01

evaluating matches between data sets in R

alexis dhembe, lesley university library

Code4Lib 2021 Poster Session

→

02

the problem

At the Lesley library, we have several projects and recurring tasks that require comparing data between our ILS and other systems, such as student data from the bursar's office, or holdings data from our knowledge base. Because the two information sources were not always perfectly matched, previous workflows involved a lot of manual review in order to account for discrepancies between the two sources.

→

the goal

→

I wanted to find a more efficient way to compare data sets, but I knew that we would have outliers that required some individual review. My goal was to find a way to compare the majority of records computationally, so staff only had to manually review records that actually warranted it.

Not all of the data sets had matching key identifiers I could use, so I also needed to account for a little fuzziness.

two use cases

01

For Patron Data

Each semester, we get a list of students who intend to graduate at the next conferral date. We wanted a better way to compare that to our patron data from the ILS to determine which students still had items checked out. We could match these accounts based on an institutional ID number, but I wanted to confirm that we didn't assign an ID number to the wrong patron at registration, or use an in-use ID as a "generic" stand-in. To typo is to be human, after all.

02

For Holdings Data

We had a mystery package in our knowledge base that looked like it was meant to represent our print journal holdings, but details on the titles (such as ISSNs) were sparse. I wanted to compare the package to our current ILS holdings, but I needed to differentiate when something wasn't an exact match because of the difference between "Journal of..." and "Journal for..." or the difference between "i-D" and "Ms."

enter: R's *stringdist* package

I've been using R (especially the *tidyverse*) to do a lot of the data wrangling that results from having information spread across several different systems, each with their own quirks. I knew I wanted to be able to compare inexact matches, and I wanted to be able to 'rank' those matches, so I could extract the more questionable ones for review.

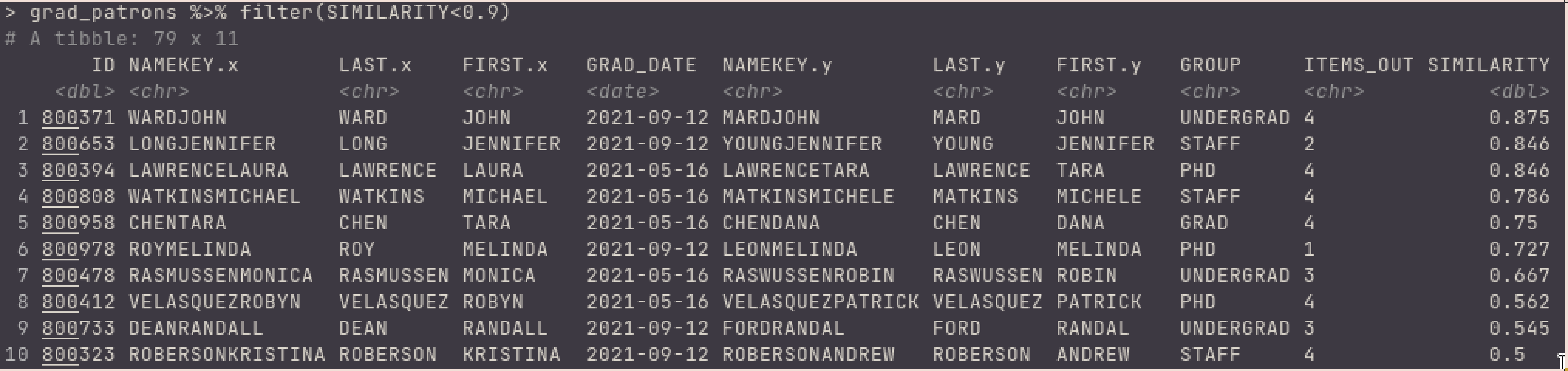
→

R's *stringdist* package ended up being a perfect fit for my needs. *stringdist* provides fuzzy string matching capabilities based on various string distance measures. The basic *stringdist* function calculates the distance between two strings, e.g. the difference between "ABC" and "ACAB."

stringsim takes this distance, divides it by the maximum possible distance, and subtracts it from 1. This produces a value where 1 is perfect similarity, and 0 is completely dissimilar--precisely the kind of 'score' I could use to evaluate and rank matches.

evaluating student data

I wanted to review records where the ID# matched between the bursar's records and ours, but the name differed. I created an additional key by combining last and first name into a single string. I joined the two tables on ID#, then used *stringsim* to create a new column comparing the name-key from the bursar records with ours. In order to err on the side of caution, given that this was a new workflow, everything with a similarity below 0.9 was flagged for additional review by library staff.



```
> grad_patrons %>% filter(SIMILARITY<0.9)
# A tibble: 79 x 11
       ID NAMEKEY.x        LAST.x    FIRST.x   GRAD_DATE  NAMEKEY.y        LAST.y    FIRST.y   GROUP     ITEMS_OUT SIMILARITY
    <dbl> <chr>            <chr>     <chr>     <date>     <chr>            <chr>     <chr>     <chr>     <chr>          <dbl>
 1 800371 WARDJOHN         WARD      JOHN      2021-09-12 MARDJOHN         MARD      JOHN      UNDERGRAD 4              0.875
 2 800653 LONGJENNIFER     LONG      JENNIFER  2021-09-12 YOUNGJENNIFER    YOUNG     JENNIFER  STAFF     2              0.846
 3 800394 LAWRENCELAURA    LAWRENCE  LAURA     2021-05-16 LAWRENCETARA     LAWRENCE  TARA      PHD       4              0.846
 4 800808 WATKINSMICHAEL   WATKINS   MICHAEL   2021-05-16 MATKINSMICHELE   MATKINS   MICHELE   STAFF     4              0.786
 5 800958 CHENTARA         CHEN      TARA      2021-05-16 CHENDANA         CHEN      DANA      GRAD      4              0.75 
 6 800978 ROYMELINDA       ROY       MELINDA   2021-09-12 LEONMELINDA      LEON      MELINDA   PHD       1              0.727
 7 800478 RASMUSSENMONICA  RASMUSSEN MONICA    2021-05-16 RASWUSSENROBIN   RASWUSSEN ROBIN     UNDERGRAD 3              0.667
 8 800412 VELASQUEZROBYN   VELASQUEZ ROBYN     2021-05-16 VELASQUEZPATRICK VELASQUEZ PATRICK   PHD       4              0.562
 9 800733 DEANRANDALL      DEAN      RANDALL   2021-09-12 FORDRANDAL       FORD      RANDAL    UNDERGRAD 3              0.545
10 800323 ROBERSONKRISTINA ROBERSON  KRISTINA  2021-09-12 ROBERSONANDREW   ROBERSON  ANDREW    STAFF     4              0.5  
```

Note: *this example uses 100% fake data created with Python's Faker package, not our real patron data!*

→

student data (cont'd.)

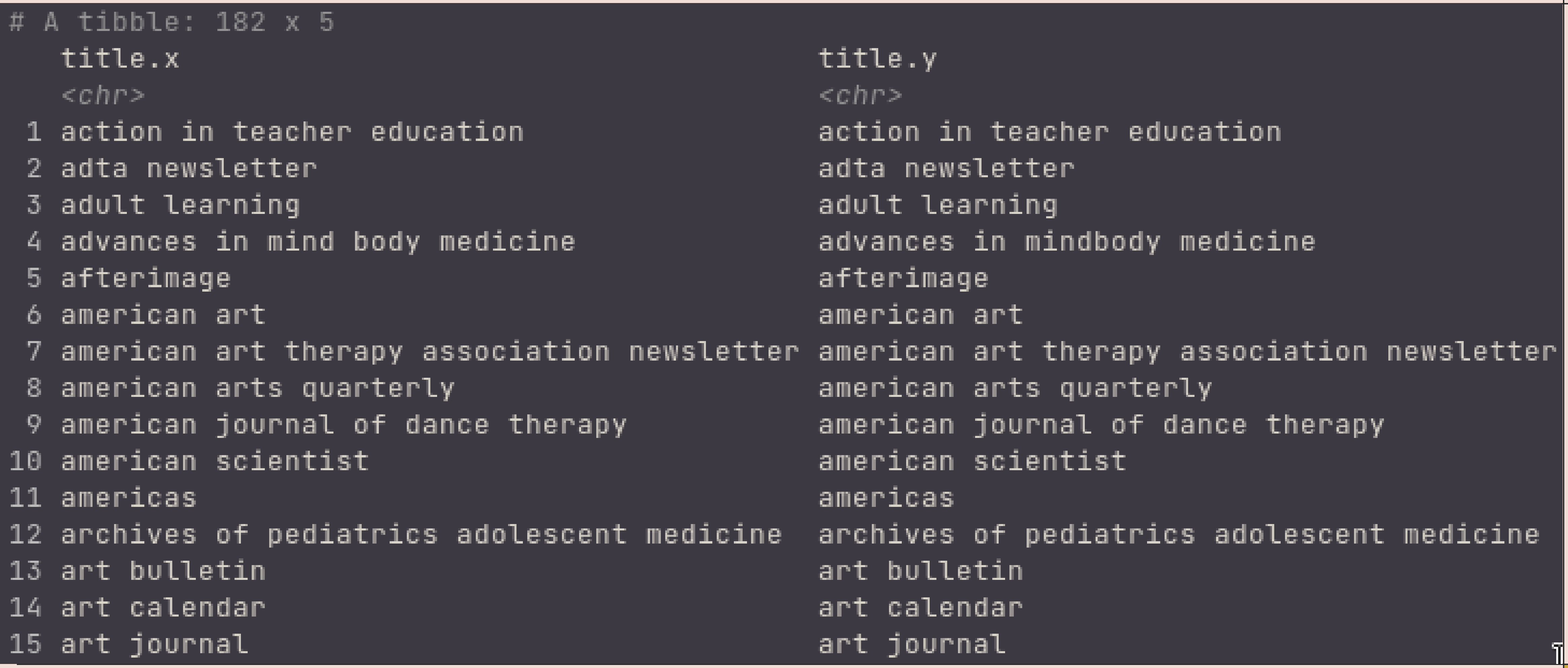
Our previous workflow for graduating patrons involved having staff and student workers look up graduates by name and review their accounts. By adding the additional step of evaluating account names, I was able to create a process that was much faster, but still had a safety mechanism for discrepancies (which was also reassuring to our stakeholders). We're able to identify ~93% of graduating patrons automatically, saving significant staff time.

comparing holdings data

→

This method of evaluating string matches has also been useful for analyzing holdings data, particularly when one set is missing information or incomplete. In this particular case, we had a package in our knowledge base that seemed meant to represent our print holdings, for inclusion on our journals list. I wanted to compare it to our current print holdings (retrieved from our ILS) to see if it was still accurate and update coverage information. I first tried comparing titles by ISSN, and found matches in print holdings for 204 of the 469 titles from the Knowledge Base Mystery Package. This left 265 mystery titles remaining, which far exceeded my personal pain threshold for dragging spreadsheet cells around.

holdings data (cont'd.)



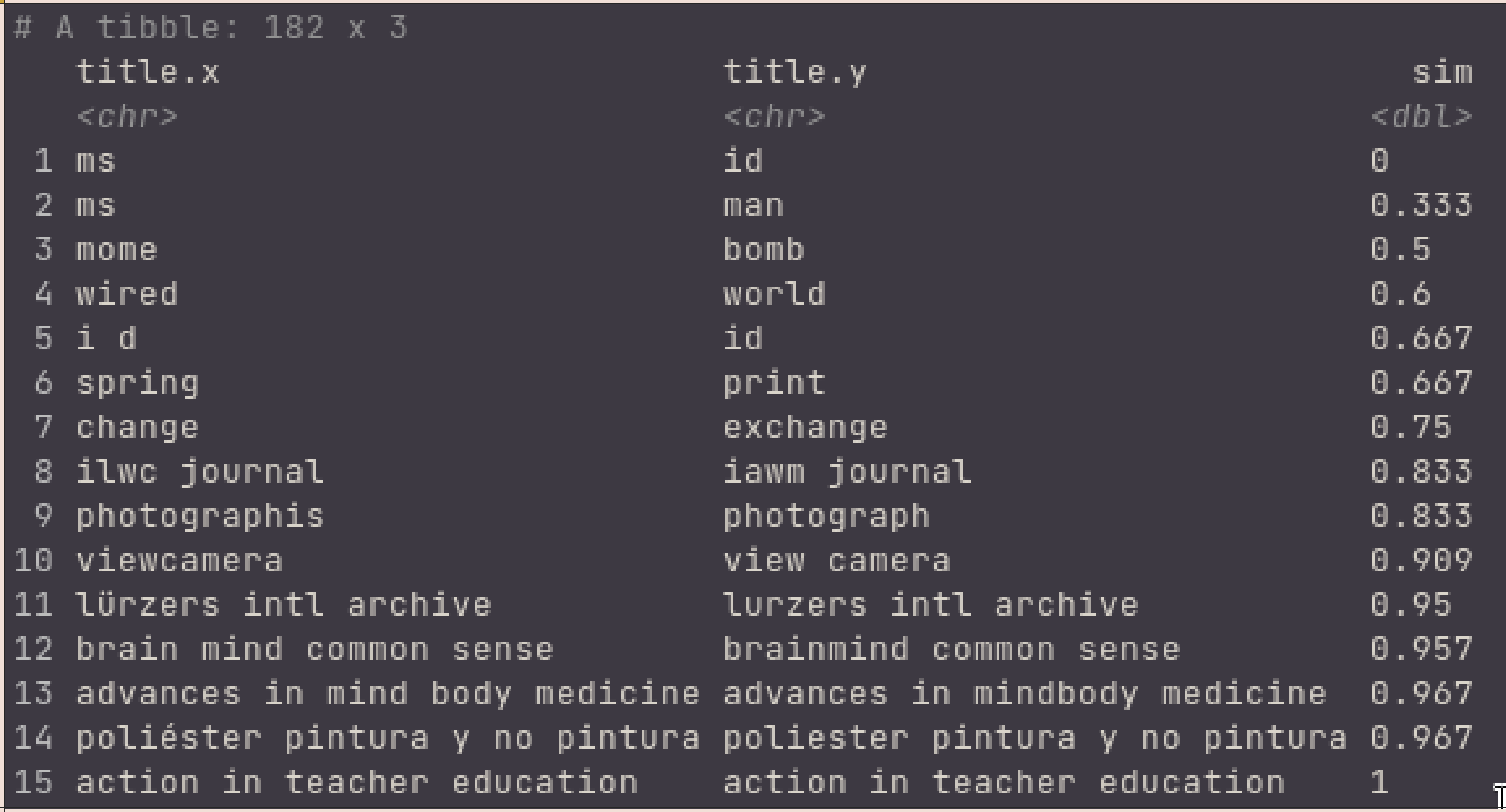
```
# A tibble: 182 x 5
   title.x                                     title.y
   <chr>                                       <chr>
 1 action in teacher education                 action in teacher education
 2 adta newsletter                             adta newsletter
 3 adult learning                              adult learning
 4 advances in mind body medicine              advances in mindbody medicine
 5 afterimage                                  afterimage
 6 american art                                american art
 7 american art therapy association newsletter american art therapy association newsletter
 8 american arts quarterly                     american arts quarterly
 9 american journal of dance therapy           american journal of dance therapy
10 american scientist                          american scientist
11 americas                                    americas
12 archives of pediatrics adolescent medicine  archives of pediatrics adolescent medicine
13 art bulletin                                art bulletin
14 art calendar                                art calendar
15 art journal                                 art journal
```

I decided to try fuzzy matching based on title, as I had little other data from the KB package to work with. I knew the *fuzzyjoin* package would be my best bet, using *stringdist_inner_join* to join the KB package to our catalog holdings on title. I cleaned extra punctuation and spaces and converted to lowercase to ease things along. This was fairly successful at first glance.

So far, so good, right? (Unlike the patron data shown earlier, this is our real holdings data; no use faking what's in the OPAC already)

holdings data (cont'd.)

I knew we had some very short titles, like the previously mentioned "i-D" and "Ms," where two edits could mean the difference between two totally different titles. I used *stringsim* to compare the matched titles from the KB and the print catalog, which allowed me to filter out those more tenuous matches. This isn't a huge dataset, but I've been able to apply this to other collection-management projects already.



```
# A tibble: 182 x 3
   title.x                        title.y                        sim
   <chr>                          <chr>                        <dbl>
 1 ms                             id                           0    
 2 ms                             man                          0.333
 3 mome                           bomb                         0.5  
 4 wired                          world                        0.6  
 5 i d                            id                           0.667
 6 spring                         print                        0.667
 7 change                         exchange                     0.75 
 8 ilwc journal                   iawm journal                 0.833
 9 photographis                   photograph                   0.833
10 viewcamera                     view camera                  0.909
11 lürzers intl archive           lurzers intl archive         0.95 
12 brain mind common sense        brainmind common sense       0.957
13 advances in mind body medicine advances in mindbody medicine 0.967
14 poliéster pintura y no pintura poliester pintura y no pintura 0.967
15 action in teacher education    action in teacher education    1    
```

fin!

That's about it! This is a pretty simple presentation on a pretty simple function, but it's helped me tackle a lot of long-standing data management projects by filtering out what actually needs a more thorough review. It's also been helpful in demonstrating to library stakeholders that we can strike a balance between computational matching and manual review, which has saved a lot of staff-time and tedious clicking. It's been very useful for me, so I hope this can be useful to someone else as well!

let's talk!

github: https://github.com/alxsdhm/c4l2021_poster
tweets: @alxsdhm
discord: alxs#4443

→