DSCI 558 Building Knowledge Graphs

# Homework 2: Knowledge Representation + Entity Resolution

## Released: Sep 12, 2022 Due: Sep 19, 2022 @ 23:59 PST

### Ground Rules

This homework must be done individually. You can ask others for help with the tools, however, the submitted homework must be your own work.

### Summary

In this homework, you will perform

1. Record Linkage between two different datasets (Goodreads, Barnes and Nobles) using [Record Linkage ToolKit](https://rltk.readthedocs.io/) (RLTK)
   1. Implement and evaluate your **blocking** techniques
   2. Implement and evaluate your **entity linking** techniques
2. Represent data in **RDF** using [RDFLib](https://github.com/RDFLib)

We provide a python notebook (*ER\_KR.ipynb*), which contains instructions, code and descriptions on how to use the tools we mentioned.

### Submission Instructions

You must submit (via Blackboard) the following files/folders in a single .zip archive named **Firstname\_Lastname\_hw01.zip**:

* Source Code: **Firstname\_Lastname\_blocked.ipynb** (.py is also acceptable)
* Blocked pairs from Task 1.2: **Firstname\_Lastname\_blocked.csv**
* Predictions from Task 1.3: **Firstname\_Lastname\_predictions.csv**
* Valid Predictions from Task 1.4: **Firstname\_Lastname\_valid\_predictions.csv**
* Model file from Task 2.1: **Firstname\_Lastname\_model.ttl**
* Model visualization from Task 2.2: **Firstname\_Lastname\_viz.png/jpeg**
* Explanations for Task 1 & 2: **Firstname\_Lastname\_report.pdf**

### Task 1: Entity Resolution (Record Linkage) - 5 points

Your goal is to match records from **Goodreads** to those in **Barnes and Nobles**.

##### Task 1.1 (1 pt) - Construct RLTK Datasets

Select at least 3 attributes that you think are the most useful for the record linkage task.

Then construct RLTK datasets with the selected fields. You can transform the data if needed. For example, if you have dates “19th Sep 2002”, you can transform it into ‘2002/09/19’ by code.

**Checklist**

* Reference - RLTK Dataset Construction: [[link](https://rltk.readthedocs.io/en/master/step_by_step.html#Construct-RLTK-datasets)]
* Construct RLTK datasets with the selected attributes (at least 3)

##### Task 1.2 (1 pt) - Blocking

Design a blocking technique to reduce the number of pairs to be compared in your record linkage task, and evaluate its effectiveness.

**Checklist**

* Reference 1 - Blocking with “dates”: [[link](https://rltk.readthedocs.io/en/master/step_by_step.html#Blocking)]
* Reference 2 - Blocking with “some tokens” in the field: [[link](https://github.com/usc-isi-i2/rltk/blob/master/examples/blocking/generate_blocks.py)]
* Output the blocked pairs to a csv with following headers: (goodreads.ID, barnes\_and\_nobles.ID) → **Firstname\_Lastname\_blocked.csv**
* Compute *reduction ratio, pairs completeness* and *pair quality with dev.csv*.
  + The *reduction ratio* should be under **0.1** while pair completeness is over **0.7**.
* Explain your design in the **report** (Task 3).
  + Put all the experimental results you have tried (different attributes, different data form of attribute, … etc.) and present which one you used for the final output.

##### Task 1.3 (1 pt) - Entity Linking

Design proper similarity functions to conduct the entity linking.

Tune your similarity functions on *dev.csv* and make final predictions on *test.csv*.

We will grade your predictions by F1 score on *test.csv*. (Full grade - F1 > 0.8)

**Checklist**

* Reference 1 - RLTK Pair-wise comparison: [[link](https://rltk.readthedocs.io/en/master/step_by_step.html#Pairwise-comparison)]
* Reference 2 - String Matching Algorithm: [[link](https://en.wikipedia.org/wiki/Approximate_string_matching)]
* Reference 3 - Demo of string similarity: [[link](http://www.alias-i.com/lingpipe/demos/tutorial/stringCompare/read-me.html)]
* Output the prediction results to a csv with following headers: (goodreads.ID, barnes\_and\_nobles.ID, prediction, confidence) → **Firstname\_Lastname\_predictions.csv**
  + You can just add (prediction, confidence) columns into *test.csv*.
* Explain your design in the **report** (Task 3).
  + Put all the experimental results you have tried (jaro winkler similarity, … etc.) and present which one you used for the final output.

##### Task 1.4 (2 pts) - Record Linkage

Now you have decent blocking technique and similarity functions for entity linking.

Use both from Task 1.2 and Task 1.3 to conduct record linkage.

We will use this further to create a knowledge graph in Task 2.

**Checklist**

* Output the valid pairs (=matching pairs) to a csv with following headers: (goodreads.ID, barnes\_and\_nobles.ID) → **Firstname\_Lastname\_valid\_predictions.csv**

### Task 2: Knowledge Representation - 4 points

Here, you will use data matching pairs from Task 1.4 to construct a knowledge graph.

##### Task 2.1 (3 pts) - Construct KG

First, retrieve all the information of entries in the matching pairs:

* **Keys from Goodreads**: Title, Description, ISBN, ISBN13, Page Count, Authors, Rating, Number of Ratings, Number of Reviews, Publisher, PublishDate, Format, Language.
* **Keys from Barnes and Noble**: Title, Authors, Publisher, ISBN13, Publication Date, Product Dimensions, Sales Rank, Ratings count, Rating value, Paperback Price, Hardcover Price, Nook Book Price, Audiobook Price.

Next, you will use ontology ‘schema.org’ to construct the knowledge graph.

Since the ontology may not include all the classes and properties that you need, you can extend the ontology with classes that you define on your own (e.g., MYNS).

**Checklist**

* Output the model → **Firstname\_Lastname\_model.ttl**
  + Did you define nodes?
  + Did you define URIs and RDF types for the nodes?
  + Did you define properties, values, and data type for the values using Schema?
* Explain your design in the **report**.

##### Task 2.2 (1 pt) - KG visualization

Visualize your model in Task 2.1 using [RDF grapher](https://www.ldf.fi/service/rdf-grapher)

**Checklist**

* Output the model → **Firstname\_Lastname\_viz.png/jpeg**

### Task 3: Report - 1 point

* Task 1.2. Blocking Techniques - Put the results of all the experimental setups you have tried (different attributes, different data form of attribute, … etc.) and present which one you used for the final output.
* Task 1.3. Entity Linking - Put all the experimental results you have tried (jaro winkler similarity, … etc.) and present which one you used for the final output.
* Task 2.1. KG Construction - Explain your KG design.

### Updates

* **False Positives?**
  + There are no “false positives” in *dev.csv*
  + For example, (Goodreads.ID=398, barnes\_and\_nobles.ID=3161) → Checkout ISBN13. They are different.
* **May I confirm if the attribute "title" is the same as "title token"?**
  + token attributes in ER\_KR.ipyng are just examples. You can define and use them if you want. As you said, if we use token-level attributes, we can match the things that are the same but are not exact matches. For example, let's assume that "One Big" and "A One Big" are the same book. Due to the "A", they cannot be matched by exact string match. But, if we use token-level, then we can do some overlap between two lists (e.g., [one, big] <-> [one, big, a]).