

New Entity Resolution Solutions for the Global Analytics Platform of JSOC

Detravious Brinkley, Kathryn Foss*, Goran Giudetti, and Arpita Khare

University of Southern California, Los Angeles, California, 90089-0482, United States

* Corresponding author: *katiefos@usc.edu*

May 2022

Abstract

Abstract goes here

1 Introduction

Entity Resolution (in its simplest form) is the task of identifying whether two pieces of information are referring to the same entity (object, person, place etc.) or if they are referring to two distinct ones. The Global Analytics Platform (GAP) team at the Joint Special Operations Command (JSOC) currently purchases a license for a proprietary entity resolution solution to resolve persons, vessels and organizations. This solution takes as input 42 text-only datasets and resolves billions of entities each week in a batch process. GAP Intelligence analysts then use the returned resolved dataset to answer mission critical questions. Our team spent the semester interviewing industry experts, GAP mission managers, Novetta engineers, and external intelligence analysts to access dual use cases, learn about entity resolution pain points, and better understand the entity resolution solution currently in place at GAP. We came to learn key requirements the current solution meets, as well as a set of future requirements that we believe GAP analysts would benefit from. In this paper we present our findings in four sections. Firstly we present what the current state of entity resolution is at GAP. Secondly we present the dual use cases outside of the Department of Defense. Thirdly we discuss a brief technical overview of entity resolution. And lastly we present a set of evaluation criteria for open source solutions as well as a comparison of six open source solutions.

2 Entity resolution at GAP

The proprietary entity resolution solution currently in use by the GAP team is Novetta Entity Analytics by Accenture Federal Services. To understand the solution at a technical level as well as its day-to-day use by end users we interviewed Novetta engineers, GAP mission managers, and GAP data scientists. Those we interviewed on the GAP team are in bold in Figure 1. Through our interviews, we identified two customer archetypes. The first is a GAP intelligence analyst. GAP intelligence analysts have to answer mission critical questions and the line between analyst and data science can be blurry. The other customer archetype we identified is an Army Battalion Intelligence Officer. The Battalion Intelligence Officer we interviewed had a team of 6 to provide for the intelligence needs of 600 people. Additionally, we learned that within GAP there are two ways entities can be resolved. Firstly is through Python scripts data scientists have written for ad hoc resolution. The second is through the Novetta (an enterprise) solution and can be accessed through direct link (to AWS) or through the JSOC Search Engine[1]. During our interviews we compiled a list of characteristics and requirements that an enterprise entity resolution solution should provide to meet the status quo. The solution should support multi-entity resolution, multi-source resolutions, be scalable, and mask U.S. Citizen personally identifiable information.

Any future entity resolution solution should support multi-entity resolution as the current solution already supports resolving three different entities, persons, organizations and vessels. COL Clark expressed interest in resolving full phone communication group networks when a group of people change phone numbers. This could be a fourth entity type that could be resolved. Additionally, a future solution should support multi-source resolution as the current solution resolves 42 different datasets. The solution should also be scalable

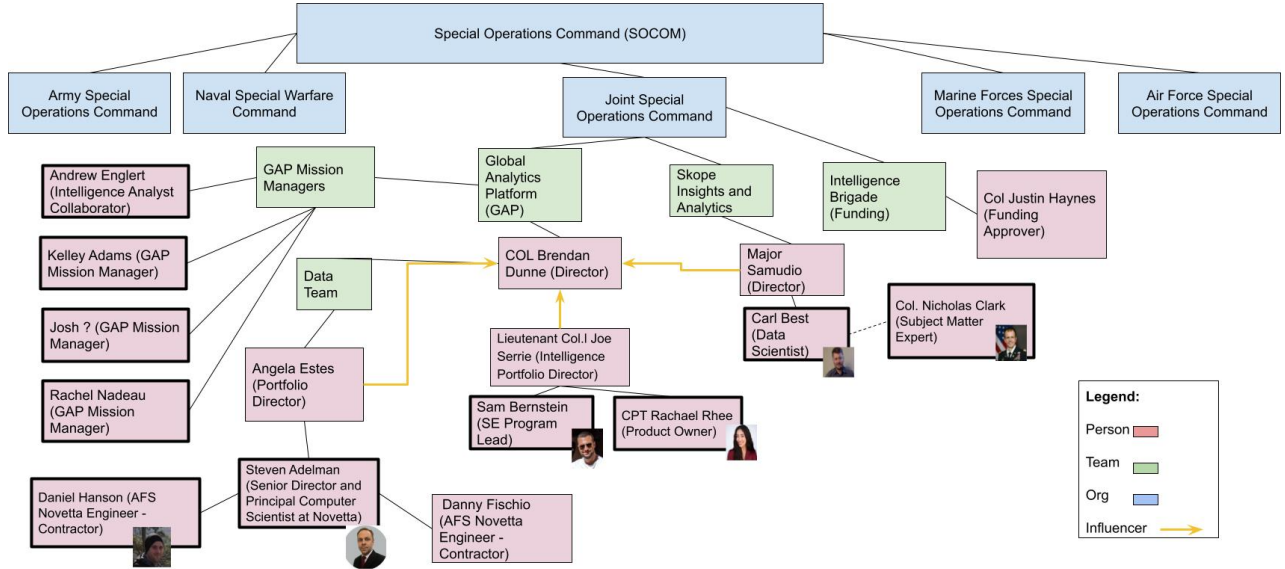


Figure 1: Unofficial org-chart of the Joint Special Operation Command.

as the current solution resolves 10s of billions of entities and the resolved dataset requires 13 TB to store. Lastly a solution will need to mask PII. While U.S. citizens PII is used to resolve entities, it has to be masked so no one within JSOC can identify the U.S. citizen.

3 Dual Use

To better understand the problem ecosystem, our team conducted interviews to learn how organizations outside of GAP approach entity resolution. While we conducted over 70 interviews across a dozen organizations, we found three sectors that either currently have entity resolution solutions or have strong use cases for an entity resolution solution.

The first sector we looked at was the intelligence community. We had the opportunity to talk to a retired FBI Program Manager (Kyle Albert) and a current FBI Intelligence Analyst (Alison McGriff). During these interviews we learned about the Parkland Shooting Investigation in 2018. The Parkland Shooter only used his full name in one or two of his social media accounts, the rest of his accounts he used alias names. To track down all of his accounts 150-200 agents sat in a SCIF (sensitive compartmented information facility) 24 hours a day for a little over a month's time. Identifying the owner of social media accounts is a form of entity resolution. While it is our understanding that this was mostly a manual process in 2018, it is probable that the FBI now has more automated tooling. We additionally conducted another interview with an algorithm engineer who works at one of the eighteen intelligence agencies under the US Intelligence Community. This engineer shared that they have in the past worked on building an internal entity resolution solution for their agency. While we were not able to determine what entity resolution tooling the FBI currently uses, we think it would be beneficial for GAP to follow-up with these intelligence agencies to determine what solutions are in place and what requirements the solutions meet.

The next sector we looked at was Entertainment. Our team interviewed a Principal Data Science Engineer and a Senior Manager in Analytics Operations at Walt Disney Media & Entertainment Distribution. Walt Disney has multiple streaming platforms, ESPN+, Disney+, Hulu and ABC. Oftentimes a customer may have an account on one streaming site and log-in, and later visits another platform without logging in. In order to best serve their customers, Disney is interested in knowing who is visiting their sites, even when they don't log in with an account. Entity resolution can be used to identify users in the above case. While Disney does not resolve the identities themselves, they use an Identity Broker to determine the identity of users whom they are not able to resolve themselves. Another important use case for identity resolution is CCPA compliance. If a user requests their personal information be deleted, Disney needs to be able resolve the identity of the user even if they have not created an account with Disney to make sure they will not now or in the future store this user's personal information. A data broker will return a unique identifier for the user that Disney can use for CCPA compliance.

The last sector to highlight is immunization and disease tracking at the state and federal levels. We interviewed the section manager of the Division of Immunizations at the Michigan Department of Health and Human services (MDHHS), and a retired Department Specialist with MDHHS. During these interviews

we learned Medical provider’s are required to report immunizations to the State of Michigan for persons under the age of 20. However, because of privacy concerns social security numbers are not used as unique identifiers when sending immunization data to the state. The state then needed a way to deduplicate immunization records, especially because patients often don’t receive all of their immunizations from one provider. Originally the state had their own deduplication algorithms they maintained, however in the last five years the state has adopted a Master Person’s Index (MPI). The MPI returns a unique identifier for a unique individual and is part of the Master Data Management (MDM) system provided by Optum. The MDM resolved entities with the help of dozens of state agencies participating to provide input data sources. The MPI is now integrated into the Health Information Exchange and seamlessly integrates into major EHR systems like Epic. Now when a doctor’s office uses an EHR system to record the administration of a vaccine, a message is automatically sent to the state.

Each of these sectors showcase the widespread use of entity resolution solutions across multiple industries. During our customer discovery we interviewed experts across many fields outside of the three sectors for example nurses, police officers, intelligence officers, educators, data standards advocates, software engineers, data engineers, and satellite engineers. We highlighted the three sectors (intelligence community, entertainment, and immunization tracking) because the pain points these sectors face and the solutions they are currently using most closely align with GAPs problem statement.

4 Technical Overview of Entity Resolution

Entity Resolution in its most basic form can be broken up into 4 steps: schema alignment, blocking, matching and clustering. Figure 2 shows entity resolution in its first generation, while Figure 3 shows the 4th and current generation of entity resolution. As the volume, veracity, and variety of data has grown, new blocking, parallelization, and prioritization techniques have been developed, but overall the building blocks in the first generation have remained constant. This section will provide a brief overview of the 4 steps of entity resolution as pictured in Figure 2.

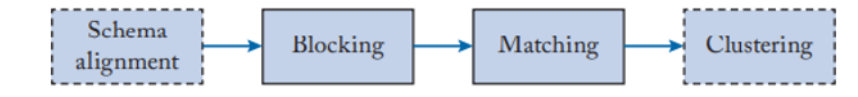


Figure 2: Source [2].

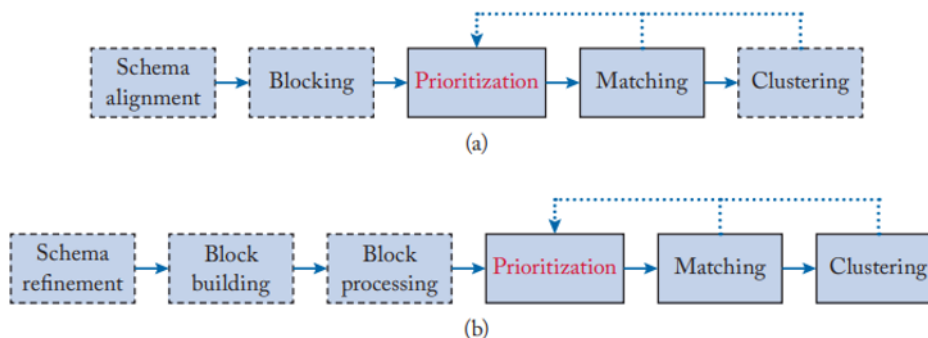


Figure 3: Source [2].

If a schema-aware blocking method is used, then the first step of entity resolution is schema alignment. (If a schema-agnostic blocking method is used, the first step is schema refinement.) Schema alignment is the process of aligning attributes across datasets to a standard schema by evaluating data headers and values.

The second step is blocking. The goal of blocking is to reduce the number of pairwise comparisons by only comparing records that fall into the same block based on a blocking key. In schema-aware blocking, a block key could be a zipcode field and only records with the same zip code would be compared when evaluating pairs for a matching in step 3. Without blocking, the number of comparisons needed would be $O(n^2)$. There are a variety of ways that blocking can be approached. Table 1 is a survey of the 4 different types of blocking methods completed by Papadakis et. al. The blocking methods described in Table 1 can be summarized as followed:

1. Non-learning vs learning-based methods: Non-learning methods rely on rules that were derived from experts while learning-based methods rely on a training set to learn the best keys using ML techniques for the blocking key selection process.
2. Schema-awareness: Schema-awareness can be separated into schema-aware and schema-agnostic methods. Schema-aware techniques rely on the schema when extracting blocking keys for matching while schema-agnostic methods disregard the schema and extract the blocking keys from all attributes.
3. Key type: The key type can be hash-based or similarity-based. Hash-based means that if two entities have the same key they are mapped to the same block. Sort-based means that if two entities have similar keys they get mapped to the same block. There can also be hybrid-based which is a combination of the two.
4. Redundancy-awareness: There's three techniques under this category. First, redundancy-awareness deals with the idea of giving each entity its own disjoint block (avoiding all redundancies). Redundancy positive methods are placing entities into multiple blocks causing an overlap to detect similarity. Redundancy-neutral is the idea that overlapping blocks are created but the degree of redundancy is arbitrary.
5. Constraint-awareness: There are two types of constraint awareness: lazy and proactive. Lazy blocking methods don't impose constraints on the created blocks while proactive blocking methods enforce constraints like maximum block size.
6. Matching-awareness: The two types of matching-awareness methods are static and dynamic. Static means the created block collection is immutable while dynamic methods allow for updates to the collection as duplicates get detected.

Method	Key type	Redundancy awareness	Constraint awareness	Matching awareness
Standard Blocking (SB) [50]	hash-based	redundancy-free	lazy	static
Suffix Arrays Blocking (SA) [3]	hash-based	redundancy-positive	proactive	static
Extended Suffix Arrays Blocking [25, 116]	hash-based	redundancy-positive	proactive	static
Improved Suffix Arrays Blocking [34]	hash-based	redundancy-positive	proactive	static
Q-Grams Blocking [25, 116]	hash-based	redundancy-positive	lazy	static
Extended Q-Grams Blocking [11, 25, 116]	hash-based	redundancy-positive	lazy	static
MFIBlocks [79]	hash-based	redundancy-positive	proactive	static
Sorted Neighborhood (SN) [64, 65, 136]	sort-based	redundancy-neutral	proactive	static
Extended Sorted Neighborhood [25]	sort-based	redundancy-neutral	lazy	static
Incrementally Adaptive SN [189]	sort-based	redundancy-neutral	proactive	static
Accumulative Adaptive SN [189]	sort-based	redundancy-neutral	proactive	static
Duplicate Count Strategy (DCS) [42]	sort-based	redundancy-neutral	proactive	dynamic
DCS++ [42]	sort-based	redundancy-neutral	proactive	dynamic
Sorted Blocks [41]	hybrid	redundancy-neutral	lazy	static
Sorted Blocks New Partition [41]	hybrid	redundancy-neutral	proactive	static
Sorted Blocks Sliding Window [41]	hybrid	redundancy-neutral	proactive	static
(a) Non-learning, schema-aware methods.				
ApproxRBSetsCover [16]	hash-based	redundancy-positive	lazy	static
ApproxDNF [16]	hash-based	redundancy-positive	lazy	static
Blocking Scheme Learner (BSL) [104]	hash-based	redundancy-positive	lazy	static
Conjunction Learner [21] (semi-supervised)	hash-based	redundancy-positive	lazy	static
BGP [49]	hash-based	redundancy-positive	lazy	static
CBlock [150]	hash-based	redundancy-positive	proactive	static
DNF Learner [55]	hash-based	redundancy-positive	lazy	dynamic
FisherDisjunctive [76] (unsupervised)	hash-based	redundancy-positive	lazy	static
(b) Learning-based (supervised), schema-aware methods.				
Token Blocking (TB) [120]	hash-based	redundancy-positive	lazy	static
Attribute Clustering Blocking [124]	hash-based	redundancy-positive	lazy	static
RDFKeyLearner [158]	hash-based	redundancy-positive	lazy	static
Prefix-Infix(-Suffix) Blocking [123]	hash-based	redundancy-positive	lazy	static
TYPiMatch [96]	hash-based	redundancy-positive	lazy	static
Semantic Graph Blocking [113]	-	redundancy-neutral	proactive	static
(c) Non-learning, schema-agnostic methods.				
Hetero [77]	hash-based	redundancy-positive	lazy	static
Extended DNF BSL [78]	hash-based	redundancy-positive	lazy	static
(d) Learning-based (unsupervised), schema-agnostic methods.				

Figure 4: Source [2].

The third step in entity resolution is matching. Papadakis et. al. break up matching functions into 7 categories as pictured in Figure 4. Across these 7 categories, in the first generation of entity resolution alone, Papadakis et. al. compiled 31 different matching functions. These matching functions determine whether or not two entities are referring to the same single entity (if a deterministic function is used) or a probability that two entities are the same entity (if a probabilistic method is used).

The last step in entity resolution is clustering. Clustering is an optional step that partitions entities so they are each in a disjoint set.

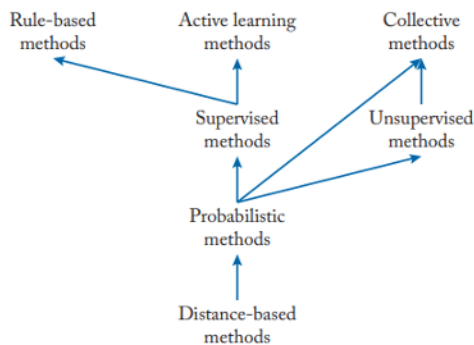


Figure 5: Source [2].

The remainder of this paper will focus on the requirements used to evaluate open source solutions and a survey of 6 open source solutions. However, it is important to note the breadth and depth of the entity resolution field. There are 4 generations of entity resolution, each adding new blocking and matching techniques. Along the way additional steps to deal with the increasing veracity, volume and variety of data have also been added. It was out of scope for our team to evaluate all of the blocking, matching and clustering methods available, but we think it important for the GAP team to be aware of the evolution of entity resolution at a high level when evaluating open source solutions.

5 Open Source Discovery Methodology

When evaluating solutions we looked at the features Novetta Entity Analytics already supports as well as pain points we identified during customer discovery. In total we evaluated open source solutions on 5 criteria: multi-entity, multi-source, data at scale, graphical user interface, process type, and underlying database. This section will focus on the criteria not yet mentioned: graphical user interface, process type and underlying database.

In an interview with mission managers we learned that visualizations of resolved entities would be helpful for intelligence analysts of all levels. Additionally during these interviews we learned that intelligence analysts find the number of clicks required to be far too high in the Novetta solution. Intelligence analysts want a seamless experience from within the JSOC search engine tool. During our search for open source solutions, we looked for solutions that had a user interface to give the GAP development team a place to start when building out a graph visualization.

The next criteria we evaluated was process type. While the Novetta solution currently resolves all entities once a week in a batch process, a Novetta Engineer shared there was a desire to move from a batch process to incremental process type. An incremental load will only resolve new entities as they become available, instead of re-running all historical previously resolved entities. When we talked to Senzing, another proprietary entity resolution provider we learned of a concept they call sequence neutrality. This means the order an entity arrives does not matter, the same result will be returned. Not all blocking and matching functions support what Senzing calls sequence neutrality or what we’ve found in literature to be called order independence [3]. When order independence isn’t satisfied and entities are resolved incrementally there can be drift from what would otherwise be the ground truth. It is for this reason our team wanted to specifically call out which solutions supported incremental entity resolution vs. which algorithms support only batch loads.

The last criteria we evaluated was underlying database type. The Novetta software is not based on a graph database while much of the literature we surveyed proposed using an underlying graph database. A graph database contains a set of nodes (also known as vertices) and a collection of edges that connect a pair of nodes. Figure 6 shows an example result of a graph-based ER solution.

Representing entities and attributes in this way provides 2 advantages. Firstly, a graph structure supports the use of deterministic, probabilistic, and clustering matching algorithms. Secondly, a graph representation of attributes and entities can be used to draw relationships between different entity types. For these reasons we prioritized graph-based open source solutions.

In addition to the above criteria, there is one additional requirement we were not able to fully evaluate. Intelligence analysts do not take data returned from a solution blindly as ground truth. Analysts build trust in a solution over time, and one of the ways trust is built is by providing the end-user with a “why and how” an entity was resolved[5]. In a graph based solution this means being able to explain why edges are drawn between entities. While we ran out of time to evaluate which algorithms support the “why and how”, we

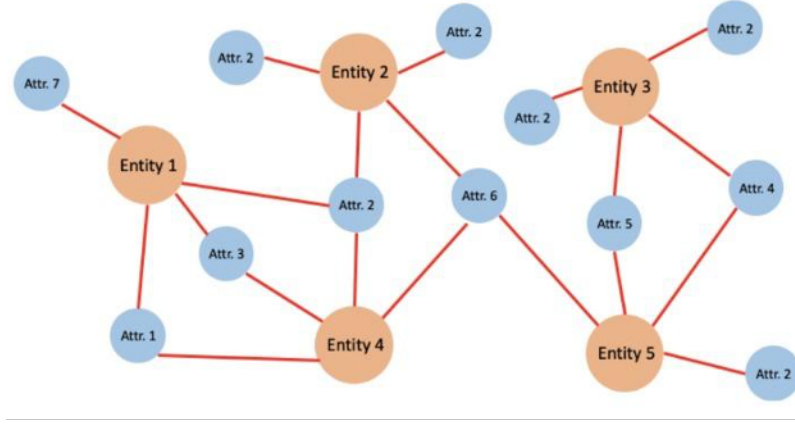


Figure 6: Example of graph that shows the common attributes among entities[4].

think including this feature will be important for the success of any future solution.

6 Open Source Solutions

After developing our criteria list, we worked on researching open source solutions that meet as many of those criteria points as possible. Below is a list of our final suggestions on open-source solutions that might be worth exploring further. For the chart above, this is what our vetting process looked like for our final



	Scalable	Open Source	Graph	Multi Source	GUI	Description	Process Type
	X	X	X	X	X	Meta Blocking, serial or parallel processing, schema or schema-agnostic	Batch
Usc Isi I2 Ritk	X	X	X	X		Extensible, Customizable	Batch
	X	X	X	X		Domain Specific	Batch
	X	X	X	X	X	Incremental Spark Blocking and Repairing. Parallel processing with Apache Flink	Incremental or Batch
SparkER	X	X	X	X		BLAST Meta Blocking Implementation for Spark	Batch
Dblink	X	X	X	X		Based on Bayesian model	Batch
	X			X	X	Proprietary, domain specific algorithm for record attributes	Incremental
	X		X	X	X	Proprietary, but extensible	?

Figure 7: List of software considered in our survey and their related properties

solutions. The bottom two solutions are proprietary solutions that will be discussed briefly at the end of this section. The rest are open-source solutions which checked off key criteria, which made them viable to propose as possible solutions.

6.1 JedAI Toolkit

JedAI toolkit can be applied to any domain and has different workflows that can be adapted. JedAI (Java generic Data Integration) and is being contributed to by multiple universities (Universite de Paris, University

of Athens, Unimore, to name some). It uses meta-blocking techniques and has a UI application that can be tested as well as a paper with all their findings[6, 7].

Pros: JedAI toolkit supports multiple techniques for entity resolution rather than relying on a select few. Their website explains that they use blocking, join-based, and progressive methods depending on which one is needed by the user. They call what they do three-dimensional entity resolution where (as shown in Figure 8), the solution space includes picking from parallelization or serial processing, schema-based or schema-agnostic, and budget-agnostic or budget-aware. More detail on each is provided in their paper which can be found on their website[8]. *Cons:* It is a Java based toolkit and has instructions on how to get it to

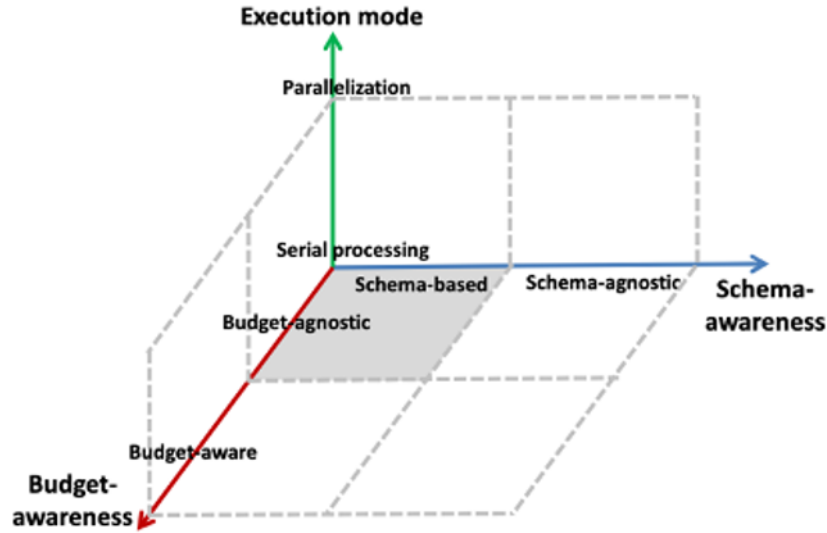


Figure 1: The solution space of the end-to-end ER pipelines that can be constructed by JedAI.

Figure 8:

work with python as well, yet this might be a restriction if a different language is being used currently for development[9].

6.2 USC/ISI RLTK(record linkage toolkit)

The Record Linkage Toolkit is being developed at USC in the Information Sciences Institute. It claims to make it very easy to add new features to the already existing codebase and gives the user arbitrary degrees of control on the individual features. It is fully scalable, and uses multi-core algorithms[10].

Pros: The USC Record Linkage toolkit is a python-based open-source solution and claims that one of its main purposes was to make a scalable solution. Beyond that, their use of multi-core algorithms include blocking, profiling data, and using ML classifiers based on Python’s sklearn library.

Cons: The solution is a python-based solution tailored for python applications. The developers are also currently working on adding record-linkage algorithms so this might not be something that’s fully developed as of today. It is also important to note that the last code contribution happened about a year ago so it may not be getting updated currently, in case any problems arise.

6.3 FAMER

FAMER, or Fast Multi-Source Entity Resolution System, is a graph-based open-source solution which involves multi-source resolution. It claims to use clustering techniques and supports incremental linking[11, 12].

Pros: FAMER does support incremental linking which makes it stand out from the other solutions. Beyond that it uses an approach called clustered similarity graph. They also have a visualization tool to visualize the cluster graphs.

Cons: A warning on their Github page is that FAMER is currently a work in progress so the APIs may change. They consider their work a proof of concept and not yet ready for production.

6.4 Zingg

Zingg is a scalable, graph-based solution. It has consistent code contributions and uses both blocking and similarity models. The github also provides an office hours link to get help or more information about this software[13].

Pros: Zingg claims to do multi-source entity resolution well. This means that the data can be in different types of databases and Zingg has a way of joining those data sources. It can also scale for large volumes of data and also claims it is easy to build models on smaller training samples for a high accuracy return. It also caters to domain-specific data, and supports several different languages (not referring to programming languages).

Cons: On an initial glance, it does not seem that Zingg has a fancy GUI that comes with it. Based on a cursory glance of open issues on their Github, it also seems they are under progress for adding support to integrate with certain platforms as well.

6.5 DBlink

Dblink is a distributed end-to-end Bayesian Entity Resolution Spark package. It provides a framework for answering probabilistic queries about entity membership[14].

Pros: Dblink supports as stated above, answering probabilistic queries about entity membership. Aspects of machine explaining can also be seen as diagnostic summaries of the clustering configurations are saved to a csv format for analysis.

Cons: The idea of Bayesian Entity Resolution seems to be an original approach proposed by dblink for entity resolution. It also seems to be the only technique they employ so it is worth doing a comparison of this open source technique with other techniques being used that might be more tried and tested. It also uses Scala as its language which is something to note.

6.6 SparkER

SparkER was created for Spark Apache, and introduces meta-blocking techniques which allows for fewer number of comparisons than normal Big Data blocking techniques[15].

Pros: The meta-blocking technique aims to increase efficiency compared to standard blocking techniques. It is built out in Scala for Apache Spark, which is a data processing framework.

Cons: This seems to only be a meta-blocking open source software. It is not a fully built out ER system, but rather just a technique and a platform would need to be built around it.

6.7 An Assessment of Proprietary Solutions

We also assessed a couple of proprietary solutions as part of our research, so that we could use those as a benchmark for our open source findings. Senzing is a solution that claims that in addition to efficient batch processing compared to its competitors, they can also accommodate for stream (incremental) processing[16]. DataWalk is a platform that uses a graph-based solution and has great visualization techniques. DataWalk is already being used by government clients and advertises intelligence analysis as one of its use cases. The platform is also extensible as they allow the user to build and add their own modules[17].

7 Next Steps/Future Considerations

There are two additional considerations we would like to highlight. Because of the classified nature of intelligence gathering our team was not able to evaluate how the Novetta entity resolution solution meets both top down and bottom up needs along the chain of command. We think the following question needs to be answered: What are the questions analysts are being asked to answer and how does the current entity resolution solution meet the top down and bottom up needs? Secondly, there are 18 different agencies that make up the U.S. intelligence community. During our customer discovery we found that at least two of the agencies have entity resolution solutions. We think it would be helpful to understand how other agencies are solving the same problem. We unfortunately were not able to get very far down this path without a security clearance.

8 Conclusions

Our team has presented a set of 5 evaluation criteria and evaluated 6 open source solutions. We presented our learnings from our customer discovery interviews, which includes how GAP currently conducts entity

resolution as well as the different dual use cases, such as in entertainment, intelligence community, and immunization reporting industries. Through our own research, we learned about the variety of ways entity resolution can be approached in terms of techniques, and we summarized a few in our technical overview. Finally, we offered proposed solutions which we feel are worth looking into, as well as future considerations moving forward. As a final note, becoming an expert in entity resolution takes time. Entity resolution is a well studied field with many approaches and contributions from a variety of fields. For just the blocking phase of entity resolution, one paper we reviewed defined 29 different blocking approaches. Therefore our survey of open source solutions is by no means an exhaustive search and serves as a starting point for GAP data scientists to begin a discussion on building a proof of concept.

9 Acknowledgements

All authors contributed equally. We thank Prof. Clifford Neuman (USC), Farzin Sabadani (USC), Tai Sunnanon (NSIN), Jesse Gipe (NSIN), and Sam Bernstein (Problem Sponsor - GAP) for the support throughout the semester that lead to this manuscript.

References

- [1] Interview with D. Hanson, Entity resolution engineer for Novetta .
- [2] G. Papadakis, E. Ioannou, E. andThanos, and T. Palpanas, The four generations of entity resolution, *Synthesis Lectures on Data Management* **16**, 1 (2021).
- [3] S. E. Whang and H. Garcia-Molina, Incremental entity resolution on rules and data, *The VLDB Journal* **23**, 77 (2014).
- [4] Using a graph database for big data entity resolution, <https://www.tigergraph.com/blog/using-a-graph-database-for-big-data-entity-resolution/> .
- [5] Interview with H. V. Nguyen, Three letter intelligence agency .
- [6] JedAI web-page, <https://jedai.scify.org/> .
- [7] JedAI GitHub repository, <https://github.com/scify/JedAIToolkit> .
- [8] G. Papadakis, L. Tsekouras, E. Thanos, G. Giannakopoulos, T. Palpanas, and M. Koubarakis, The return of JedAI: End-to-end entity resolution for structured and semi-structured data, *Proc. VLDB Endow.* **11**, 1950–1953 (2018).
- [9] G Papadakis, D Skoutas, E Thanos, and T Palpanas, Blocking and filtering techniques for entity resolution: A survey, *ACM Comput. Surv.* **53** (2020).
- [10] USC ISI RLTK GitHub repository, <https://github.com/usc-isi-i2/rltk> .
- [11] FAMER web-page, https://dbs.uni-leipzig.de/research/projects/object_matching/famer .
- [12] FAMER GitHub repository, <https://git.informatik.uni-leipzig.de/dbs/FAMER/> .
- [13] Zingg GitHub repository, <https://github.com/zinggAI/zingg> .
- [14] DBlink GitHub repository, <https://github.com/cleanzr/dblink> .
- [15] SparkER GitHub repository, <https://github.com/Gaglia88/sparker> .
- [16] Senzing web-page, <https://senzing.com/> .
- [17] Datawalk web-page, <https://datawalk.com/> .