Identifying homonyms and affiliation for bibliographic data using web scraping

Mentor : Dr. Suman kundu

Department of Computer Science and Engineering
Indian Institute of Technology, Jodhpur

Akshay Malav (B16CS003)

Chinmay Garg (B16CS041)

1. Introduction:

The population of the world is huge and so is the probability of two people having the same name. Let's look at some statistics: there are 5.1 million people with name John as their first name and there are 47,000+ people with the name John Smith, there are 181,491 people with their first name as Steve and 3591 people with the name Steve Smith, same goes with Michael, there are 241,654 many people with this as their first name and 3191 as Michael Jordan and this list goes on for many more such names and this is a serious issue.

Imagine if a bank credits a person's whole salary to another person having the same name or while assigning grades someone gets the grades of another person. These mistakes are very disastrous as they can lead to huge losses both monetary and non-monetary. So we have tried to solve one such homonym problem in the field of computer science publications.

The consequence of this problem is that, it may be possible that publications of two different people with the same name remains undistinguished. That is the database is treating those two different people as a single author, and now their publications are all mixed up.

This documentation includes step by step process and approach on how we tried to tackle this problem. It also includes the algorithm that we proposed, the references that we took and the libraries that we used.

2. Objective:

The objectives of this project are:

a.) Developing an algorithm which can solve the homonym problem using clusterization.

b.) Finding the affiliation history of a researcher.

We have used dblp dataset for this project which is an online computer science bibliography.

Further objectives for this project are:

c.) Construct a means to create gender dataset for these researchers.
d.) Finding mis-entries in DBLP identified homonyms.

3. Sub-Problems:

1. DBLP Search

i. Approach 1 - NOT Chosen

- Download the dtd file for DBLP database.
- Write a code to parse this file and convert it to a text file.
- Required relations are stored in this text file.
- Now we can search the information regarding any author in it.
- Pros:
    - No internet connection required.
- Cons:
    - High one time calculation cost (during parsing).
    - Updates on original DBLP database would not be reflected.

ii. Approach 2 - Chosen

- Use web scraping to get real-time search results from the DBLP website.
- Referred to paper [1] in references for the way data is represented in xml format on the DBLP website.
- Used the dblp python library :
    - https://github.com/chinmaygarg13/dblp-python/tree/patch-2
- Pros:
    - Gives the latest data.
    - No one time calculation cost.
- Cons:
    - Requires internet.
    - Takes a little more time to load.

2. Author's Affiliation

i. Approach 1 - NOT Chosen

- Download the corresponding research paper for the author.
- For this google scholar can be used, but it does not provide pdf links for all the papers.
- There is one another website: sci-hub, but it is illegal, and requires captcha. This image captcha can be circumvented using OCR.

Approach 1.1

- Each author has its own unique email id.
- Affiliation can be found from email id's domain name.
- For finding out the email, we can search for the character: '@'
- Problems:
    - Not all (actually, quite a large number) research papers have email id's of authors.
    - There is a problem of linking the email id with the corresponding author.
    - Many a times, authors from same institute club their email ids together, for example, [Person_A, Person_B]@uni.edu.

Approach 1.2

- We can search for keywords like 'institute', 'university', etc.
- Problems:
    - Many institutes do not have such keywords in their name.
    - Even if they may have such a keyword, many a times it is in their regional language, e. g., 'Vishwavidyalaya'
    - Not every author belongs to an educational institute, some do their research while working in their organization's lab, e. g., IBM Research Lab.

ii. Approach 2 - NOT Chosen

- After reading research papers [1], [2], [3], and [4], mentioned in readings, we decided to go for Header Extraction from the research papers.
- The tool we chose for this was CERMINE.
- Accuracy that was claimed was 80 to 90%.
- We did not get a single satisfactory result in our initial few tries.
- Thus, this approach was dropped.

iii. Approach 3 - Chosen

- DOI link was extracted from the DBLP database for the corresponding publication.
- Using this DOI link, we can go to the website where this publication is published.
- From this website, using web scraping, we can extract the affiliations for all corresponding authors.
- We have written the scraping code for the following journals/ digital libraries:
    - IEEE
    - Sciencedirect
    - Springer
    - ACM
- The institute has subscriptions to these journals.

- Pros:
    - Error-free mapping of affiliation to author.
- Cons:
    - Requires Internet.
    - Not exhaustive.

3. Publication Subject Categories

i. Approach 1 - NOT Chosen

- From the research paper [3] in references, we learned that we can guess the subject of the paper from the conference or journal it is published in.
- For this we needed a mapping between conferences and their corresponding subjects.
- We found a few lists which classified conferences in different subject categories:
    - https://webdocs.cs.ualberta.ca/~zaiane/htmldocs/ConfRanking.html
    - https://en.wikipedia.org/wiki/List_of_computer_science_conferences
    - http://www.cs.jhu.edu/~taochen/SoC_Conference_Ranking.html
    - https://www.scimagojr.com/journalrank.php?area=1700&category=1702
- We were able to categorize only less than a thousand conferences using the first three lists, and a few thousand using the last web link.
- But, DBLP has a total of 11940 conferences and 4612 journals.
- The number we were able to categorize was nowhere near it.

ii. Approach 2 - NOT Chosen

- Make an exhaustive list of all the Computer Science subjects.
- Download the corresponding PDF of the publication.
- Apply distant learning on it.
- Match if any of the most recurring words belong to our prepared list.
- If yes, then it is the subject of the paper.
- Cons:
    - No legal way of downloading all the research papers.
    - No guarantee of getting correct results all the time.
    - No guarantee that the list is exhaustive.
    - Author may not use the subject name recurringly.

iii. Approach 3 - NOT Chosen

- Many authors write a list of keywords in the research paper.
- Subject of the publication can be predicted from those keywords.

- Because a large number of authors do not write such keywords, it is not a reliable method.

iv. Approach 4 - Chosen

- When we search the name of the conference or the journal on the Microsoft Academic website (https://academic.microsoft.com/), we can get a list of subject topics associated with the search item.
- We can scrap this list of subject names, from the webpage.
- But here's the trick, Microsoft Academic does not allow web scraping.
- So, first we take the screenshot of the search result.
- Then we crop this image, so that now we only have the subject section.
- We apply OCR to this image, to extract the subject tags.
- By far, this is the most exhaustive approach, but also the most time taking.

4. Homonyms

- We read multiple papers, [5] to [11] in readings.
- All these papers were using machine learning, co-authorship networks, and bibliographic relationships for the classification.
- Then we came across paper [2] mentioned in the references.
- This paper also uses machine learning, but it also mentions a scoring schema.
- We decided to use rule-based inferencing using a schema similar to what mentioned above.

5. Gender Dataset

- We will be making a website where all the above information can be accessed.
- On this website, we will also be predicting the gender of the author using:
    - Existing python libraries.
    - Face recognition APIs, by first searching for the author online.
- The visitors of the website can claim, if the predicted gender is correct or incorrect.
- This way by using crowdsourcing, we can create a gender dataset for corresponding computer science authors.

4. Algorithm:

1. Scoring Schema:

Category	Criterion	Score
Common Co-Authors (other than the author in question)	1 (both papers have less than 4 authors each)	8
	1 (any of the paper has more than 4 authors)	4
	2 or more	4 per common co-author
Affiliation (of the author in question)	matched	10
Journal/Conference	Exact match	6
	1 or more corresponding subjects match	3
Subject	1	2
	2	4
	3 or more	5
Self-Citation		10
Bibliographic coupling	1	2
	2	4
	3 or more	5
Threshold		11

2. Damerau - Levenshtein Distance

Consider the following three cases:

- The author's name that is given on the DBLP might not be in the exact same manner as on the website from which we are scraping the author - affiliation mapping.
- Affiliation of the author in two different papers might be represented in different manner, e. g., IIT Jodhpur, Indian Institute of Technology Jodhpur, etc.
- Author may use different ways of writing citations/references in different papers.

To tackle such kind of conditions we will need to use some algorithm to find the string similarity.
Thus, we are using Damerau-Levenshtein Distance algorithm.
It is the minimum number of operations (consisting of insertions, deletions or substitutions of a single character, or transposition of two adjacent characters) required to change one string into the other.

5. Observations:

- A number of the test runs were giving exactly accurate results.
- Most of the test runs were giving slightly accurate test results. This was due to the fact that not all research papers are published on the online journals that we selected. Thus, we were unable to extract affiliations and references of those papers.
- We were able to identify wrong entries.
- No mis-clustering was observed, where two different authors were clustered together.
- In few of the cases, there was not adequate data even on the DBLP.
- The whole process is very time taking. Corresponding subjects of all the conferences and journals can be pre stored in a file.

6. Conclusion:

The method that we used for solving the homonym problem gave us satisfactory results on most of the use cases that we tested on.
We were also able to find the affiliation history of a researcher provided his/her publications are from the four publishers: IEEE, Sciencedirect, LinkSpringer, ACM.

The work that is remaining is including more publication websites, from which the affiliation will be extracted, so that affiliation history contains more number of years than before, and also the clustering is more accurate.

Unexpectedly, we were also able to identify misenteries in the DBLP identified homonyms, that is, for example, a publication by Aihua Wu 0003 was under the name of Aihua Wu 0002, but even then it was clusterized with the publications of Aihua Wu 0003 and not with Aihua Wu 0002.

Also for prediction of gender of a researcher we have made a basic website. Few of the functionalites are remaining to be implemented.

7. References:

1. Michael Ley: Appendix to the paper “DBLP - Some Lessons Learned” (June 17, 2019).
2. E. Caron, N. J. van Eck, "Large scale author name disambiguation using rule-based scoring and clustering".
3. Laurel Orr and Jennifer Ortiz: Clustering with the DBLP Bibliography to Measure External Impact of a Computer Science Research Area.

8. Readings:

1. Ozair Saleem, Seemab Latif: Information Extraction from Research Papers by Data Integration and Data Validation from Multiple Header Extraction Sources.
2. Mario Lipinski, Kevin Yao, Corinna Breitinger, Joeran Beel, Bela Gipp: Evaluation of Header Metadata Extraction Approaches and Tools for Scientific PDF Documents.
3. Dominica Tkaczyk, Pawel Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, Lukasz Bolikowski: CERMINE: automatic extraction of structured metadata from scientific literature.
4. Zhixin Guo, Hai Jin: Reference metadata extraction from Scientific Papers.
5. Marcel R. Ackermann, Florian Reitz: Homonym Detection in Curated Bibliographies: Learning from dblp’s Experience.
6. Fakhri Momeni, Philipp Mayr: Using co-authorship networks for author name disambiguation.
7. E. Amigó, J. Gonzalo, J. Artiles, F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints".
8. A. A. Ferreira, M. A. Gonçalves, A. H. F. Laender, "A brief survey of automatic methods for author name disambiguation".
9. T. Gurney, E. Horlings, P. V. den Besselaar, "Author disambiguation using multi-aspect similarity indicators".
10. P. Mayr, F. Momeni, "An open testbed for author name disambiguation evaluation".
11. H. T. Nguyen, T. H. Cao, "Named entity disambiguation: A hybrid statistical and rule-based incremental approach".