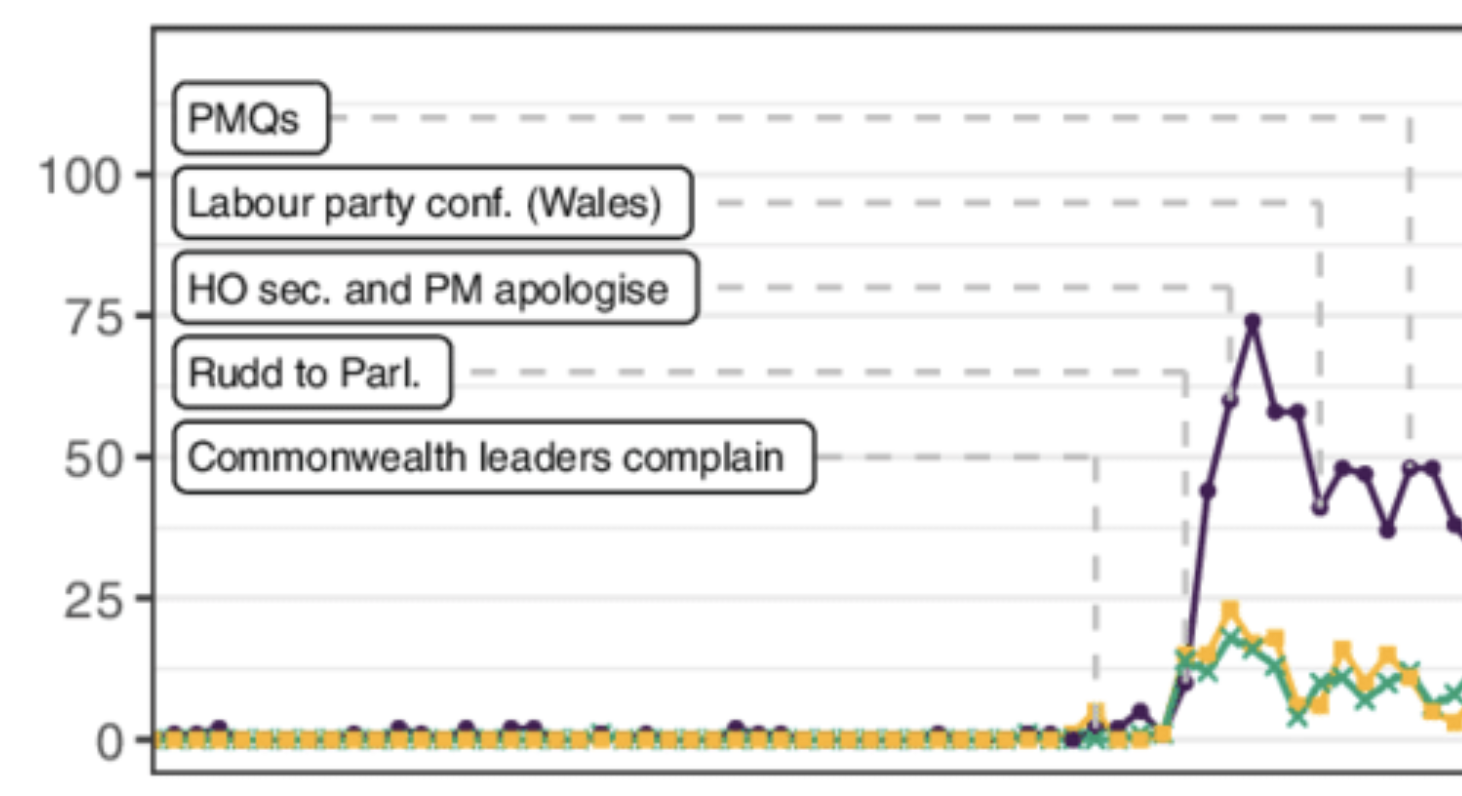
**Matching Algorithm [Google Scholar to Twitter]**

**Performance**

| **Metrics** | **Original [**[AIScholarTwitter](https://drive.google.com/drive/folders/1nLFIem2oZajygSItOWYeJFLhJKrN2Sa3)**] [**[Code](https://colab.research.google.com/drive/1HrblNF8zLp7jTnW679GeerGRBQ1Wdd_u?authuser=1#scrollTo=hpE6y3PB8Yxr&uniqifier=1)**]** | **V1-all branches  [**black text below**] [**[Code](https://colab.research.google.com/drive/1hyPc9u8C9DA3WpLwOxTrJkVmuA5sYbLu?authuser=1#scrollTo=P-rWb6EJh94R)**]** | **V2-all changes [**black+red+green**] [**[Code](https://colab.research.google.com/drive/1wu5F9jW3mcQTm8YKxQudd572fcANJQh3?authuser=1#scrollTo=PEWY6iMCOrxf)**]** |
| --- | --- | --- | --- |
| Total rows [i] | 501 | 501 | 501 |
| Twitter ID in top 10 [ii] | 102 | 129 | 119 |
| Twitter ID in top 1 [iii] | 93 | 86 | 101 |
| Twitter ID with no candidates [iv] | 6 | 5 | 28 |
| Twitter ID not in candidates [v] | 60 | 34 | 21 |
| Null ID with no candidates [vi] | 51 | 81 | 150 |
| Null ID with candidates [vii] | 282 | 252 | 183 |
| **P@10  [** ii / ii + v + vii **]** | 0.2297 | 0.3108 | **0.3684** |
| **R@10  [** ii / ii + iv + v **]** | 0.6071 | **0.7678** | 0.7083 |
| **F@10  [**2\*P@10\*R@10 / P@10 + R@10**]** | 0.3333 | 0.4425 | **0.4847** |
| **P@1 [** iii / ii + v + vi **]** | 0.2094 | 0.2072 | **0.3126** |
| **R@1** | 0.5536 | 0.5119 | **0.6011** |
| **F@1** | 0.3039 | 0.2950 | **0.4114** |
| **Correctly predicted twitter IDs (top 10) out of all scholars having Twitter account  [**ii / ii + iv + v**]** | 0.6071 | **0.7678** | 0.7083 |
| **Correctly predicted twitter IDs (top 1) out of all scholars having Twitter account  [**ii / ii + iv + v**]** | 0.5535 | 0.5119 | **0.6012** |
| **Twitter IDs missed out of all scholars having Twitter account** | 0.3928 | **0.2321** | 0.2917 |

**TODO 20230316**

* Plot a trace how “recall@all” decrease during the logic of search\_scholar\_twitter()



* Annotate top 100 cited scholars’ twitter account
* Check # tweets

|  | Top\_100\_cited scholars | Random 500 scholars | Random 10K (?) scholars | All 78K scholars |
| --- | --- | --- | --- | --- |
| # perfectly correct twitter accounts |  |  |  |  |
| # tweets |  |  |  |  |

* Plan A: high F1 twitter matcher
* Plan B (where high recall is needed): Play around to check whether **MTurk** can really handle this task

See whether MTurk is better or worse than our algorithm

* Plan C (where high recall is needed): Friends that you think can handle this task. (Budget: 500 USD)
* Plan D: only our annotated 100+500

Past tasks

* Add recall@10, recall@1, F1@1
* [Try by this weekend] Add 5 different versions of the algorithms
* Backup option if we don’t get decent F1
  + take highest P@10, also balancing the recall
  + Annotate random 1k-2k AND top 100-200 cited scholars

**Algorithm**

* Search name through Google Scholar list and get name (exclude words of length one from the name eg *Benjamin J Ainscough -> Benjamin Ainscough*), organisation (may have to use NER), homepage [present in ‘homepage’, or extracted from ‘email’], email [extracted from email\_info, or from ‘email’], etc.
* If no info is found
  + Google Search for *name* “twitter” and rank by frequency of appearing in search
  + Twitter Search users with *name*, with every id having equal rank of 1
  + Combine above two lists by taking max of rank for each user
  + Rank all twitter IDs based on latest few tweets having AI/ML/conference keywords
  + Take highest ranked users to be candidates and rank using string similarity with <FirstNameLastName> (threshold >=0.20)
* If info is found,
  + Search Twitter URL from homepage;
    - Ignore IDs with length <2 and ‘githubstatus’ (this appears when homepage URL is broken)
    - If multiple URLs are found, rank based on string similarity with <FirstNameLastName> (threshold >=0.20)
    - If twitter ID is found, return highest ranked ID [this method should give 100% correctness]
  + Run Google Search for the following and rank each by frequency of appearing in search (all 4 lists are combined by taking max of rank of each) | Sometimes correct result does not have more frequency
    - *name* *email\_info* “twitter”
    - *name* *organisation* “twitter”
      * [?] Latest Organisation may have been changed - get latest org name from LinkedIn and search using that instead
      * [?] Use Spacy NER to pass Organisation name (sometimes org contains designation which hurts search results); if no entities detected, use full text of org
    - *name* “twitter”
      * [?] Use clean name - without other language symbols
    - *FirstName* *website*(after http://) "twitter"
  + Twitter Search users with *name*, with every id having equal rank of 1
  + Combine candidate IDs from Google and Twitter search
  + Rank all twitter IDs based on
    - matching twitter profile and google scholar profile info (twitter bio URL with homepage, twitter bio description with GS Org, [?] twitter bio with GS research interests)
    - Return highest ranked if there is only one such ID
  + Get IDs ranked according to Google search frequency
    - Filter out to get highest ranked candidates
    - Rank and filter candidates according to String similarity with <FirstNameLastName> (threshold >=0.20)
      * Improve string similarity matching, currently errors occur when ‘simonwells’, ‘simonwells61', 'simonwells7', 'simon\_wells3', all map to ‘simonwells'; tune threshold
    - Return highest ranked candidate
    - Return None if this ranked candidates list is empty
    - browse through latest 10 tweets to find AI/ML/conference keywords

**Problems**

* **Generating** candidates which have correct account
* **Choosing** correct account from candidates [Ranking]
* Search results vary - algorithm gives different answers on different runs - webdriver starts to give empty results after a few runs, need to start colab notebook again after 10-15 runs

**[Milestone 1] Generating candidates**

* Improve extracting twitter ID from homepage
  + Sometimes not extracted if present if HTML page is broken/ not secure eg :<http://zive.info/> <https://web.media.mit.edu/~zive/>
  + Website may be outdated - GitHub site error (returns ‘githubstatus’, ignore this output)
* GS name and Twitter name may not match; eg: (Filip KRZN @krzn@masto.ai, Filip Korzeniowski), (小程 || Xiaocheng Tang, Xiaocheng Tang (唐小程))
  + Search -> firstName website(after http://) “twitter” [people generally mention their website on their twitter bio]
  + Modify name to include only a-zA-Z\s
* Search using different org name
  + Modify org name using Spacy -> search name org
  + Obtain latest org name using LinkedIn -> search name org
* Extract email from ’Verified email at ….’
* If <2 candidates are obtained
  + Search name on twitter

**[Milestone 2] Ranking Twitter IDs**

* Improving String matching to avoid losing important handles
  + Include metric to indicate top result
  + Include all candidates unless highest frequency is too high (>=4)
  + If max freq=[1-3], consider all candidates and rank acc to fuzzy string matching
  + Filter handles based on string matching score
  + Rank again according to twitter bio match with google scholar bio
* If Twitter name match with google scholar name is below a threshold, do not consider that results
* Twitter website matching with Google Scholar homepage [100% correct]
* Twitter bio similarity with Google Scholar topics and organisation
* String similarity of Google Scholar name and Twitter handle
* Match location of organisation and twitter user
* Frequency of appearing in search
* Crawl latest 5-10 tweets, check topic similarity with Google Scholar topics
  + In-domain keyword matching
  + Topic similarity
* Number of followers and following
* Top 1 twitter handle is chosen only if it crosses a string similarity threshold - this is done to increase precision in cases where twitter handles of scholars do not exist

**Correct Twitter ID not in top 10 candidates:**

* 32 - improve string matching algo to avoid mapping clashes between different twitter IDs
* 37
* 68 - Search for -> firstname website(after http://) "twitter"
* 76 - name extracted-entit-from-org "twitter"  
  import spacy  
  nlp = spacy.load('en\_core\_web\_sm')  
  doc = nlp("Senior Staff Member, MIT Lincoln Laboratory")  
  doc.ents[0]
* 110 - correct
* 113
* 115
* 134 - Search for -> firstname website(after http://) "twitter"  
  ans = re.search('[a-zA-Z\s]\*', 'Xiaocheng Tang (唐小程)').group(0).strip()
* 150 - improve string matching algo, match twitter bio with google scholar info, frequency of appearing in google search result is important
* 178 - is a protected account
* 200 - search name on twitter - use tweepy
* 202 - username is very different from name (uomian49, Mushtaq Raza) - improve ranking algo
* 222 - exclude words of length one from the name *twitter name is different from gs scholar name (Benjamin J Ainscough, Benjamin Ainscough) - excluding 'J' from googlesearch gives correct result*
* 224 - name (William R. Gray-Roncal, Ph.D.) and organisation name too complex (Johns Hopkins University Applied Physics Laboratory) - looks like they have 2 accounts - wgrbrains, WillGrayRoncal - also ambigous in tagging
* 226 - extract email from 'email' Verified email at umich.edu - <a href="http://vivekjoshi.ai/" rel="nofollow" class="gsc\_prf\_ila">Homepage</a>  
  re.search("Verified email at ([^\s]\*)", 'Verified email at umich.edu - <a href="http://vivekjoshi.ai/" rel="nofollow" class="gsc\_prf\_ila">Homepage</a>').group(0).split("Verified email at ")[-1]
* 229 - private account
* 241 - search twitter users, browse through latest few tweets to find AI/ML/conference keywords
* 248 - twitter user is very unlikely to be found by search (has 0 following, 0 followers, only 2 tweets since 2010)
* 255 - search on twitter, match expanded URL with homepage, description with GS bio
* 259 - search on twitter, match user.\_json['entities']['url']['urls'][0]['expanded\_url'] or ['display\_url] with homepage, 'description' with GS bio (people who do not have recent tweets or have 0 tweets are not shown in google search)  
  print(user.\_json['description'])  
  print(user.\_json['entities']['url']['urls'][0]['display\_url'])  
  print(user.\_json['entities']['url']['urls'][0]['expanded\_url'])  
  re.search("href=\"([^\s]\*)\"", gs\_scholar['email']).group(0).split("href=")[-1].replace("\"","").split("://")[-1].strip("/") == "https://www.rcv.sejong.ac.kr".split("://")[-1].strip("/")
* 264 - search on twitter, look for ML/AI related keywords in description and latest few tweets
* 271 - search on twitter
* 284 - private account