INF344 2019–2020

Tableau de bord / Mes cours / INF344 2019–2020 / Disambiguation, Web sémantique / Disambiguation

Description Submission view

Disambiguation

Due date: Thursday 18 June 2020, 23:59
Requested files: simpleKB.py, parser.py, page.py, disambiguate.py (Download)
Nombre maximal de fichiers: 20
Type of work: Individual work
Reduction by automatic evaluation: 10 **Free evaluations**: 2

Purpose

The goal of this lab is to disambiguate entities in a text. For example, given a Wikipedia article

```
    Paris_17
    Paris is a figure in the Greek mythology.
```

... the goal is to determine that

```
    Paris_17 = https://www.wikidata.org/wiki/Q167646
```

Here, https://www.wikidata.org/wiki/Q167646 is the URI of the Greek hero in Wikidata. Paris_17 is an artificial title of the Wikipedia article.

Task

Your task is to edit the file disambiguate.py to disambiguate every Wikipedia article given in input. In the previous example, one would call:

```
print("Paris_17", "<Q167646>", sep="\t", file=output)
```

in order to register the answer. The grade will take into account precision and recall (f-0.5 measure).

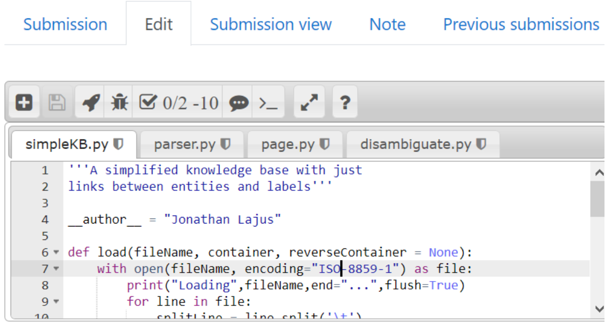
Resources

You find here https://louis.jachiet.com/tmp/Yo9kDfnWsYb_files.zip

- A simplified Wikidata, which consists just of the links between entities (without relation names), dates related to entities and the labels of entities.
- A prepared corpus of ambiguous Wikipedia pages (of the form shown above).
- A sample of a gold standard for the task.

Code

This lab can be programmed, run, debugged, evaluated, and graded directly in the Moodle.



```
'''A simplified knowledge base with just
links between entities and labels'''

__author__ = "Jonathan Lajus"

def load(fileName, container, reverseContainer = None):
    with open(fileName, encoding="ISO-8859-1") as file:
        print("Loading",fileName,end="...",flush=True)
        for line in file:
            splitLine = line.split('\t')
```

In the “Edit” tab, you will find:

- The Python module Page, which represents a page in Wikipedia.
- The Python module Parser, which allows iterating over the pages of a Wikipedia corpus.
- The Python module simpleKB, a simplified knowledge base.
- The Python module disambiguate, a skeleton that you should modify.

You can directly modify the code in the Web interface. Click the disk symbol to save. Click the bug to debug the code. Click the rocket to run it. This will also automatically compute precision and recall.

If you want to work offline: You can download the code files, modify them, and then upload them (in the tab “Submission”). The Web interface already knows Wikidata and Wikipedia, so you do not need to upload them.

Submitting the lab

When you are done, you can submit the lab (click the checkbox). This will directly compute an estimated grade on our gold standard. You can submit 2 times. If you submit more often, you lose points.

Plagiarism is sanctioned with a grade of 0/20 (plus potentially other sanctions). Submissions that share material all receive grade 0/20, no matter who is the author of the code.

NOTE: The submission server only works inside Télécom. Please visit https://www.telecom-paristech.fr/vivre-ecole/services-numeriques-dsi/connexion-depuis-lexterieur.html to know how to connect to the internal network from the outside.

Requested files

simpleKB.py

```python
'''A simplified knowledge base with just
links between entities and labels'''

__author__ = "Jonathan Lajus"

def load(fileName, container, reverseContainer = None):
    with open(fileName, encoding="utf-8") as file:
        print("Loading",fileName,end="...",flush=True)
        for line in file:
            splitLine = line.split('\t')
            if len(splitLine) is not 2:
                raise RuntimeError('The file is not a valid KB file')
            subject=splitLine[0]
            obj=splitLine[1].strip('"\n')
            container.setdefault(subject,set()).add(obj)
            if reverseContainer!=None:
                reverseContainer.setdefault(obj,set()).add(subject)
        print("done",flush=True)

class SimpleKB:
    def __init__(self, yagoLinksFile, yagoLabelsFile, yagoDatesFile = None):
        self.links = {}
        self.labels = {}
        self.rlabels = {}
        self.dates = {}
        load(yagoLinksFile, self.links, self.links)
        load(yagoLabelsFile, self.labels, self.rlabels)
        if yagoDatesFile:
            load(yagoDatesFile, self.dates, None)

```

parser.py

```python
'''Parses a Wikipedia file, returns page objects'''
from page import Page
__author__ = "Jonathan Lajus"

class Parser:
    def __init__(self, wikipediaFile):
        self.file = wikipediaFile
    def __iter__(self):
        title, content = None,""
        with open(self.file, encoding='utf-8') as f:
            for line in f:
                line = line.strip()
                if not line and title is not None:
                    yield Page(title, content.rstrip())
                    title, content = None,""
                elif title is None:
                    title = line
                elif title is not None:
                    content += line + " "
```

page.py

```python
import sys

class Page:
    def __init__(self, title, content):
        self.content = content
        self.title = title
        if sys.version_info[0] < 3:
            self.title = title.decode("utf-8")
            self.content = content.decode("utf-8")

    def __eq__(self, other):
        return isinstance(other, self.__class__) and self.title == other.title and self.content == other.content

    def __ne__(self, other):
        return not self.__eq__(other)

    def __hash__(self):
        return hash((self.title, self.content))

    def __str__(self):
        return 'Wikipedia page: "'+(self.title.encode("utf-8") if sys.version_info[0] < 3 else self.title)+'"'

    def __repr__(self):
        return self.__str__()

    def _to_tuple(self):
        return (self.title, self.content)

    # Only used for Disambiguation TP
    def label(self):
        return self.title[1:self.title.rindex("_")].replace("_", " ")

```

disambiguate.py

```python
usage='''
  Given as command line arguments
  (1) wikidataLinks.tsv
  (2) wikidataLabels.tsv
  (optional 2') wikidataDates.tsv
  (3) wikipedia-ambiguous.txt
  (4) the output filename'''
'''writes lines of the form
        title TAB entity
  where <title> is the title of the ambiguous
  Wikipedia article, and <entity> is the
  wikidata entity that this article belongs to.
  It is OK to skip articles (do not output
  anything in that case).
  (Public skeleton code)'''

import sys
import re
from parser import Parser
from simpleKB import SimpleKB

wikidata = None
if __name__ == "__main__":
    if len(sys.argv) is 5:
        dateFile = None
        wikipediaFile = sys.argv[3]
        outputFile = sys.argv[4]
    elif len(sys.argv) is 6:
        dateFile = sys.argv[3]
        wikipediaFile = sys.argv[4]
        outputFile = sys.argv[5]
    else:
        print(usage, file=sys.stderr)
        sys.exit(1)

    wikidata = SimpleKB(sys.argv[1], sys.argv[2], dateFile)

# wikidata is here an object containing 4 dictionaries:
## wikidata.links is a dictionary of type: entity -> set(entity).
##                It represents all the entities connected to a
##                given entity in the yago graph
## wikidata.labels is a dictionary of type: entity -> set(label).
##                It represents all the labels an entity can have.
## wikidata.rlabels is a dictionary of type: label -> set(entity).
##                It represents all the entities sharing a same label.
## wikidata.dates is a dictionnary of type: entity -> set(date).
##                It represents all the dates associated to an entity.

# Note that the class Page has a method Page.label(),
# which retrieves the human-readable label of the title of an
# ambiguous Wikipedia page.

    with open(outputFile, 'w', encoding="utf-8") as output:
        for page in Parser(wikipediaFile):
            # DO NOT MODIFY THE CODE ABOVE THIS POINT
            # or you may not be evaluated (you can add imports).

            # YOUR CODE GOES HERE:
            pass

```

VPL

◄ Semantic web

Aller à…

Captation Disambiguation ►