Uncovering Students' Social Networks: Entity Resolution Methods for Ambiguous Interaction Data

Mr. Adam Steven Weaver, Utah State University

Adam Weaver is a B.S. Mechanical Engineering student at Utah State University. His research is focused on developing explicit and efficient disambiguation methods for large-scale social network studies. In addition, he works with applications of Particle Image Velocimetry (PIV) and wrote curriculum using PIV to teach energy conservation to high school students.

Mr. Jack Elliott, Utah State University

Jack Elliott is a concurrent M.S. (Mechanical Engineering) and Ph.D. (Engineering Education) graduate student at Utah State University. His M.S. research is in experimental fluid dynamics, his Ph.D. work examines student social support networks in engineering education, and his other research activities include developing low-cost technology-based tools for improving fluid dynamics education.

Uncovering Student Social Networks: Entity Resolution Methods for Ambiguous Interaction Data

Introduction

Over the last century, cognitive psychologists have proposed that social interactions are a key component of student learning [1]–[4]. For example, Albert Bandura's Social Learning Theory [5] posits that students are influenced by their observation of *models* (e.g., peers, parents, etc.). Beyond learning, researchers have identified that students' retention rates are positively correlated to their access to individuals who can provide affective, financial, or informational support, especially in traditionally underrepresented groups [6]. Within these or similar theoretical foundations, engineering educators have identified several specific ways social interactions positively influence academic outcomes [7]–[12].

Among the methods for studying student interactions, Social Network Analysis (SNA) is uniquely suited to quantitatively explore relationships between social interactions and student learning. To conduct an SNA study, researchers use *name generator surveys* to prompt egos (individual study participants) to identify *alters* (individuals the ego interacts with). After collecting information about interactions, researchers represent interactions in an *adjacency matrix* by placing weighted values at the intersection between an individual's assigned row index and the alter's assigned column index. Using adjacency matrices, engineering education researchers can visually and quantitatively analyze student social networks and compare the students' network traits to desired outcomes.

However, the resources necessary to develop accurate, large-scale adjacency matrices may be limiting the scope of current SNA studies. One contributing factor to the high resource cost of large-scale SNA studies is the problem of *reference ambiguity* – references to an *entity* (a real-world individual) varying from the entity's *identity* (the real-world individual's correct name). Eliminating reference ambiguity is traditionally completed through entity resolution (ER) [13]. However, we are not aware of any studies in engineering education which have applied ER to raw interaction data. Rather, our literature searches show prior studies avoid reference ambiguity by focusing on simplified social environments, like single classrooms or online domains, where researchers can consistently collect accurate interaction data [14]–[16].

While these small-scale studies provide valuable insights on engineering students' networks, they neglect a significant portion of students' interactions [17], [18]. To analyze more holistic student networks, our research group is completing a large (i.e., 1000+ students), open-response network study of all first- and second-year undergraduate engineering students at a large, public land grant university in the U.S. (details in [19]). While completing this study, we realized that manually resolving students' ambiguous references to their correct entities was a significant challenge [20]. Further, we found that openly available ER resources often required training data and/or prior data-filtering [21], [22]. To this end, this paper is intended to describe and make available our efforts to filter and resolve raw interaction data using an automated ER module: EntityRAID (Entity Resolution for Ambiguous Interaction Data).

Background

Existing SNA research indicates there are several relationships between students' interactions and learning outcomes. For example, Grunspan and colleagues [23] performed a semester-long SNA study on 187 university students to identify that *network centrality* (how connected a student is to others) was significantly related to the exam scores in a university course. Further, Putnik et. al [24] performed an SNA study on fourth year engineering students, finding that students who had more frequent and varied types of interpersonal interactions attained higher final course grades. Using data from a learning management system, Gupta [25] found that high-school students who reported more interactions and/or were more popular had better grades than their counterparts.

While these and similar studies provide valuable insights regarding students' interpersonal networks and outcomes, Elliott and colleagues [18] found that students' course-specific study networks included ~20% non-course enrolled students. Thus, researchers may find broader relationships between social interactions and learning outcomes by performing SNA beyond a single cohort, classroom, or online environment.

Researchers hoping to generate broader conclusions, however, often face difficulties scaling to a larger network scope. For example, identifying holistic student networks may require researchers to use open-ended name generators. In Campbell and Lee's [26] review across four SNA studies, open-ended name generators prompting intimate connections yielded smaller ego-networks than studies asking about multiple types of connections, demonstrating the impact of survey length on participants' responses. While allowing holistic student responses, open-ended name generators can also introduce survey fatigue and reference ambiguity, causing many researchers to defect to close-ended name generators. To realize conclusions generalizable for engineering education, however, it is necessary to deploy less-bounded name generators and address the resulting reference ambiguity.

SNA study participants introduce reference ambiguity by referring to their alters by a) references that are spelled closely (*literal string similarity*) to the alter's actual identity or b) references that sound the same (*phonetic similarity*) to the alter's actual identity. For example, the alter "John Deer" could be referenced 'Jon Deare' in the name-generator data. In both cases, it is important for researchers to identify and correctly resolve name variants in the interaction data to build correct ego-networks. While manual approaches to find name variances exist [20], several computational methods for identifying string similarity show promise in rapidly resolving reference ambiguity.

The Levenshtein Distance (LD) [27], generates a number representing literal string similarity between a source string and a target string. Specifically, each letter insertion, deletion, or replacement adds a selected weight to the LD as shown in Figure 1.

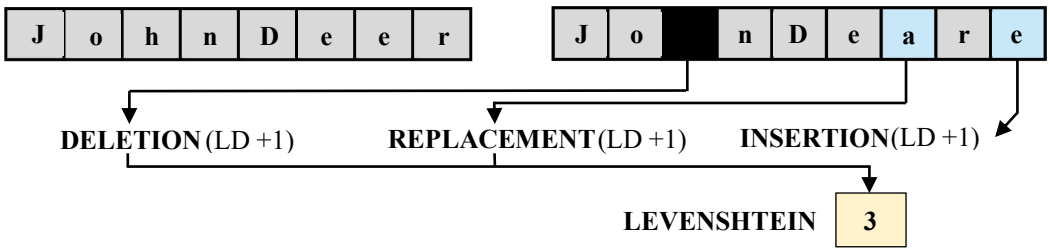

Figure 1. Visual representation of the Levenshtein Distance with weights of one assigned to: deletions, replacements, and insertions between the target string (JohnDeer) and source string (JonDeare).

Researchers have used the LD for various applications. For example, Haldar and Mukhopadhyay [28] applied the LD to improve dictionary lookup methods, and Klapaftis and Manandhar [29] used the LD in a combined-method entity resolution effort to disambiguate name references in Wikipedia articles.

The Double Metaphone algorithm (DM) [30] generates a key value that represents the pronunciation of a string. For example, names 'Caitlyn' and 'Katelynn' are not string similar (LD of 4) but have the same DM key value: ['KTLN', ''']. To gain a numerical understanding of phonetic similarity, researchers can use the LD on source and target DM key values. Both the LD and DM algorithms assisted our efforts in addressing SNA data reference ambiguity.

Methods

We began our efforts to disambiguate student network data by manually resolving a data set which contained 3.997 references (data collection described in [19]). We realized that our manual efforts, while effective, were time-consuming [20]. Therefore, to reduce the time spent to resolve future data sets, we wrote an automated entity resolution module, EntityRAID, to filter the ambiguous data in the same manner as our manual methods. To run EntityRAID, we input raw name-generator data. Additionally, we opted to input a roster of known identities to increase ER accuracy. After filtering the data in stages, EntityRAID returned a fully resolved data set and accompanying adjacency matrix, as depicted by Figure 2.

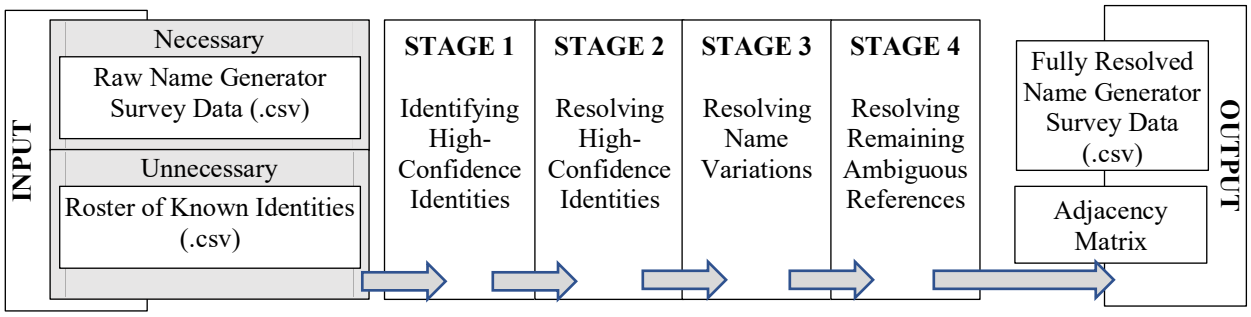

Figure 2. EntityRAID Input and Stages

Specifically, EntityRAID resolved *high confidence* references (references we could attribute to their correct identities in one step: participants' self-reported identities or roster-identities) first. EntityRAID then resolved *low confidence references* (references to non-participant identities) and concluded with *no confidence* references (first name references to non-

participant identities). Consequently, our confidence in accurately attributing references to identities decreased with each stage.

Stage 1: Identifying High Confidence Identities

To begin the EntityRAID module, we needed a method for recording identities, including self-reported nicknames. For this purpose, we wrote and called the function 'registryNames'. In 'registryNames', we initialized an empty list of key values and propagated the key with user-provided data containing each unique participant's identity and identifying number (when possible) as shown by Table 1. In our data collection methods, we used a registry query to invite potential participants to the study [19]. Recognizing the potential in these full-confidence identities, we used this registry of known identities to initialize our high confidence key.

Table 1. Example of High-Confidence Key

				Self-Reported Nicknames				
School #	Number	First Name	Last Name	First Name	Last Name	First Name	Last Name	Cont.
1583923	1	John	Doe	Jon	Doe	Jonny	Doe	…
1434950	2	Bob	Social	Bobbie	Social	Bobby	Social	…
…	…	…	…	…	…	…	…	…

After initializing a key of known identities, we wrote and called the function 'addParticipantNames' to add participants' *self-reported* full-name identities to the key. This function first checked if the participants' school numbers, which were also self-reported in our study, were present in the registry-initialized high confidence key. If the school number was present in the key, the 'addParticipantNames' function added the students' self-reported identity, including their self-reported nicknames, to the same registry-initialized key row where the school number appears. If not, 'addParticipantNames' assigned the participants' self-reported identity to a new key row.

One complication in the 'addParticipantNames' section of the module was adding participants' self-reported nicknames. In our study, participants reported their nicknames in a single survey entry. Specifically, we found students reported their nicknames in the following formats:

- [Nickname**, Lastname**]
- [Nickname 1**,** Nickname 2]
- [Nickname 1 **'and'** Nickname 2]
- [Nickname 1 **'or'** Nickname 2]

To address these inconsistencies, we wrote 'addParticipantNames' to identify if a participant used one of three common delimiters (i.e., "and", "or", or ",") in their response. If so, 'addParticipantNames' split the response into multiple nicknames by the respective delimiter. If participants reported their last name as part of their nickname, we neglected their last name to prevent 'addParticipantNames' from mixing first and last name entries in the key. After

delimiting values and assigning the correct last name, 'addParticipantNames' added each self-reported nickname to the correct key index, as shown by Figure 3.

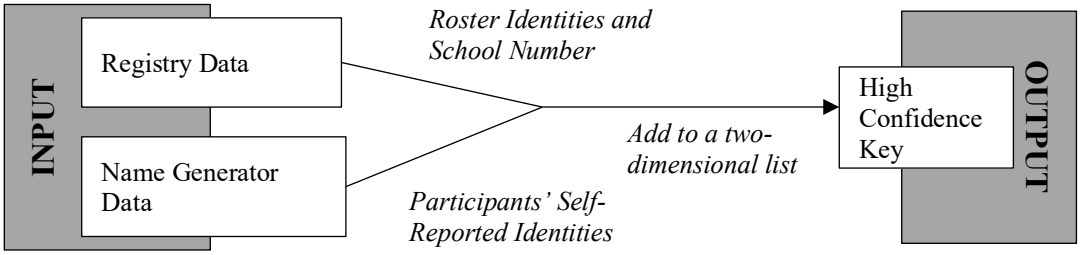

Figure 3. Stage 1 Data Flow Diagram.

At the conclusion of Stage 1, EntityRAID returned a *high-confidence key* which contained the participants' school number, assigned key number, and identity (including nicknames) in each row.

Stage 2: Resolving High-Confidence Identities

The second stage in the EntityRAID module was to resolve exact instances of high confidence identities we identified in Stage 1. To accomplish this step, we wrote and called the function 'replacingFunc', which looped through the filtered name generator data and replaced exact references to key identities with their corresponding key number, as portrayed by Figure 4.

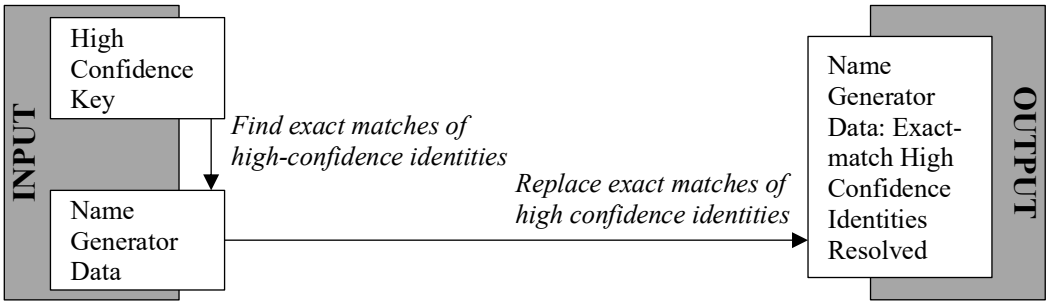

Figure 4. Stage 2 Data Flow Diagram.

At the conclusion of Stage 2, EntityRAID returned the name-generator data with all high-confidence references replaced by their corresponding identities' key number. At this stage of the data filtering, we left all remaining ambiguous references unresolved.

Stage 3: Resolving Name Variations

The third stage in the EntityRAID module was to discover similar *name pairs* (ambiguous references that are potential name variants of high confidence identities in the key).

To discover name pairs, we wrote and called the function 'compareKeytoData', which compared every ambiguous reference in the filtered data to every identity in the key, as shown by Figure 5.

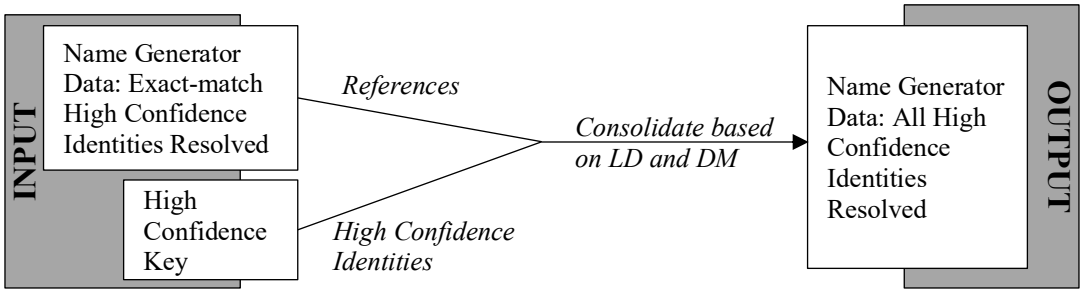

Figure 5. Stage 3 Data Flow Diagram

For each key identity and reference pair, we used the LD to identify the pair's literal string similarity and the LD of the DM key values (LD(DM)) to identify phonetic similarity. To identify thresholds for the LD and LD(DM), we first called 'compareKeytoData' with presumed over-relaxed thresholds. Using the thresholds to filter out unnecessary comparisons (known as *block cleaning* [31]), we inspected the remaining comparison results in .csv and tightened the thresholds to achieve more accurate pairs. After iterating, we heuristically determined that first and last name $LD \leq 2$ and first and last name $LD(DM) \leq 1$ consolidated name variances accurately. At the conclusion of Stage 3, EntityRAID returned the name-generator data with all high-confidence identities (including their variants) resolved.

Stage 4: Resolving Remaining Ambiguous References

After we resolved all participant references with confidence, the last stage in the EntityRAID module was to resolve all remaining ambiguous references. To accomplish this step, we wrote and called the function 'remainingAmbiguousNames', which created a *low confidence key* using non-participant references *with first and last names*. To distinguish the disambiguation results, we started the low confidence key numbering at the largest high-confidence key index plus one. Using the lower-confidence key, we used 'remainingAmbiguousNames' to loop through the data and replace exact matches of low confidence key identities with their corresponding key number, as depicted by Figure 6.

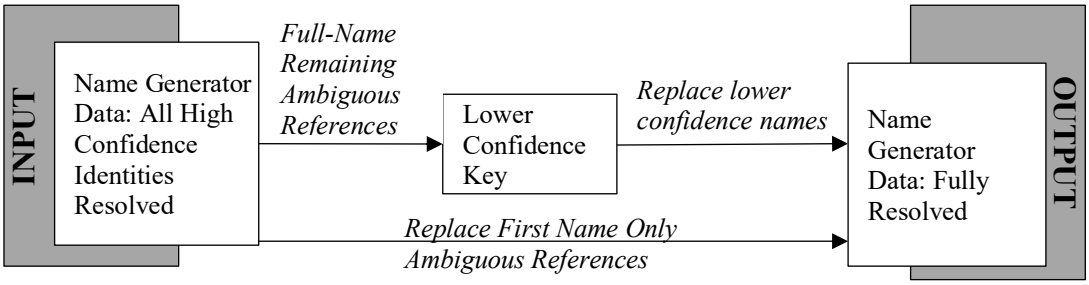

Figure 6. Stage 4 Data Flow Diagram

After developing and resolving references in the low confidence key, we determined that we did not have enough information to resolve first name-only references. For this reason, we replaced each first name-only reference, *including identical references*, with a newly assigned

number. Although we did not create a key for first name-only references, we started the numbering at the largest low-confidence key index plus one.

At the conclusion of the 'remainingAmbiguousNames' function, EntityRAID returned a fully resolved version of the raw survey data and the final key. With a resolved data set, EntityRAID also returns the adjacency matrix of the resolved interaction data.

Results and Discussion

We deployed EntityRAID on our study data with 8,034 references. For simplicity, we summarized our results in Figure 7.

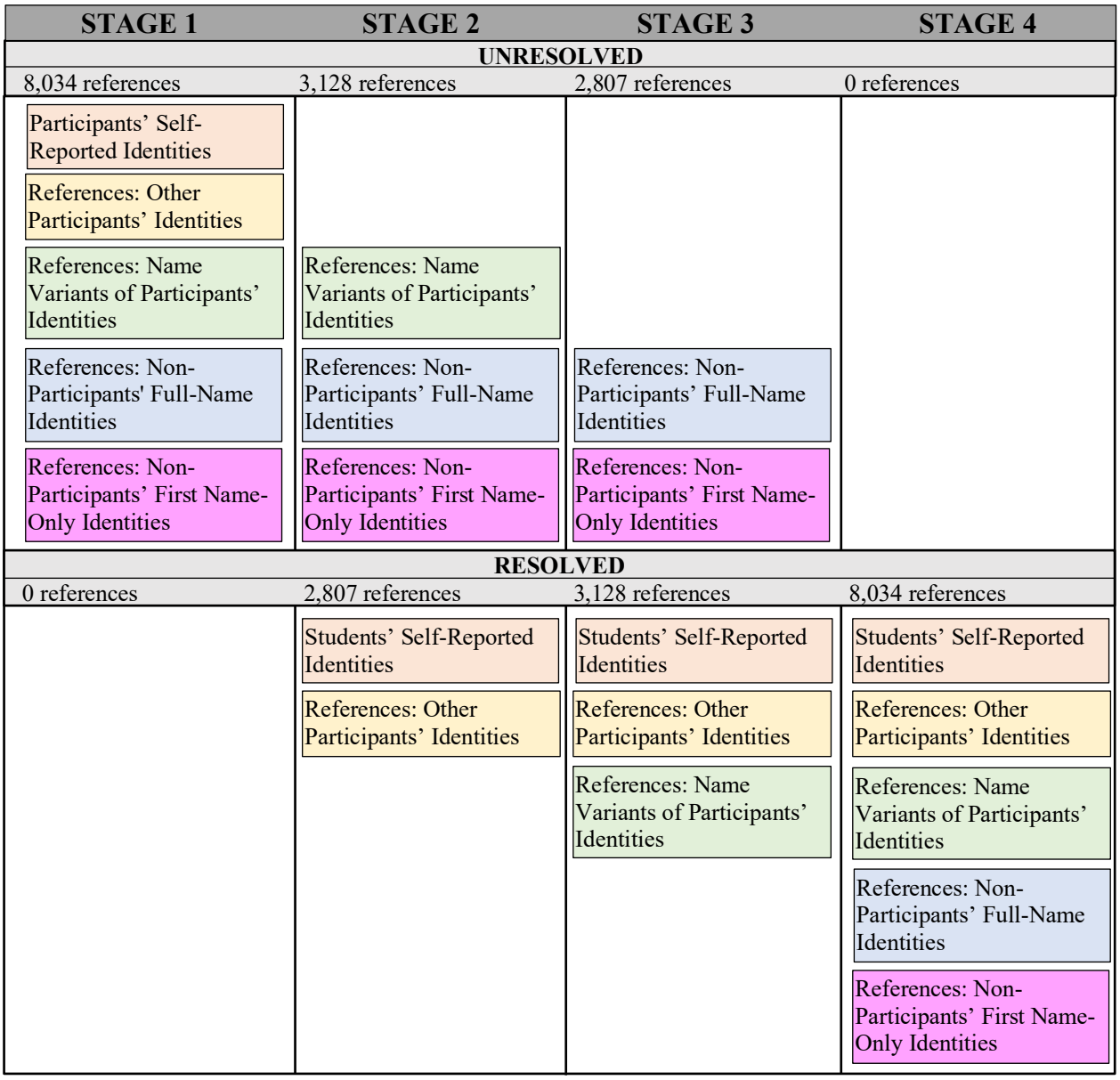

Figure 7. EntityRAID Disambiguation Map

More specifically, EntityRAID resolved 4,906 high confidence references (2,747 alters and 2,159 egos) in addition to 321 high confidence name variations, with the high-confidence key containing 1,848 identities. Of the remaining 2,807 ambiguous references in the data, we added 1,379 references to the full-name, lower confidence key. Using the lower confidence key, EntityRAID resolved 2,220 ambiguous references. Finally, EntityRAID provided 587 first name-only references (13 of which were self-reported identities) a new key number. We summarized the results in Figure 7.

We consider EntityRAID successful in reducing the researchers' resource cost, completing the majority of an estimated 200-hour manual disambiguation process in 2 hours. While retaining the significant time-savings, we could manually identify network proximity measures to resolve the remaining ~7% (i.e., first name-only entries) with higher confidence. Specifically, by combining network proximity scores with the existing literal- and phonetic-string similarity scores [32], we may further gain the information we need to resolve first name-only entries with confidence. For example, if 'Andrew Dendrogram' interacts with 'Benjamin' and 'Benjamin Cluster' reports interacting with 'Andrew Dendrogram', then we are more confident that the reference 'Benjamin' should be consolidated to 'Benjamin Cluster'.

Notably, EntityRAID returns an adjacency matrix resolved according to the full key. If researchers choose to use network similarity measures to resolve first name-only entries, a capable computational method to deploy is *community detection* (the process of identifying closely related nodes in a large nodal network) using the adjacency matrix. However, employing community detection techniques can be time-consuming and must be analyzed for accuracy. With only ~7% of references remaining to resolve, manual resolution [20] proved sufficient for our subsequent ego-network analysis.

Conclusion

Automated entity resolution methods can significantly reduce the time-resource cost for large-scale, holistic, educational SNA. To make automated entity resolution methods more accessible, we wrote EntityRAID to filter and resolve raw student interaction data. EntityRAID operates in four main functions: 1) 'addParticipantNames' builds a key using high confidence identities (i.e., students' self-reported names/nicknames and roster names) 2) 'replacingFunc' resolves each exact-match reference to the high-confidence key 3) 'compareKeytoData' resolves name variants to the high confidence key using LD and DM and 4) 'remainingAmbiguousNames' resolves remaining ambiguous references with a lower confidence key or new number.

Using EntityRAID on a data set of 8,034 references, we resolved 5,227 references with high confidence and 2,220 with lower confidence (~93% total references resolved with confidence). Further, we resolved 587 references with no confidence. For our ego-centric study, EntityRAID was adequate to gain helpful conclusions for large interaction data sets. To improve accuracy, users may deploy manual or clustering techniques directly on EntityRAID's output data. Overall, EntityRAID significantly reduces the resource requirement for performing large-scale SNA. Reducing resource cost will enable engineering educators to research more holistic

student networks than previously studied. Results of these future studies may yield more generalizable and accurate conclusions about which social practices help students succeed.

Acknowledgements

This material is based upon work supported by the second author's National Science Foundation Graduate Research Fellowship under Grant No. DGE1745048. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] A. Kozulin, *Vygotsky's Psychology: A biography of ideas*. Cambridge, MA: Harvard University Press, 1990.

[2] P. E. Doolittle, "Understanding Cooperative Learning through Vygotsky's Zone of Proximal Development.," 1995.

[3] J. G. Greeno, A. M. Collins, and L. Resnick, "Cognition and learning," in *Cognition and Learning*, 1996, pp. 15–46.

[4] L. S. VYGOTSKY, *Mind in Society*. Harvard University Press, 1978. doi: 10.2307/j.ctvjf9vz4.

[5] A. Bandura, "Albert Bandura- Social Learning Theory," *Simply Psychology*, 1977.

[6] J. P. Martin, D. R. Simmons, and S. L. Yu, "The Role of Social Capital in the Experiences of Hispanic Women Engineering Majors," *Journal of Engineering Education*, vol. 102, no. 2, pp. 227–243, Apr. 2013, doi: 10.1002/jee.20010.

[7] S. Freeman *et al.*, "Prescribed Active Learning Increases Performance in Introductory Biology," *CBE—Life Sciences Education*, vol. 6, no. 2, pp. 132–139, Jun. 2007, doi: 10.1187/cbe.06-09-0194.

[8] R. R. Hake, "Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," *Am J Phys*, vol. 66, no. 1, pp. 64–74, Jan. 1998, doi: 10.1119/1.18809.

[9] D. J. Zimmerman, "Peer effects in academic outcomes: Evidence from a natural experiment," *Review of Economics and Statistics*, vol. 85, no. 1. 2003. doi: 10.1162/003465303762687677.

[10] N. A. Bowman, L. Jarratt, L. A. Polgreen, T. Kruckeberg, and A. M. Segre, "Early identification of students' social networks: Predicting college retention and graduation via campus dining," *J Coll Stud Dev*, vol. 60, no. 5, 2019, doi: 10.1353/csd.2019.0052.

[11] B. Hurst, R. Wallace, and S. B. Nixon, "The impact of social interaction on student learning," *Reading Horizons*, vol. 52, no. 4, 2013.

[12] C. C. Liu, Y. C. Chen, and S. J. Diana Tai, "A social network analysis on elementary student engagement in the networked creation community," *Comput Educ*, vol. 115, 2017, doi: 10.1016/j.compedu.2017.08.002.

[13] L. Getoor and A. Machanavajjhala, "Entity resolution: theory, practice & open challenges," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2018–2019, 2012.

[14] D. R. Berg, "Twitter in the engineering classroom," *Journal of Online Engineering Education*, vol. 6, no. 2, p. 4, 2015.

[15] M. de Laat, V. Lally, L. Lipponen, and R.-J. Simons, "Investigating Patterns of Interaction in Networked Learning and Computer-Supported Collaborative Learning: A Role for Social Network Analysis," *Int J Comput Support Collab Learn*, vol. 2, no. 1, pp. 87–103, 2007, [Online]. Available: http://dist.lib.usu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=eric&AN=EJ854448&site=eds-live

[16] H. B. Love, M. E. Ozbek, and J. E. Cross, "Assessment of the Development of Social and Learning Networks in Construction Management Courses Using Social Network Analysis," *Int J Constr Educ Res*, vol. 16, no. 4, pp. 290–310, 2020, doi: 10.1080/15578771.2019.1657208.

[17] N. Pearson, J. Major, A. Godwin, and A. Kirn, "Using social network analysis to study the social structures of inclusion," in *ASEE annual conference & exposition*, 2018.
[18] J. Elliott, A. Minichiello, and J. Ellsworth, "Examining Relationships Between Student Interactions with Peers and Resources and Performance in a Large Engineering Course Using Social Network Analysis," in *2020 ASEE Virtual Annual Conference Content Access Proceedings*, ASEE Conferences, Jun. 2020. doi: 10.18260/1-2--34612.
[19] J. Elliott, A. Minichiello, and J. D. Marquit, "Work in Progress: An Investigation of the Influences of Peer Networks on Engineering Undergraduate Performance Outcomes." Jul. 2021.
[20] A. Weaver and J. Elliott, "Work in Progress: Developing Disambiguation Methods for Large-Scale Educational Network Data," Vancouver, B.C., 2022.
[21] "Zingg." Aug. 04, 2022.
[22] J. de Bruin, "Python Record Linkage Toolkit: A toolkit for record linkage and duplicate detection in Python." Zenodo, Dec. 2019.
[23] D. Z. Grunspan, B. L. Wiggins, and S. M. Goodreau, "Understanding classrooms through social network analysis: A primer for social network analysis in education research," *CBE Life Sci Educ*, vol. 13, no. 2, 2014, doi: 10.1187/cbe.13-08-0162.
[24] G. Putnik, E. Costa, C. Alves, H. Castro, L. Varela, and V. Shah, "Analysing the correlation between social network analysis measures and performance of students in social network-based engineering education," *Int J Technol Des Educ*, vol. 26, no. 3, pp. 413–437, Aug. 2016, doi: 10.1007/s10798-015-9318-z.
[25] A. Gupta, "Application of Human Factors Engineering Principles to the Devlopment Of Social Network Analysis (SNA) Assessment Tools for Use By Teachers Within A Collaborative Educational Environment," Master's Thesis, Tufts University, Medford, MA, 2014.
[26] K. E. Campbell and B. A. Lee, "Name generators in surveys of personal networks," *Soc Networks*, vol. 13, no. 3, pp. 203–221, Sep. 1991, doi: 10.1016/0378-8733(91)90006-F.
[27] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady*, vol. 10. pp. 707–707, Feb. 1966.
[28] R. Haldar and D. Mukhopadhyay, "Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach," Jan. 2011.
[29] I. P. Klapaftis and S. Manandhar, "Unsupervised named entity resolution," in *Proceedings of the 3rd IEEE International Conference on Multimedia Communications, Services and Security*, 2010.
[30] L. Philips, "The Double Metaphone Search Algorithm," *C/C++ Users Journal*, vol. 18, pp. 38–43, Jun. 2000.
[31] G. Papadakis, J. Svirsky, A. Gal, and T. Palpanas, "Comparative analysis of approximate blocking techniques for entity resolution," *Proceedings of the VLDB Endowment*, vol. 9, no. 9, pp. 684–695, May 2016, doi: 10.14778/2947618.2947624.
[32] P. W. Horng-Jyh, N. Jin-Cheon, and C. K. Soo-Guan, "A hybrid approach to fuzzy name search incorporating language-based and text-based principles," *J Inf Sci*, vol. 33, no. 1, 2007, doi: 10.1177/0165551506068146.