DSCI 550
Professor Muric
Professor Burghardt
3/6/22

Assignment #1: Analysis of Media and Semantic Forensics in Scientific Literature

By: Sarah Hsuan Chu, Matthew Fishman, Audrey Lin,
Elena Pilch, Alex DongHyeon Seo, Andy Xiang

Assignment Overview & Observations:

In preparation for the scraping portion of the assignment, we began by preprocessing the Bik et al. dataset, which we renamed Bik.tsv for convenience, as it was necessary to streamline the automated processes required to extract additional features and join additional datasets. This involved loading the data into a Pandas dataframe, removing the 800 empty rows from the end of the dataset, reformatting incorrect character encodings, standardizing data types within columns, which Tika is highly sensitive to, replacing incorrect values that could be inferred, and removing those that could not. In particular, special characters in DOI numbers were replaced, years were converted from float types to integer types, months were standardized numerically as string types to preserve their chronological meaning, boolean values were converted from float types to string types, and NaN float types were converted to empty strings for the appropriate features.

Once the dataset was cleaned, we were able to begin extracting features for Part 4. For this assignment, we relied heavily on Research Gate to scrape data about each publication's authors, as other websites for less established researchers did not have profiles or additional information. Consequently, we utilized Research Gate in order to extract the majority of the features. As an established database, Research Gate author profiles reliably showed up in the top Google Search results for all authors. Furthermore, Research Gate even compiles profiles for authors who are not officially registered with them, which is distinctive from otherwise similarly robust databases like Google Scholar. Both of these aspects make Research Gate a reliable source to scrape data from for feature extraction and make it conducive to automation with Selenium and Beautiful Soup. We attempted to scrape Web of Science, which had very specific information about authors, but commercial access was not permitted. Whilst multiple APIs were also considered for feature extraction, the required features were not available, and limitations to API calls presented additional setbacks. In addition, freely available APIs only offered basic information, most of which were already provided in the original Bik et al. dataset, and ones that provided more information required payment.

All of the features in Part 4, excluding lab size, were scraped using the extracted URLs using BeautifulSoup and Selenium. To obtain the Research Gate author profile URLs, we created a function using the Google Search library to automatically query the author's name plus "research gate" and extract the URL of the first Google Search result. Then Selenium and Beautiful Soup were used to scrape those pages. Research Gate's website structure is consistent overall, but there are slight differences between registered Research Gate author profiles and author profiles automatically compiled by Research Gate for unregistered authors. This was accounted for by identifying the distinct tags locating the target text for each type of profile. Then the extracted features were saved to lists that could be added directly to the dataframe as new columns.

We also considered extracting lab size from Research Gate by using Selenium to browse Research Gate author profiles and click on the link to their affiliated university, and Beautiful Soup to extract the number of members from the appropriate department if possible, or the number of members of the entire university otherwise. However, the difference between the number of members in a department versus an entire university is huge and would skew our Tika Similarity analysis. Furthermore, the membership count provided by Research Gate is cumulative, so we felt totaling the authors in a paper would be a closer representation of an author's lab size.

One of the major challenges when scraping Research Gate was the efficiency of our scraping script. Since we had scraped the site from the same IP address with such frequency in a short span of time we were flagged as a bot and continued to get blocked, at which point Research Gate would force us to click a checkbox or images to confirm that we were not robots. As a result, Selenium and Beautiful Soup would scrape HTML pages with empty body content, and we could do nothing but wait it out and try again. To circumvent this, we added a 60 second delay in our loops with time.sleep(60), and we also practiced some makeshift distributed computing by splitting the dataset amongst ourselves, running the script on our individual laptops, and combining our results. Even after implementing distributed computing it was an extremely time consuming process, and we still got blocked eventually but managed to recover a few more instances. We also encountered issues with Cloudflare.

Automated processes can render imperfect results like these, but it was critical for us to have as many values for Affiliated Universities as possible, as that is where we would join our additional datasets for Part 5. To supplement the missing values, we scraped and accessed the journal pages, deriving the URLs from the DOI and PMID numbers provided in the original dataset. This feature was especially important to be as thorough as possible in order to merge more of our datasets. This was a laborious process that required the exploration of fifteen different home websites with eight unique website structures. Another challenge with this methodology was that the affiliation from these journal pages were entire addresses rather than just the name of the university itself. Because this was a supplemental process, we decided it would be sufficient to extract phrases that contain “Univers,” as many affiliated institutions are universities, and the word for “university” in a variety of languages begins with “univers.” Our initial extraction for Affiliated Universities yielded 90 values – to the fault of getting blocked from web scraping rather than the code – but with our supplemental extraction yielded 145. Data is lost as it filters through the data pipeline; consequently, it is imperative to the continued quality of our dataset that we begin with as complete and robust a dataset as possible.

In order to transition to part 5 of the assignment, we began by sourcing appropriate datasets. We imported the CSV file of the web-scraped features, created in part 4, in order to create a new dataframe. We had two additional CSV files: the first file represents university rankings and the second includes university demographics. We made certain to select data from multiple MIME types including CSV (application MIME type) and TXT files (text MIME type). The text file had to be cleaned and then reformatted to a dataframe so that we can use the text file in a practical format for this assignment. We then cleaned the university names by removing hyphens and spaces and used the data from the year of 2015. The third source is a text file that includes information based on each county and their subsequent unemployment and labor data. The following uscities.csv file that we imported provided the city and county names, a viable joiner for the various datasets we had. We wrote a new scraping function to find the city for each university name. For each university name, we searched for their mailing address and extracted the city. Based on the city, we were able to join by county, which allowed us to join countydata.txt (labor data for each county) We also cleaned the county data by removing spaces and shifting all characters to lowercase so that we could do a left join on the data. This allowed us to join Bik_pt4.tsv, with university ranking, university demographics, county labor data, city (queried with the Google Search library), and the city data (sourced from uscities.csv). After joining the datasets, we looked through the amalgamated dataset and dropped the columns that we found to be irrelevant or duplicates. Since there were only around 145 universities extracted in Part 4, we were limited in our ability to join universities by name. Consequently, this caused a limited number of counties to be found through an automated search of the universities’ mailing addresses. In Python, a left join would have to be an exact match by university name, county, or city. If one letter is capitalized or missing, then Python would not recognize this match. The results of web scraping and finding an exact match with unstructured data created numerous missing rows in the added features.

Additional features include: 'university_name', 'year', 'world_rank_x', 'country_x', 'national_rank', 'quality_of_education', 'alumni_employment', 'quality_of_faculty', 'publications', 'influence', 'citations_x', 'broad_impact', 'patents', 'score', 'year_x', 'world_rank_y', 'country_y', 'teaching', 'international', 'research', 'citations_y', 'income', 'total_score', 'num_students', 'student_staff_ratio', 'international_students', 'female_male_ratio', 'year_y', 'university_name', 'cities_from_extract', 'city', 'city_ascii', 'state_id', 'state_name', 'county_fips', 'county_name', 'lat', 'lng', 'population', 'density', 'source', 'military', 'incorporated', 'timezone', 'ranking', 'zips', 'id', 'county_name_test', 'county', 'labor_force', 'employed', 'unemployed', 'rate'.

Evaluation:

- **What questions did your new joined datasets allow you to answer about the Bik et al papers previously unanswered?**

The questions generated from our new joined dataset provided several answers regarding the Bik et al. papers that were previously unanswered. One of the new questions we can answer based on the economic status of the area is whether a lower unemployment rate leads to a better university ranking. Another question we can answer based on the demographic and population data is whether more international students in a university affect the economic status, unemployment rate, income, university ranking, and the quality of the faculty.

We can also test and reveal correlation strengths between university rankings with either citations, rankings, faculty, or teaching scores, along with economic or demographic features including unemployment rate or population size. For instance, we can observe whether a densely populated city is more likely to have better research, citation, or income. Given the numerous additional features added, such as world_rank, university_score, density, labor_force, etc, we can continue answering various questions related to universities' performance given their economic status or demographic background.

By merging the university rankings, university demographics, labor data by county, and city census data, we are able to provide in-depth insights, primarily for U.S. universities, regarding their economic, academic, and demographic information.

- **What clusters were revealed?**

We used the Apache Tika-Similarity package to calculate the similarity measures and produced CSV files of row-by-row similarity. Since this is a combination dataset for each paper, we made a heatmap to visualize the similarities that were measured. We did not further cluster using the Apache Tika-Similarity package but we could observe outstanding patterns from the similarities heatmap visualization that we produced. For example, Jaro-Winkler distance and Levenshtein distance calculate similarities in a related manner, as demonstrated between the datasets, displayed by the patterns of the heatmap. Interestingly, Gaussian overlap which uses hashed float data and Levenshtein & Jaro-Winkler metrics, which uses string data, showed similar patterns from the heatmap. With further investigation, these patterns could be clustered based on the similarity data we calculated.

- **What similarity metrics produced more (in your opinion) accurate measurements? Why?**

We think accurate measurement of similarity between the data depends on the data. Therefore, we picked four different similarity metrics including Cosine Similarity, Levenshtein Similarity, Jaro-Winkler Distance, and Gaussian Overlap to compute the similarity measurements. Edit-distances (Jaro-Winkler and Levenshtein) take string data into account which we accomplished by changing the whole row into one string data. Cosine Similarity & Gaussian Overlap take a list of float data to calculate the similarity which we did by hashing the string data and adding up all the numbers in the list. These methods of preprocessing came from the Tika Similarity packages. By using two different methods of preprocessing data, we were able to tell which similarities were more suitable for our dataset and produce accurate measurements. Edit-distances such as Jaro-Winkler and Levenshtein reveal observable clusters (rows with high similarities between all rows) in the heatmap visualizations, but they require changing all of our features to string types, which is not practical and might omit integral measurements. Therefore, we suppose that for the Bik et al. dataset, with the features that we have found, Cosine Similarity and Gaussian Overlap are more suitable. Between these two similarities, we suppose Cosine Similarity produces a more accurate measurement in our case, because it shows consistency in similarity and also takes a variety of data types and empty data in the features into account.

- **What did the additional datasets suggest about "unintended consequences" related to media forensics data?**

Although it is difficult to determine the specific relationships between features from the visualization of the heat maps, rather, we can make larger commentary on the success between the various similarity metrics that we ran, and make some general assumptions about the "unintended consequences."

We can also investigate unintended consequences of university rankings. For instance, one might assume that higher ranked universities would produce higher quality papers. However, by looking at our data on university world rankings, we observe several top 100 universities have produced papers with potentially problematic media manipulations compared to lower ranked universities. Perhaps some papers go through less rigorous review if they come from higher ranked universities, so some papers may slip through the cracks. However, this observation can also be attributed to other factors, such as that higher ranked universities produce more papers or are better documented. Regardless, it is unexpected to observe such papers produced by so many top universities.

Through the additional datasets found and merged, we can further investigate unintended consequences such as if economic or demographic factors: international student population, female ratio, or city density, can have a positive or negative impact on university performance. We found numerous metrics to evaluate university performance such as ranking, quality of faculty, research, publications, etc. This begs such questions: “Is a lower unemployment rate correlated with university ranking?” or if there are serious economic disparities that can affect the quality of the university. Are socioeconomic status and number of publications per university unrelated?

Thinking more broadly, do you have enough information to answer the following (You don’t need to limit yourself with this list of questions. You can extend if you find it appropriate.):

1. Are there clusters of authors with similar media manipulations that exhibit at least 3 similarities between the students and properties of their lab?

   - Our data does not have relevant features about lab properties.

2. Does staying up late at night matter?

   - Our data does not have relevant features to inform us whether authors stay up late or not.

3. Does the animal population for available specimens in the area influence the type of manipulations?

   - Our data does not have relevant features to inform us about the animal population for available specimens in the area.

4. What insights do the “demographic” features of the authors tell us about the data?

   - Our data does not have the demographic features of the authors. We have data on demographic features of the university and university’s city.

5. What do animal population demographics tell us about the institutions in which media manipulations occur?

   a. Densely populated? Sparsely populated?

      - Our data does not have relevant features about the animal population demographics.

6. What insights do the “indirect” features you extracted tell us about the data?

   - When observing the degree areas of first authors, we noticed media manipulation occurring across many scientific fields. It isn’t limited to one area.
   - Out of all the universities that we were able to identify for the first authors, Chung Shan Medical University and Louisiana State University were responsible for three known

publications that were flagged. Eight other universities published two known publications that were flagged. Most universities were featured once, which may indicate that universities attempt to maintain their reputation and prevent papers using media manipulation to be published after one instance. Since these results only correspond to first authors and there were 69 universities unidentified, it might not be significant.

- The highest number of flagged papers occurred in July (49) and April (40). Deadlines for grants, fellowships, and promotions frequently occur at the end of the school year; thus, researchers might be turning publications in before they are perfected in order to qualify for these distinctions.

Also include your thoughts about Apache Tika – what was easy about using it? What wasn't?

We encountered several issues when utilizing the Apache Tika Similarity. Although downloading Apache Tika was successful, running the commands for each python file within the tika-similarity package was troublesome. First, the package's functions are not generalized to various datasets. Consequently, we had to pre-process and clean each of our datasets in order to adapt to the needs and format that the packages' functions work with, in addition to changing the source code itself. Second, we had to manually edit portions of code for the cosine-similarity.py, feature.py etc., and edit nearly 50% of the vector.py file in order to successfully run the commands. In general, we thought the Tika Similarity package was poorly written regarding the python code and documentation. The documentation was also not informative. The README.md guidelines were unclear. The files are not up to date and do not run. Moreover, there are a multitude of syntax errors, misspelled variables, and functions.

Overall, we believe that if Tika had been regularly updated and maintained, the package could be incredibly useful. However, it is poorly maintained, and as a result, is easily replaced with other tools. The concept itself has potential with its D3 visualization tools, but only if the code were rewritten in a clear and readable manner.