Data mining II
Term project

Instructor Dr. NAIT-HAMOUD

Department of Second Cycle - fourth year Applied statistics
National higher School of Mathematics
Algiers, Algeria

Dec 22, 2025

1 Introduction

The aim of this project is to implement an entity linking system. Entity Linking (EL) is an information extraction task that links entity mentions present in a text to knowledge base entries such as Wikipedia, Dbpedia, etc. Figure 2 illustrates this task, the chunks of text highlighted in green represent mentions to entities coherent with the context of the respective texts where they occur. Mainly, the EL task can be boken down into two main sub-tasks.

- First, entity mentions (highlighted in the figure in green) must be located in the text, we refer to this sub-task as "recognition",
- Second, those mentions must be associated to the appropriate Wikipedia page.

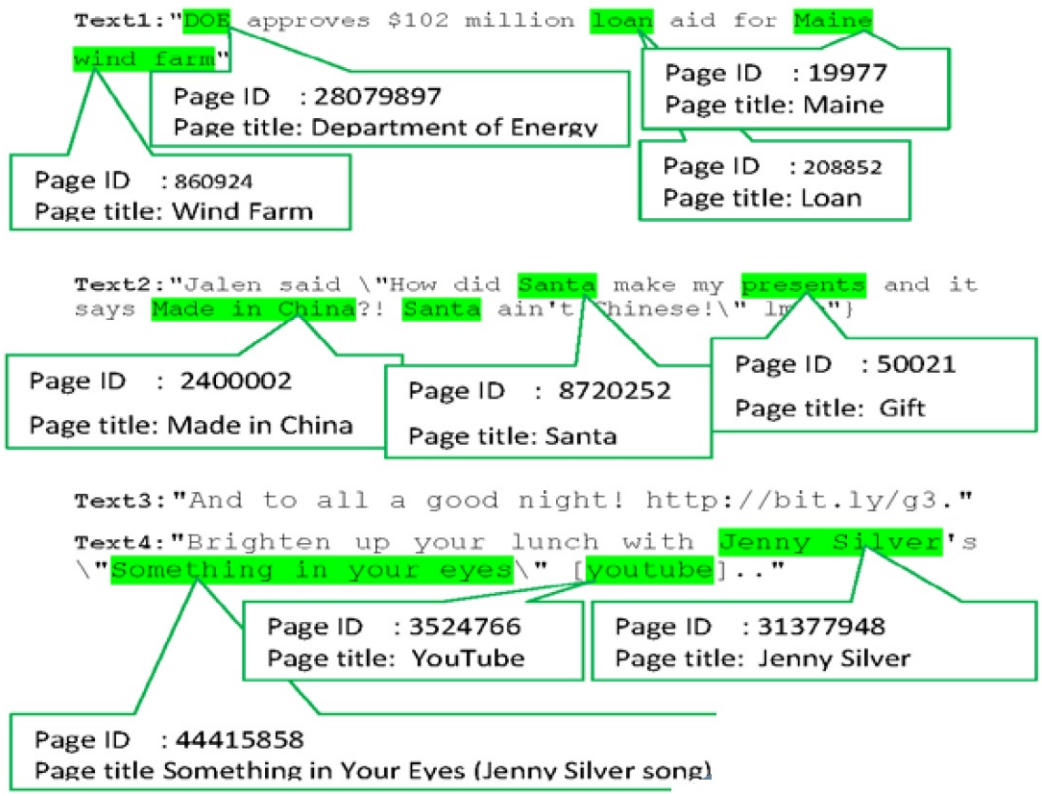


Figure 1: Example of linking mention to entities (Wikipedia page in this case)

Wikpedia knowledge base include three type of pages.

1. Redirect pages: redirects are used to make searching for information in Wikipedia easier,
2. Disambiguation pages (pages that do not cover a unique topic, but help to find the specific article when multiple topics share the same name),
3. Proper Wikipedia pages that cover a unique topic (see defintion 2 of section 4 of the provided paper).

One of the methods used by Wikipedia to search for pages is called **direct matching and redirection** that consists in two alternatives.

- If your search query exactly matches an article title, Wikipedia usually navigates you directly to that page instead of showing results (Exact Matches).

– It checks for "redirects"—alternative titles (like "Alan Mathison Turing" redirecting to "Alan Turing") to ensure you find the correct page even with variations (redirects).

For instance, figure 2 depicts the "Alan Turing" Wikipedia page. The information surrounded in red shows that there is a redirect page with the title "Turing" that redirects to the proper page of "Alan Turing". Besides, the information surrounded in green informs that there is a disambiguation page for the word "Turing". Figure 3 shows a snapshot of the disambiguation page corresponding to "Turing".

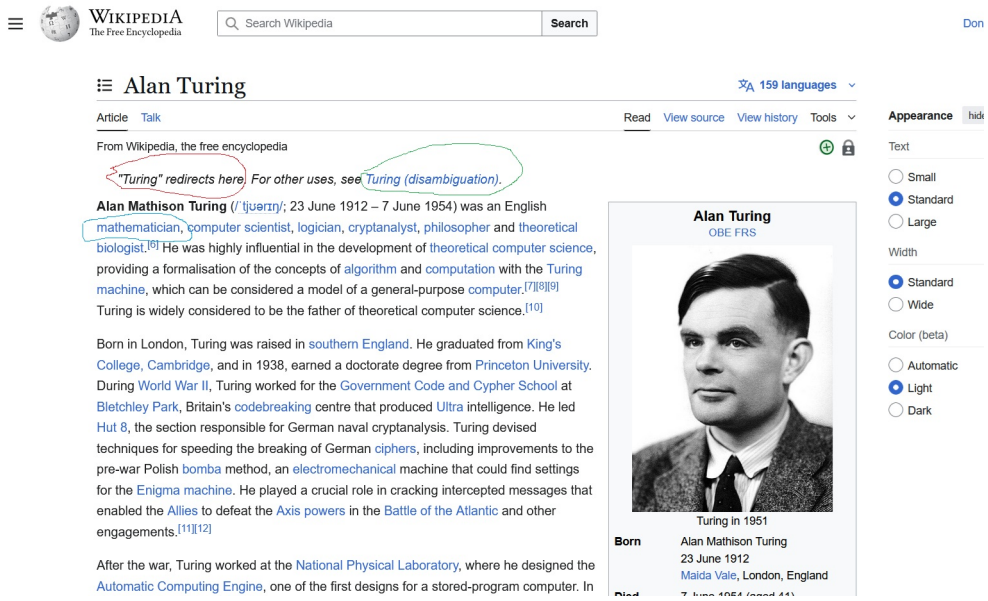


Figure 2: Example of a Wikipedia page: the text surrounded in blue indicates an example of an anchor (clickable text used to refer to a given wikipedia page)

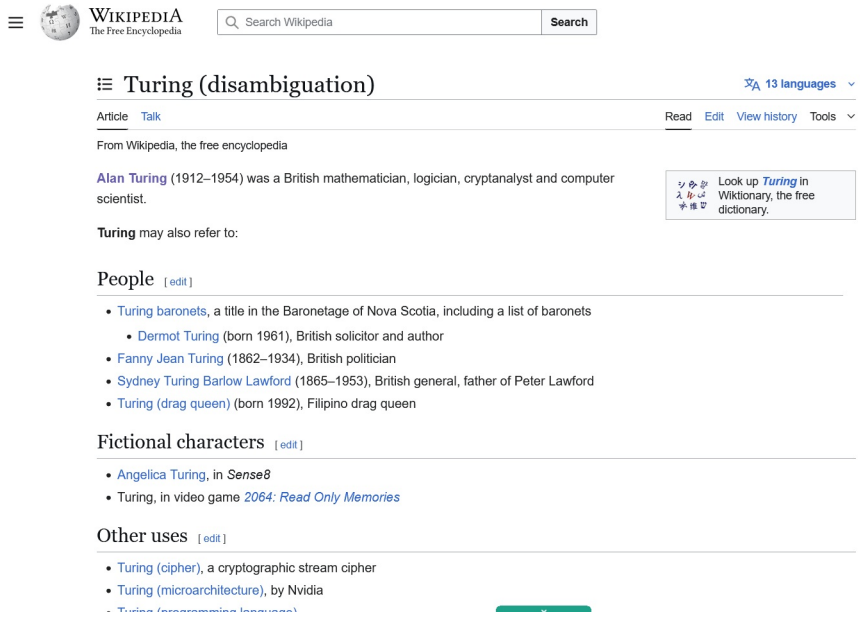
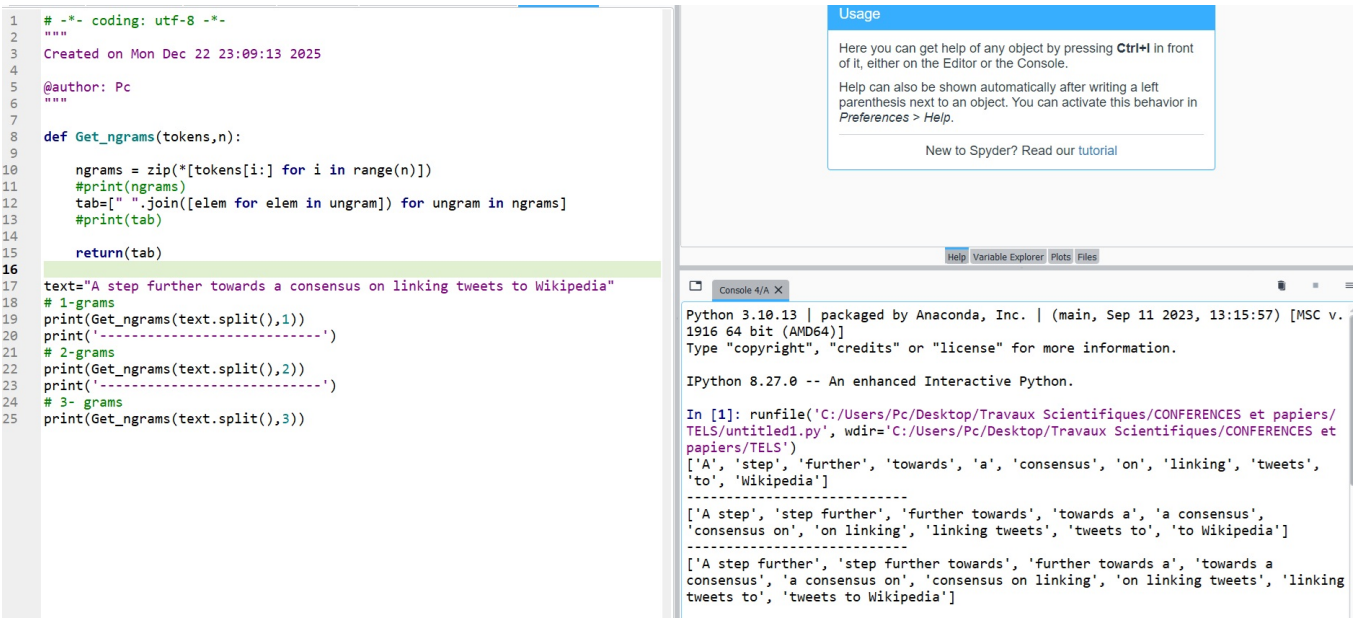


Figure 3: a snapshot of "Turing" Disambiguation page

2 Entity extraction and disambiguation

The EL system TELS (for Tweet Entity Linking System) introducted in the paper of reference [1] is and end-to-end system. This means that the two phases namely: *mention extraction* (or recognition) and *linking* are carried out in one shot. The authors in [1] did not extract the relevant mentions as a first phase, and in a second phase link the relevant extracted mention with the appropriate Wikipedia page.

After a preprocessing phase, described in [1], the authors extract the Ngrams from the cleaned text using a function similar to the one depicted in figure 4. Then, each Ngram is looked for in the inverted index, if it exists; candidate pairs (Ngram,PageID) are generated for each Page referenced by the Ngram. Afterwards, for each pair (Ngram,PageID) a feature vector is build on the basis of the selected features (see the work in [1]). Finally, a Model is trained using the provided revised Meij dataset.




```python
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 22 23:09:13 2025

@author: Pc
"""

def Get_ngrams(tokens,n):

    ngrams = zip(*[tokens[i:] for i in range(n)])
    #print(ngrams)
    tab=[" ".join([elem for elem in ungram]) for ungram in ngrams]
    #print(tab)

    return(tab)

text="A step further towards a consensus on linking tweets to Wikipedia"
# 1-grams
print(Get_ngrams(text.split(),1))
print('----------------------------')
# 2-grams
print(Get_ngrams(text.split(),2))
print('----------------------------')
# 3- grams
print(Get_ngrams(text.split(),3))
```

Figure 4: A python function to extract Ngrams from a text. If the Ngram position in the text is required the code should be enhanced

The provided material (files) for this project include the paper on TELS system [1] (to read carefully), prebuilt datasets (PostingsLast and PageIdToContext2), the python files to be used for reading records from these datasets, the gold standards, and the Meij dataset (multiple versions).

2.1 Provided prebuilt datasets and manipulation python code

For the needs of the project, the following datasets have been made available for you.

– **Inverted index**: A dictionary of anchors inlcuding redirect page titles was created from the English Wikipedia dump of January 2019. The dictionary of anchors is a key-value store that associates to each entry (Wikipedia anchor or redirect page title) the set of Wikipedia pages it refers to, along with a score representing the number of times the entry was used as cross-reference to each page.

The provided files to use for manipulating the inverted index are:

1. The Lmdb **PostingsLast**: To implement the database, LMDB (Lightening-Mapped Database) storage engine was used due to its high read performances. Moreover, to improve serialization, Google Protocol Buffers were used due to their best performances compared with XML, JSON and other existing formats.
2. **InvertedIndexAccess.py**: can be used, through the function call **IndexAccess(ngram,trxn)**, to read from the lmdb **PostingsLast**. When you run the python code in InvertedIndexAccess.py,

you are asked to enter a mention (text) **m** or an ngram, if this mention was used as an anchor, or z redirect page, in Wikipedia to refer to specific pages, the result is a list of records. Each record include the page ID of the referenced page, the score and the type of the referenced page.

- **pageId**: The ID of the Wikpedia page
- **score**: It tells you how many time the text or the mention **m** was used, as an anchor, in the whole Wikipedia to refer to this page.
- **type**: 0 for anchor (the Ngram was used as an anchor), 1 for redirect (the Ngram was used as a redirect page title) and 2 for both (the Ngram was used as a redirect page title and as an anchor to cross-reference the page)

3. SerializedListNew_pb2 (it is the defined **Google protocol buffer**), it should be imported in the file InvertedIndexAccess)

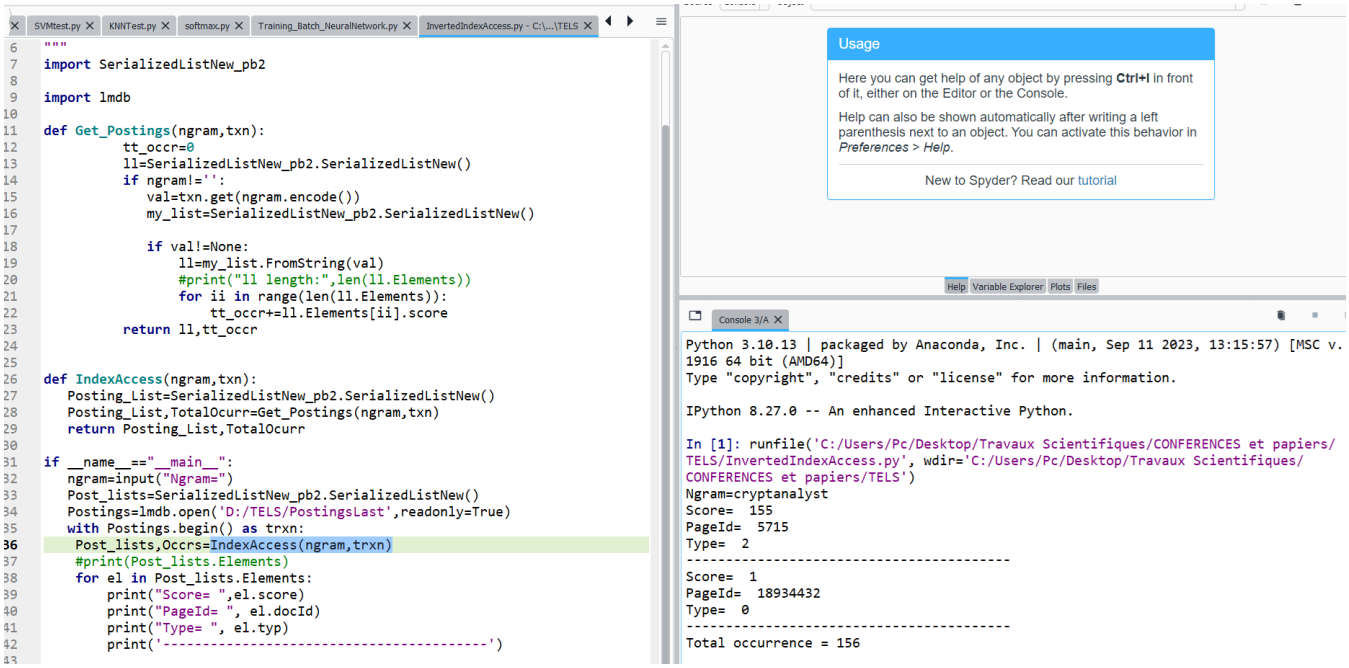
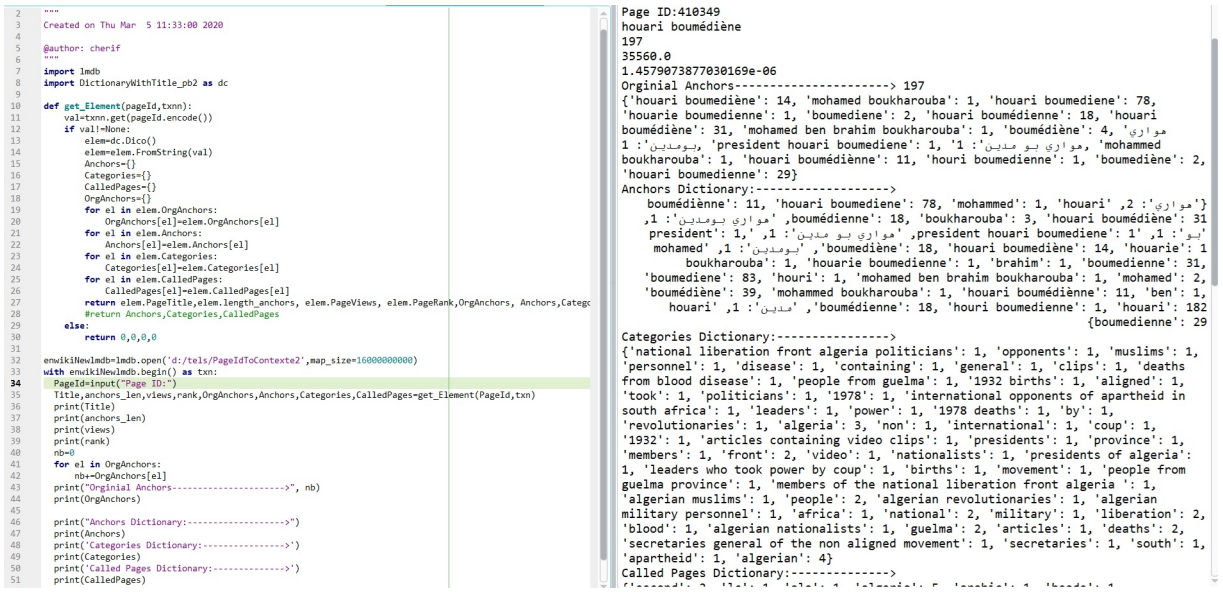


Figure 5: Example of a search for the Ngram "cryptanalyst" in the inverted index

- **The dataset of contexts PageIdToContext2**: It is also a key-value store that includes the following information for each Wikipedia page:
  1. Page title,
  2. List of anchors and redirect page titles that link to this page,
  3. Categories of the page included in the Wikipedia categories section,
  4. list of anchors and titles of Wikipedia pages referenced in the abstract,
  5. number of times the page was visited during the year 2019,
  6. Page rank of the page in the Wikipedia knowledge base graph.
- **InterrogatePageIdToContextLmdb.py**: This python code allows for interrogating the database (lmdb) **PageIdToContext2** that represents the dataset of contexts of each Wikipedia page. Figure 6 shows an example of the execution of this python code.

Each Wikipedia page include a "categories" section. This latter is a navigational system used to group related articles. They allow to browse through related topics even if they do not know the exact title of a specific page. Figure 7 shows the "categories" section of the Wikipedia page entitled "Alan Turing".



```
"""
Created on Thu Mar  5 11:33:00 2020

@author: cherif
"""
import lmdb
import DictionaryWithTitle_pb2 as dc

def get_Element(pageId,txnn):
    val=txnn.get(pageId.encode())
    if val!=None:
        elem=dc.Dico()
        elem=elem.FromString(val)
        Anchors={}
        Categories={}
        CalledPages={}
        OrgAnchors={}
        for el in elem.OrgAnchors:
            OrgAnchors[el]=elem.OrgAnchors[el]
        for el in elem.Anchors:
            Anchors[el]=elem.Anchors[el]
        for el in elem.Categories:
            Categories[el]=elem.Categories[el]
        for el in elem.CalledPages:
            CalledPages[el]=elem.CalledPages[el]
        return elem.PageTitle,elem.length_anchors, elem.PageViews, elem.PageRank,OrgAnchors, Anchors,Categ
        #return Anchors,Categories,CalledPages
    else:
        return 0,0,0,0

enwikiNewlmdb=lmdb.open('d:/tels/PageIdToContexte2',map_size=16000000000)
with enwikiNewlmdb.begin() as txn:
  PageId=input("Page ID:")
  Title,anchors_len,views,rank,OrgAnchors,Anchors,Categories,CalledPages=get_Element(PageId,txn)
  print(Title)
  print(anchors_len)
  print(views)
  print(rank)
  nb=0
  for el in OrgAnchors:
      nb+=OrgAnchors[el]
  print("Orginial Anchors---------------------->", nb)
  print(OrgAnchors)

  print("Anchors Dictionary:------------------->")
  print(Anchors)
  print('Categories Dictionary:---------------->')
  print(Categories)
  print('Called Pages Dictionary:-------------->')
  print(CalledPages)
```

```
Page ID:410349
houari boumédiène
197
35560.0
1.4579073877030169e-06
Orginial Anchors----------------------> 197
{'houari boumediène': 14, 'mohamed boukharouba': 1, 'houari boumediene': 78,
'houarie boumedienne': 1, 'boumediene': 2, 'houari boumédienne': 18, 'houari
boumédiène': 31, 'mohamed ben brahim boukharouba': 1, 'boumédiène': 4, 'هواري
1 :'بومدين, 'president houari boumediene': 1, '1 :'هواري بو مدين, 'mohammed
boukharouba': 1, 'houari boumédiènne': 11, 'houri boumedienne': 1, 'boumediène': 2,
'houari boumedienne': 29}
Anchors Dictionary:------------------->
   boumédiènne': 11, 'houari boumediene': 78, 'mohammed': 1, 'houari' ,2 :'هواري'}
  ,1 :'هواري بومدين' ,boumédienne': 18, 'boukharouba': 3, 'houari boumédiène': 31
   president': 1,' ,1 :'هواري بو مدين' ,president houari boumediene': 1' ,1 :'بو'
    mohamed' ,1 :'بومدين' ,'boumediène': 18, 'houari boumediène': 14, 'houarie': 1
         boukharouba': 1, 'houarie boumedienne': 1, 'brahim': 1, 'boumedienne': 31,
  'boumediene': 83, 'houri': 1, 'mohamed ben brahim boukharouba': 1, 'mohamed': 2,
  'boumédiène': 39, 'mohammed boukharouba': 1, 'houari boumédiènne': 11, 'ben': 1,
      houari' ,1 :'مدين' ,'boumédienne': 18, 'houri boumedienne': 1, 'houari': 182
                                                                  {boumedienne': 29
Categories Dictionary:---------------->
{'national liberation front algeria politicians': 1, 'opponents': 1, 'muslims': 1,
'personnel': 1, 'disease': 1, 'containing': 1, 'general': 1, 'clips': 1, 'deaths
from blood disease': 1, 'people from guelma': 1, '1932 births': 1, 'aligned': 1,
'took': 1, 'politicians': 1, '1978': 1, 'international opponents of apartheid in
south africa': 1, 'leaders': 1, 'power': 1, '1978 deaths': 1, 'by': 1,
'revolutionaries': 1, 'algeria': 3, 'non': 1, 'international': 1, 'coup': 1,
'1932': 1, 'articles containing video clips': 1, 'presidents': 1, 'province': 1,
'members': 1, 'front': 2, 'video': 1, 'nationalists': 1, 'presidents of algeria':
1, 'leaders who took power by coup': 1, 'births': 1, 'movement': 1, 'people from
guelma province': 1, 'members of the national liberation front algeria ': 1,
'algerian muslims': 1, 'people': 2, 'algerian revolutionaries': 1, 'algerian
military personnel': 1, 'africa': 1, 'national': 2, 'military': 1, 'liberation': 2,
'blood': 1, 'algerian nationalists': 1, 'guelma': 2, 'articles': 1, 'deaths': 2,
'secretaries general of the non aligned movement': 1, 'secretaries': 1, 'south': 1,
'apartheid': 1, 'algerian': 4}
Called Pages Dictionary:-------------->
```

Figure 6: Example of the execution of the python code InterrogatePageIdToContextLmdb.py

v·t·e **Alan Turing** [show]

v·t·e **Fellows of the Royal Society elected in 1951** [show]

v·t·e **Timelines of computing** [show]

v·t·e **Early history of video games (1947–1971)** [show]

Authority control databases [show]

Portals: Biography | England | LGBTQ | Greater Manchester | Mathematics | Computer programming

Categories: Alan Turing | 1912 births | 1954 deaths | 1954 suicides | 20th-century atheists | 20th-century English LGBTQ people | 20th-century English mathematicians | 20th-century English philosophers | Academics of the University of Manchester Institute of Science and Technology | Academics of the University of Manchester | Alumni of King's College, Cambridge | Bayesian statisticians | Bletchley Park people | British anti-fascists | British artificial intelligence researchers | British cryptographers | British people of World War II | Castrated people | Computability theorists | Computer chess people | Computer designers | Early history of video games | English atheists | English computer scientists | English gay sportsmen | English inventors | English LGBTQ scientists | English logicians | English men long-distance runners | British men long-distance runners | English people of Irish descent | English people of Scottish descent | Enigma machine | Fellows of King's College, Cambridge | Fellows of the Royal Society | Foreign Office personnel of World War II | Former Protestants | Gay academics | Gay scientists | GCHQ people | History of computing in the United Kingdom | LGBTQ mathematicians | LGBTQ philosophers | LGBTQ track and field athletes | LGBTQ people who died by suicide | Officers of the Order of the British Empire | People convicted for homosexuality in the United Kingdom | People educated at Sherborne School | People from Maida Vale | People from Wilmslow | People who have received posthumous pardons | Princeton University alumni | Recipients of British royal pardons | Scientists of the National Physical Laboratory (United Kingdom) | Suicides by cyanide poisoning | Suicides in England | Theoretical biologists | British theoretical computer scientists | People from St Leonards-on-Sea

Figure 7: Categories of the Wikipedia page corresponding to Alan Turing

2.2 Gold standards and training set

The gold standards and the Meij datasets, used in [1], are also provided for the comparison of your results with the baselines: TagMe, AIDA, DBpedia Spotlight and WAT.

Each gold standard is provided in a separate folder that includes two files:

- a "tsv" file for the tweet texts
- a "tsv" file for the annotations.

- **Remark:** The columns in the annotation file of the gold standards are not the same.

3 Required tasks

1. Train a model, using Meij revised dataset, with at least two different techniques seen and experienced during lab and lecture sessions (**DNN** is mandatory). your code must be modular and include:
   - a text processing module,
   - a feature extraction module (use at least 7 features defined in [1])
2. Test your models on all the provided gold standards, as in the work in [1])
3. Compare your implemented EL system with TagMe and AIDA.
4. Provide a detailed and well-organized report on your project, including an argumentation for all your choices.

4 Submission deadline

The deadline will be announced later.

References

[1] Nait-Hamoud, M.C., Lahfa, F. & Ennaji, A. *A step further towards a consensus on linking tweets to Wikipedia*. Evol. Intel. 16, 1825–1840 (2023). https://doi.org/10.1007/s12065-020-00549-8