CS132a General instructions for programming assignments

Every programming assignment should include a **readme.pdf** file. This file should include all the information needed for a reader unfamiliar with the assignment to understand what your code does, how to build and use it.

It should contain the following sections.

Title: Use the full title of the assignment
Author: Your name
Date:
Description: A sentence or two describing what the program does at a high level
Dependencies: List all software required to run the system. Include version and, where appropriate, a url where the software was obtained. Avoid code or dependencies that are operating system dependent. Code should run identically under OSX and Windows. Python has OS independent ways of handling pathnames.
Build Instructions: Describe clearly how to build the system as a sequence of steps.
Run Instructions: Describe how to run the system. What is the input and output produced? If there is a user interface, describe how to use it. What is legal input? How does the system handle boundary conditions or inappropriate input?
Modules: Give an overview of each module (file) in your system, its purpose, the major data structures used, and a high level description of the key functions/methods within it.
Testing: Describe how the system was tested to ensure it is working properly.

Submission: The readme.pdf and code, data files should be submitted as a single zip file to latte.

CS132a spring 2017 Assignment 2: Building a Corpus

A2 (due before class 2/3/17): In this assignment you will gain experience creating a corpus for indexing and retrieval. You will extract pages from Wikipedia, identify fielded data, clean unstructured data, and store the results in json format.

First steps:

1. Install the latest Python 2.7x version (not Python 3!)
2. Install Python package wikitools ('pip install wikitools' or https://pypi.python.org/pypi/wikitools)
3. In order to overcome the default limit of 500 files for getAllMembers, you will need to make an edit in your wikitools library code.

   Go to where you installed wikitools (usually in 'Lib/site-packages') and make the following change:

   In the file 'category.py', line 113, replace:

```
params['cmcontinue'] = data['query-continue']['categorymembers']['cmcontinue']
```

   with

```
params['cmcontinue'] = data['continue']['cmcontinue']
```

Extracting a corpus from Wikipedia:

Using the Python wikitools module, get all pages (not including sub-categories) of the category "2016 films" (https://en.wikipedia.org/wiki/Category:2016_films).

To get you started:

```
wikiobj = wiki.Wiki("https://en.wikipedia.org/w/api.php")
wikicat = category.Category(wikiobj, title = "2016_films")
wikipages = wikicat.getAllMembers()
```

For each page, you should then extract the following fields:

- Title
- Director
- Starring
- Running_time (in minutes)
- Country
- Language
- Time (what year(s) the story takes place)
- Location (city/country(s) where the story takes place)
- Categories (list of Wikipedia categories that the page belongs to)
- Text (the full text of the page, excluding the data extracted from templates)

Storing the data:

About json format https://en.wikipedia.org/wiki/JSON:

- "Javascript object notation"
- Standard for human readable data transfer
- Language independent
- Based on attribute-value pairs
- Basic types:
  - Number
  - String
  - Boolean
  - Array []
  - Object (key/value pairs) {}

Notice that the json format is similar to Python dictionaries. Your program should build a python dictionary and then use json.dump to write the json formatted output to a file called "2016_movies.json" (that can be json.loaded afterwards).

Each movie document should be associated with a unique integer id, serving as the primary key. Each document will be stored as a nested dictionary, where the keys are the extracted field names (e.g. 'Title', 'Director',…), and the values are either strings, integers or lists:

```
{1:
        {'Title': 'Absolution',
```

'Director: ['Keoni Waxman'],
'Starring': ['Steven Seagal', ...]
'Running_time': 95
.
.
.
},
2:
{...
}

}

Note: integer keys will be output as strings in json.

What you need to submit (via latte):

- Python code of the Wikipedia extraction and storing.
  - This code should include internal documentation, i.e., comments briefly describing what each data structure and function does.
- The json file you have created.
- readme.pdf, as described above, including a description of the heuristics you used in order to extract fields, clean, and/or organize the data.

Notes and hints

1. Press the "edit" button in the top right corner of the Wikipedia page to explore the data format obtained by wikitools.
2. Use the "infobox" section of the Wikipedia page to get most of the fields required.
3. Many of the fields can have different names (e.g. "Running time" vs. "Duration") or are missing from the infobox. Try, by iterating and re-checking, to populate correctly as many fields as you can.
4. For the fields that can potentially contain more than one value (e.g. "starring"), store the values in a list (the json format supports lists).
5. In order to populate the missing fields, you can use any useful data in the page. Be creative.
6. The data in all the fields should be as clean as possible. Make sure to detect and remove meta data tags like links and headers.
7. For assignment 3, you will need to tokenize/normalize your textual data for use in indexing. You do **NOT** need to do this for assignment 2.
8. The *text* field (and the other fields as well) should contain clean text without marks/tags. You can do it by either detecting and handling the common patterns yourselves, or by using some other external Python package you may find (if so, please mention it in the write-up).
9. Notice that some of the pages retrieved through "getAllMembers" are actually categories (with "Category: - " in them), and should be ignored.
10. In case the page represents a film *series* (thus containing different running_time, release date, etc. for each of the parts), just choose the values of the first part (to keep it consistent with other films).
11. The *text* field should contain all the “free text’ in the article that is not included in templates. You can use some discretion about what to include/exclude. Anything that a user might want to include in a query should be included, but you can omit links, markup tags, and embedded templates (such as references and external links).

12. In some cases, it may not be possible to determine the value for a field, even using heuristics (e.g., Time, Location).

Example json output:


```
{
  "344": {
    "language": "English",
    "title": "Bridgend",
    "country": "Denmark/United States/United Kingdom",
    "time": [],
    "director": "Jeppe Ronde",
    "location": [
      "Wales"
    ],
    "starring": [
      "Hannah Murray",
      "Josh O'Connor",
      "Adrian Rawlins"
    ],
    "text": "\n\nBridgend is a 2015 Danish drama film directed by Jeppe Rnde and written by Rnde alongside Torben Bech and Peter Asmussen. The film is based on true events that happened in the welsh town Bridgend, and it consist on the mystery Bridgend suicides of South Wales . The film had its World Premiere International Film Festival Rotterdam and North American premiere at the Tribeca Film Festival, in both it was completely acclaimed and received 3 awards in the last one, including Best Actress to Hannah Murray.\n\n==Cast==\nHannah Murray as Sara\nSteven Waddington as Dave\nJosh O'Connor as Jamie\nAdrian Rawlins as Vicar\nPatricia Potter as Rachel\nAled Thomas as Danny\nElinor Crawley as Laurel\nScott Arthur as Thomas \nJamie Burch as Angus\n\n==Awards and nominations==\n class=\"wikitable sortable\"\n\n Year  Award  Category  Result\n\n 2015  International Film Festival Rotterdam  Hivos Tiger Award \n\n 2015  Tribeca Film Festival  Best Narrative Feature  \n\n2015  Tribeca Film Festival  Best Actress \n\n 2015  Tribeca Film Festival  Best Cinematography \n\n 2015  Tribeca Film Festival  Best Editor \n\n 2015 Gteborg International Film Festival  Special Mention, Debut Award \n\n 2015  Brussels Film Festival Distribution Award \n\n 2015  Montreal Fantasia Int Film Festival  Prix AQCC - Critics Award \n\n 2015 Mannheim-Heidelberg, Int. Film Festival  Special Mention - Cinematography \n\n 2015  Cracow Off Camera International Film Festival  Best Cinematography \n\n 2015  Ourense Independent Film Festival  Best Actress \n\n 2015  Palma de Mallorca Evolution IFF  Best Actress  \n\n 2015  Palma de Mallorca Evolution IFF  Best Cinematography\n\n",
    "runtime": "95",
    "categories": [
      "2015 films",
      "2010s drama films",
      "English-language films",
      "British films",
      "Danish drama films",
      "Danish films",
      "Films set in Wales"
    ]
  },
  "345": {
    "language": "English",
    "title": "Broken Horses",
    "country": "United States",
    "time": [],
    "director": "Vidhu Vinod Chopra",
    "location": [],
    "starring": [
      "Vincent D'Onofrio",
```

```
        "Anton Yelchin",
        "Chris Marquette",
        "Maria Valverde",
        "Thomas Jane"
      ],
      "text": " Broken Horses is a 2015 American mystery thriller film directed by Vidhu Vinod Chopra and starring Mara Valverde, Thomas Jane, Anton Yelchin, Vincent D'Onofrio and Sean Patrick Flanery. It was released on April 10, 2015. The film is a remake of the 1989 Hindi film Parinda.\n\n==Plot==\nFirmly in the tradition of American Westerns, it follows the lives of the two orphaned brothers. The older one, Buddy, sees his father being shot. Vulnerable and described as slow, Buddy gets co-opted by gangster Julius Hench (Vincent DOnofrio) and turns into his key assassin. While Buddy grows up in a lawless environment, younger brother Jakey is a violinist auditioning for the New York Philharmonic and on the verge of marrying his Portuguese girlfriend. But first Jakey must return to his dusty home town near the U.S.-Mexican border to receive his wedding present from his older brother. Returning to that one-horse town opens up unhealed wounds and forces Jakey and Buddy to confront some ugly truths. \n\nHench wont let Buddy quit the job. He will do anything to keep his most efficient, easily manipulated killing machine on his rolls, including bumping off Jakey. When Jakey realizes what Buddy is up against, he orchestrates a rather poorly designed plan to help them both escape from Hench.\n\n==Cast==\nVincent D'Onofrio as Julius Hench\nAnton Yelchin as Jacob Heckum\nNicholas Neve as Jacob (during adolescence)\nChris Marquette as Buddy Heckum\nHenry Shotwell as Buddy (during adolescence)\nMara Valverde as Vittoria\nThomas Jane as Gabriel Heckum\nSean Patrick Flanery as Ignacio\nWes Chatham as Ace\nGreg Serano as Miguel Santion\nSadie Alexandru as Santion's Wife\nJeremy Luke as Franco\nJuan Riedinger as Eric\nJordi Caballero as Mario Vargos Garza\n\n==Production==\nPrincipal photography began on October 29, 2012, in and around Los Angeles.\n\nThe teaser trailer for the film was released on December 19, 2014. Reliance Entertainment is presenting the film together with Vinod Chopra Films.\n\n==Reception==\n\n===Critical reception===\nThe film received mixed reception from the critics. Writing for Variety, Ben Kenigsberg gave a mixed review and said, \"This overwrought tale of two orphaned brothers and their violent hometown reunion fails to convince on several crucial levels, including plotting and dialogue. Despite name cast members and ace work from regular Clint Eastwood d.p. Tom Stern, the audience for this curio exists mainly in the Twilight Zone, which is where the movie often seems to be set.\"\n\n",
      "runtime": "101",
      "categories": [
        "2015 films",
        "2010s crime thriller films",
        "2010s mystery films",
        "American films",
        "American crime thriller films",
        "American mystery films",
        "English-language films",
        "Films about drugs",
        "Films directed by Vidhu Vinod Chopra",
        "Films shot in California",
        "Films shot in Los Angeles, California",
        "Mandeville Films films",
        "Sony Pictures Classics films",
        "Film scores by John Debney"
      ]
    }
}
```