13.3%)
Total	1134 (100%)	356 (31.4%)	180 (15.8%)

Table 6.1: Percentage of used pages in the sample set

6.2 Manual annotation

In order to manually annotate those 180 pages, I wrote a simple script, which opened the specific wikipedia, asked me for sex/gender, birthdate (deathdate) and imagedate. Then it proceeded to show me the image with visualised face annotation, I input if it was correct/usable or not. The last part was the script showing me the occupation and me confirming whether it is correct. All of the data was saved into a separate csv file, which was then compared with the original output file. Both of these helper scripts will be attached to the work as well as the csv file with the manual annotation.
Before we go over the results, the author reserves the right for a reasonable margin of error. Even though 180 pages may not seem like a high number, manually annotating them took multiple hours of monotonous work and even though the results were checked once over, it is still possible that the manual annotation contains a mistake or two.

Figure 6.1: Example of graffiti deceiving the face recognition

6.3 Evaluation results

The result of the comparison are the statistics presented in tab.6.2.

Total	**Total deaths**	**ages**	**genders**	
180	79	141 (78.3%)	180 (100%)	
birth dates	**death dates**	**occupation**	**faces**	**complete entries**
173 (96.1%)	79 (100%)	168 (93%)	177 (98.3%)	**130 (72.2%)**

Table 6.2: Results of the precision evaluation

The most important number is in the cell "complete entries". Out of 180 rows in the merged.csv file, 130, or 72.2% were considered correct.

The attributes with the highest precision rates were death dates and genders, all of which were considered correct.

The attribute with the lowest precision rate was age on the photo, scoring 78.3% of correct results.

Other attributes were birth dates scoring 96.1%, occupation scoring 93% and faces scoring 98.3%.

6.4 Discussing the results

Even though the sample set is on the smaller side, the results were somewhat expected after going through the manual annotation process.
High scoring death dates are not a surprise due to the nature of Wikipedia categorisation.
The most unexpected result was the sex/gender, especially because all the data came from a single source - Wikidata.
As for the low precision rate with ages - this test was unfortunately performed without 2 safety measures working properly - age range check and the image date being lower than the death date. By a quick manual check, the age score could possibly increase by 5%. Another reason as to why there were so many incorrect ages is most likely the misuse of the "Date" field in the summary on Commons file pages. Users seem to often use them to post upload dates instead of it's intended purpose, the original date. Nevertheless, even the achieved result would be considered good in my personal opinion.

6.5 Possible improvements

The areas which require improvements are obvious - it is mostly the low amount of correctly identified image dates that drag down the precision rate of the scripts. More sources for imagedates should be found as well as better decision making system when processing the collected data (based f.e. on counting of the occurences of values in collected data)

Use of other language versions of wikipedia could be considered, as they may contain other images with required information.

Lastly, a solution could be engineered to make use of third party projects that attempt to parse out information from wikipedia pages due to the difficulty of parsing them out with simple methods.

Chapter 7

Conclusion and future work

This thesis presented the method and scripts for automated downloading and processing of personal wikipedia homepages, converting them into a database of annotated facial images. We explained the purpose and function Wikipedia, Wikidata and Wikimedia Commons, the relationship between them and the benefits of their inter-connection.

We have analyzed the options for programmatically traversing Wikipedia and collecting it's data. In the end, we have chosen Pywikibot as a framework for collecting data. We have also explained the method with which we intend to collect, process and store data from wikipedia, as well as the important implementation details of the two scripts.

In the last part, we tested the script against a manually annotated sample set, achieving a success rate of over 72%. We discussed the results, the possible relevant causes and we suggested possible improvements on our work.

This thesis can serve as an introduction and a source of information on collecting data from Wikipedia and other related projects. It also provides a library for extracting specific information from Wikipedia and Wikidata, which could be used in future projects.

Appendix A

Bibliography

[1] Quora user, "What are the most hilarious examples of vandalism on wikipedia? - quora," 2012. [Online]. Available: https://www.quora.com/What-are-the-most-hilarious-examples-of-vandalism-on-Wikipedia

[2] Wikidata, "Bill gates — wikidata,," 2021. [Online]. Available: https://www.wikidata.org/wiki/Q5284

[3] Wikipedia contributors, "František křižík — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Franti%C5%A1ek_K%C5%99i%C5%BE%C3%ADk

[4] ——, "Murder of isabel cabanillas — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Murder_of_Isabel_Cabanillas

[5] Wikimedia Commons, "Donald trump - wikimedia," 2021. [Online]. Available: https://cs.wikipedia.org/wiki/Soubor:Donald_Trump_August_19,_2015_(cropped).jpg

[6] Wikipedia contributors, "Wikipedia — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Wikipedia&oldid=1038131725

[7] MediaWiki, "Differences between wikipedia, wikimedia, mediawiki, and wiki — mediawiki,," 2021. [Online]. Available: https://www.mediawiki.org/w/index.php?title=Differences_between_Wikipedia,_Wikimedia,_MediaWiki,_and_wiki&oldid=4646038

[8] Wikipedia, "Wikipedia — policies and guidelines," 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Wikipedia:Policies_and_guidelines&oldid=1036946849

[9] Meta, "Wikimedia projects — meta, discussion about wikimedia projects," 2021. [Online]. Available: https://meta.wikimedia.org/w/index.php?title=Wikimedia_projects&oldid=21517429

[10] Wikidata, "Introduction — wikidata,," 2021. [Online]. Available: https://www.wikidata.org/wiki/Wikidata:Introduction

[11] Wikipedia, "Purpose — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Wikipedia:Purpose&oldid=1037525740

[12] MediaWiki, "Api:main page — mediawiki,," 2021. [Online]. Available: https://www.mediawiki.org/w/index.php?title=API:Main_page&oldid=4710378

[13] ——, "Api:etiquette — mediawiki,," 2021. [Online]. Available: https://www.mediawiki.org/w/index.php?title=API:Etiquette&oldid=4577677

[14] ——, "Manual:maxlag parameter — mediawiki,," 2021. [Online]. Available: https://www.mediawiki.org/w/index.php?title=Manual:Maxlag_parameter&oldid=4649109

[15] ——, "Manual:pywikibot/en — mediawiki,," 2021. [Online]. Available: https://www.mediawiki.org/w/index.php?title=Manual:Pywikibot/en&oldid=4744202

[16] ——, "Manual:pywikibot/scripts — mediawiki,," 2021. [Online]. Available: https://www.mediawiki.org/w/index.php?title=Manual:Pywikibot/Scripts&oldid=4639622

[17] Wikipedia, "Living people — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Category:Living_people

Appendix B

List of attachments

- This thesis as a PDF file
- wikidownload.py - the download script
- wikiprocess.py - the processing script
- wikipeoplelib.py - the module containing functions use in the download script
- annotate.py - helper script that was used to manually annotate the sample set
- compare.py - helper script that was used to compare the manually annotated file with the script output
- annotated.csv - the manual annotation