A Pattern-Based Approach to Name and Address Parsing with Active Learning

Onais Khan Mohammed, Khizer Syed, John Talburt, Adeeba Tarannum,
Abdul Kareem Khan Kashif, Salman Khan, Najmudin Syed and Syed Yaser Mehdi
Department of Information Quality, University of Arkansas at Little Rock,
2801 S. University Ave., Little Rock, AR, U.S.A.

Keywords: Address Parsing, Active Learning, Data Standardization, Entity Resolution, Pattern Recognition, Knowledge Graphs, Tokenization, US Postal Standards, Machine Learning Models in Parsing, Semantic Data Integration.

Abstract: Processing population data often requires parsing demographic items into a standard set of fields to achieve metadata alignment. This paper describes a novel approach based on token pattern mappings augmented by active learning. Input strings are tokenized and a token mask is created by replacing each token with a single-character code indicating the token's potential function in the input string. A user-created mapping then directs each token represented in the mask to its correct functional category. Testing has shown the system to be as accurate as, and in some cases, more accurate than comparable parsing systems. The primary advantage of this approach over other systems is that it allows a user to easily add a new mapping when an input does not conform to any previously encoded mappings instead of having to reprogram system parsing rules or retrain a supervised parsing machine learning model.

1 INTRODUCTION

Many systems designed to process multiple sources of the same information require each source to define the same attributes for essential parts of the record (Mohammed, Mahmood, 2022). This is especially true when improving the quality of record linkage, entity resolution, and data integration processes. For example, one source may have a single field containing the entire postal address of a person or business whereas another source may separate the street address from the city name and yet another even may have separate fields for the components of the street address such as street number and street name. When different sources define attributes in different ways, the process to standardize the attributes is called metadata alignment. The most common approach to standardization of attributes is to parse (decompose) less structured attributes into the same set of fundamental components (Elhamifar, E., Sapiro, G., Yang, A., & Sasrty, S. S., 2013). For population (name and address) data these fundamental components, while not universally standardized, are generally well understood. These components may vary by region or country, however. at its core, the US Address Data Preparation Function is designed to take an unstructured address string and break it down into a series of meaningful components. This includes identifying and separating the street address, city, state, and postal code. These components are then stored in a structured format, allowing them to be easily retrieved and used in various applications.

The use of HiPER indices, Boolean rules, and scoring rules is one of the key benefits of the US Address Data Preparation Function. HiPER indices are used to search and retrieve data from large data sets and are essential for high-performance data processing applications. Boolean rules are used to determine the validity of data, and scoring rules are used to determine the relevance of data. (M. Mohammed, 2021) These features allow for the efficient processing of large amounts of data and ensure that only accurate and relevant data is used in applications. In addition to the US Address Data Preparation Function, there are other data preparation functions that require the definition of new XML elements in the Entities Script. These functions are designed to perform specific tasks, such as removing duplicates, reformatting data, and transforming data into a standardized format.

Mohammed, O. K., Syed, K., Talburt, J., Tarannum, A., Kashif, A. K. K., Khan, S., Syed, N. and Mehdi, S. Y.
A Pattern-Based Approach to Name and Address Parsing with Active Learning.
DOI: 10.5220/0013077500003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 70-77
ISBN: 978-989-758-737-5; ISSN: 2184-433X

In the context of record linkage, address parsing is a critical step due to the ubiquity of address fields in databases (Churches, T, 2002) This process involves segmenting raw addresses into semantic fields, which is essential for identifying records referring to the same entity across different data collections. The main challenge in address parsing arises from the variability and inconsistency in address formats, including differences in field orders and the presence of errors. (Sorokine, A., Kaufman, 2002) The evaluation of address parsing accuracy is focused on the correct assignment of each element of an address string into its appropriate field, with individual fields like 'Street' being assessed separately. Traditional rule-based parsing methods often struggle with the complexity of real-world addresses, leading to a growing preference for flexible, learning-based approaches, such as machine learning models (Elhamifar, 2013). Overall, effective address parsing is crucial for ensuring high-quality data in record linkage, especially given the diverse and imperfect nature of address data encountered in various real-world scenarios. Dictionary-based address parsing uses the concept of string matching, where input address strings are compared against a pre-compiled dictionary containing known addresses or key address components. This method is straightforward, involving matching segments of an address with dictionary entries to accurately identify and classify address components.

The effectiveness of rule-based approaches is particularly pronounced with standard and well-known address formats. Rule-based address parsing adopts a structured approach, utilizing predefined rules reflective of common address formats and components. This method focuses on developing techniques based on the unique traits of address components and their organizational structure. Parsing algorithms under this approach are designed to recognize and interpret various address elements, understanding their hierarchical and syntactical relationships. This method's flexibility allows it to adapt to a variety of address structures, including complex and non-standard formats. However, it requires a detailed understanding of address formatting rules and can be more complex to implement.

2 METHODOLOGY

The system has two major components, the automated parsing system, and a manual exception handling system. In the method described here, the system assumes that the input comprises only name and address words and that the name words precede the address words. The automated parsing system has three major sub-components. The first component reads the name and address input, parses it into tokens, removes some punctuation, uppercases the tokens, then creates a token mask by assigning each token a character based on a small lookup table of clue words.

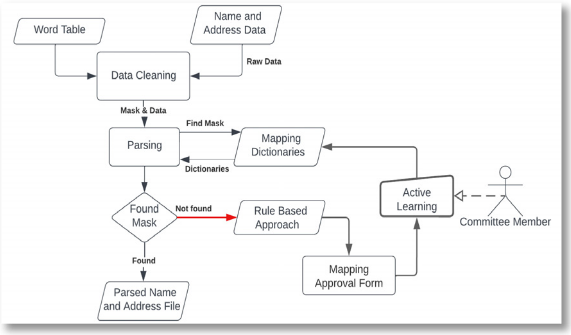

Figure 1: Flow Diagram of the name and address parsing using active learning.

2.1 Parsing Module

Note some punctuation, especially commas and hyphens, are kept as important pattern clues. If a token is not found in the lookup table it is classified as either a "W" token type if it starts with a letter, or an "N" token type if it starts with a digit. Based on an analysis of the mask, the system divides the input tokens into two groups, name tokens and address tokens. The second sub-component processes the name tokens. In this step, the name tokens are used to generate a name token mask using a different clue word table. The name clues identify tokens commonly associated with the five name word categories including prefix titles such as "MR" and "MRS", common given and family name, generational suffixes, and suffix titles such as "PHD" and "CPA". As before, tokens not found are assigned either "W" token type or "N" token type. If the resulting name mask is found, then each token is mapped to its appropriate name word category.

It is important to note that the token mappings are created by domain experts, not automatically assigned by token type. The expert can interpret the overall meaning of the mask's pattern. For example, the token "JUNIOR" is tagged in the name clue table by its most likely use as a generation suffix token, a type "J" token. However, it could also occur as a given name.

For example, assume that "JOHN" is in the table as a common given name of "G" token type, and that

"DOE" is not in the table. The name "JOHN DOE, JUNIOR" would generate the mask "GW,J" which would be mapped by the domain expert as "G" to the given name attribute, "DOE" to the surname attribute, and "J' to the generational suffix attribute. On the other hand, the name "JUNIOR DOE" while generating the mask "JW" would be mapped by the domain expert as "JUNIOR" to the given name attribute and "DOE" to the surname attribute despite "J" indicating a generational suffix.

The third sub-component processes the address tokens following a similar scheme. This component has a separate address-specific clue word table. In addition, there are currently 6 address token types used to identify 15 address word categories as shown in Table 1. So, for example the "D" token type identifies a directional token such as "N" for north and "SE" for southeast.

However, the token identified as type "D" may be used as a predicational address attribute, e.g., "N OAK ST" or a post directional address attribute, e.g., "E ST NORTH". These examples again illustrate the need for pattern interpretation of the mask by a domain expert.

In the street address "N OAK ST", the "N" token is identified as a "D" type token and would be mapped to the predicational attribute. However, in "E ST NORTH", both the "E" token and the "NORTH" token would be identified as "D" type tokens, the "E" token would be mapped to the street name attribute, and the token "NORTH" would be mapped to the post directional attribute.

For both the name parsing and the address parsing to successfully complete, the token mask generated by the lookup table process must be found in a knowledge base of previously created mask mappings. If either the name mask or address mask is not found, then the parsing operation fails for that input and the input and mask are both written to an exceptions file that is the input to the Exception Processing System, This mask mapping entry is then inserted into the mask-mapping knowledge base so that thereafter, any input generating the same mask will be automatically processed by the automated parsing system when a match for a token is not found in the pre-defined dictionary of masks. This step is important because not all addresses will conform to the standard format used in the dictionary, and some addresses may contain non-standard or ambiguous components that cannot be matched to a specific address field.

(i) Searching for a Match:
When a token is generated and the corresponding mask does not match any of the pre-defined masks in the dictionary, the program will search for a match by comparing the token to a list of common address components. This list may include common street names, city names, and state abbreviations it also uses the Levenshtein similarity scores to to find the closest match for example Junior and 'Junir' wherein we have a missing 'o' can help us achieve a robust mechanism to assign token which is the epitome of correctness.

(ii) Assigning to an Exception:
If a match is still not found after searching the list of common components, the program will assign the token to an exception. This is a catch-all category that represents any component of the address that cannot be matched to a specific field. Examples of exceptions may include apartment or suite numbers, building names, or unusual address formats.

(iii) Adding the Exception to the US Address:
Once the token has been assigned to an exception, the program will add it to the US address components as a separate field. This allows the exception to be included in the final output, even if it cannot be matched to a specific address field.

The fifth step of the program involves adding the address tokens that were generated in the second step to the US address components. This step builds on the previous steps by combining the cleaned and tokenized address components with the assigned address fields to create a complete US address.

(i) Assigning Tokens to Address Fields:
Based on the comparison of masks in the third step, the program assigns each token to a specific address field, such as the street name, city, state, and zip code. This creates a set of address fields, where each field corresponds to a specific component of the address.

(ii) Adding Address Tokens to the Address Fields:
Once the tokens have been assigned to the address fields, the program then adds these tokens to the appropriate address field in the US address components. For example, if the token "Main" was assigned to the street name field, the program would add "Main" to the street name component in the US address.

(iii) Building the Complete US Address:
Once all tokens have been added to the appropriate address fields, the program can then combine these components to create a complete US address. This involves concatenating the address components in the correct order, and separating them with the appropriate punctuation, such as commas and spaces.

2.2 Annotated Dataset

We have tested our program by giving inputs like attention line addresses, individual addresses, highway addresses, university addresses, P.O. Box addresses, Puerto Rico addresses and individual names and generated truth files respectively. The focus of our program is to generate as much of Mask to Dictionaries, and that too by ensuring Data Quality. A truth file is a reference file that contains known or verified names and addresses. The parser uses this file to compare the input name or address to the known or verified names and addresses, helping to identify and correct errors in the input data.

Table 1: An example of address parsing using active learning.

Pos	Token	Code	Dictionaries
1	123-1/2	N	USAD_SNO
2	N	D	USAD_SPR
3	OAK	W	USAD_SNM
4	STREET	F	USAD_SFX
5	APT	S	USAD_ANM
6	3A	N	USAD_ANO
7	LITTLE	W	USAD_CTY
8	ROCK	W	USAD_CTY
9	ARK	T	USAD_STA
10	72203-4352	N	USAD_ZIP

Parsing Puerto Rico addresses presented significant challenges, particularly due to the inclusion of Spanish street names, suffixes, and regional dialects. For example, interpreting street names like 'Callejón' or suffixes such as 'Esq.' (Esquina, meaning corner) proved to be particularly difficult. The complexity of these language-specific elements often led to errors in address processing.

However, by employing active learning techniques alongside a non-exhaustive token table populated with contextual clue words, we were able to overcome these obstacles. This method significantly improved our ability to accurately parse and process these complex addresses, demonstrating the effectiveness of our approach.

The approach of tokenizing addresses and mapping tokens to specific categories like USAD_SNO, USAD_CTY, or USAD_ZIP as shown in Table 1 plays a vital role in standardizing and systematizing the parsing process. In this methodology, once a token pattern has been identified and mapped, the same pattern can be automatically recognized and categorized in future inputs without requiring manual intervention each time. This capability is particularly advantageous in applications that deal with large volumes of address data, ensuring consistency and efficiency

Table 2: An example of name parsing using active learning.

Pos	Token	Code	Dictionaries
1	DR	P	PREFIX_TITLE
2	JOHN	G	FIRST_NAME
3	TABLURT	L	SURNAME
4	JR	J	GEN_SUFFIX
5	PHD	Q	SUFFIX_TITLE

For instance, when the pattern "123-1/2 N OAK STREET APT 3A LITTLE ROCK ARK 72203-4352" is first processed, it involves manual or semi-automated categorization where each segment of the address is assigned a specific dictionary based on its identified role (numeric, directional, word, etc.). After this initial classification, the system stores these mappings in a knowledge base.

When a new address comes in that matches a previously encountered pattern, such as another entry starting with a similar structured numeric street number followed by a directional indicator, the system automatically applies the same categorization rules. It recognizes that "123-1/2" should be classified under USAD_SNO and "N" under USAD_SPR, and so forth, based on the stored mappings. This pattern recognition not only accelerates the parsing process but also enhances accuracy by applying proven rules to new data.

In the provided example for name parsing in Table 2, the tokenization process for a personal name "Dr. John Tablurt Jr. IQCP" involves breaking down the name into discrete elements and categorizing each part using a specific code linked to a predefined dictionary. The sequence starts with "DR," positioned as the first token and categorized under the code 'P' for PREFIX_TITLE, recognizing titles or honorifics.

Following this, 'JOHN,' the first or given name, is assigned the code 'G' and classified under FIRST_NAME, which is vital for personal identification. 'TABLURT' functions as the surname or last name, coded as 'L' and categorized under SURNAME, playing a key role in family identification. 'JR,' a generational suffix that indicates lineage, is given the code 'J' and falls under GENERATIONAL_SUFFIX. Lastly, 'PHD,' representing a professional or academic qualification, is coded as 'Q' and grouped under SUFFIX_TITLE.

Now, consider a scenario where the last name is missing. The address parser, using context clues and the code 'Q,' can intelligently determine that 'IQCP' should not be interpreted as a last name but rather as a suffix title. This structured categorization is essential for accurately processing and analyzing each name component for various administrative and data management purposes.

Table 3: Comparison of Active Learning and USaddress Performance on Puerto Rican Address Parsing.

Component	Active Learning	USAddress
Address Number	1000	1000
Street Name	PONCE DE LEON	Avenida Ponce de Leon
Street Suffix	AVENIDA	N/A
Occupancy Type	STE	Ste
Occupancy Identifier	5	5
City Name	SAN JUAN	San Juan
State Name	PR	PR
Zip Code	907	907

3 RESULTS

We have conducted a comparison between the results of an active learning approach and the usaddress Python library, which employs a rule-based method to parse U.S. addresses. This comparison highlights the effectiveness of active learning, particularly in handling complex addresses such as those found in Puerto Rico.

The active learning method demonstrated superior performance by accurately parsing each component of the address, including the differentiation between the street name, suffix, and occupancy type. For example, while the usaddress library struggled to parse certain elements like the street suffix ("AVENIDA"), the active learning model correctly identified and categorized it.

This is particularly important in addresses from Puerto Rico, where non-standard formats and Spanish language elements often pose challenges for rule-based parsers.

Table 4: Precision and Recall metrics: Active Learning Vs Rule-Based System.

Index	P-Active Learning	P-Rule-Based	R-Active Learning	R-Rule - Based
USAD_SNO	0.9914	0.9167	1	0.6
USAD_SPR	0.9231	0	1	0
USAD_SNM	0.9854	0.6329	1	0.2924
USAD_SFX	0.985	0.6912	1	0.3672
USAD_CTY	0.9931	0.5165	0.9797	0.5639
USAD_STA	0.9932	0.7748	1	0.5911
USAD_ZIP	0.9968	0.8341	0.9968	0.5981
USAD_ORG	1	0	0.9796	0
USAD_MGN	1	0	1	0
USAD_HNO	1	0.7123	0.9859	0.7324
USAD_ANM	1	1	0.9892	0.0114
USAD_MDG	1	0	1	0
USAD_HNM	1	0.7692	1	1
USAD_SPT	0.9231	0	1	0
USAD_RNM	0.98	0.96	1	0.9796
USAD_BNO	1	1	0.9864	0.9592
USAD_ANO	1	0.44	0.9892	0.1264
USAD_RNO	0.9804	0.875	1	0.98
Overall Accuracy	0.9934	0.4755	-	-
Micro Average	0.9934	0.7586	0.9934	0.4755
Weighted Average	0.9935	0.6046	0.9934	0.4755

In this evaluation, we compare the performance of two approaches—Active Learning and a Rule-Based System—across various address components, as defined by the USAD Conversion Dictionary. This dictionary maps human-readable address parts to standardized USAD codes, including components like street number (USAD_SNO), city (USAD_CTY), and organization (USAD_ORG). Active Learning demonstrated significantly better performance across all metrics. For instance, it achieved near-perfect precision, recall, and F1-score for street numbers (USAD_SNO), with values of 0.9914, 1.0000, and 0.9957, respectively. In contrast, the Rule-Based System struggled with a precision of 0.9167 and a recall of 0.6000, resulting in an F1-score of 0.7253. This indicates that the rule-based approach could not adequately capture the variations in street numbers, likely due to rigid or incomplete rule definitions.

For pre-directionals (USAD_SPR), the discrepancy was even more pronounced. The Active Learning model managed a perfect recall and a precision of 0.9231, leading to an F1-score of 0.9600. Meanwhile, the Rule-Based System completely failed in this category, with all metrics at zero, suggesting a lack of adequate rules for recognizing pre directionals. The street name (USAD_SNM) component also showed a clear advantage for Active Learning, with an F1-score of 0.9926 compared to 0.4000 for the Rule-Based System. The latter's performance was hindered by a low recall of 0.2924 and a precision of 0.6329, indicating that it could neither consistently identify nor correctly label street names.

Similarly, in identifying city names (USAD_CTY), the Active Learning approach exhibited high precision and recall, achieving a F1-score of 0.9863. The Rule-Based System lagged significantly, with metrics indicating a much lower accuracy in parsing this component. Overall, the Active Learning method

4 CONCLUSION

In this paper, we have presented a method for address parsing that employs an active learning approach, particularly effective for handling complex cases such as Puerto Rican addresses. It is important to clarify that our use of "active learning" refers to an interactive, user-driven process rather than a machine learning-based approach. Our system is primarily dictionary-based, relying on a robust set of predefined rules and dictionaries to parse addresses. What sets our approach apart is the integration of user feedback to handle exceptions and edge cases. When the system encounters an address that doesn't fit the existing rules, it prompts the user to provide feedback. This feedback is used to refine the rules and update the dictionaries, allowing the system to adapt to new or unusual address formats.

Rather than employing complex machine learning algorithms, our system uses simple distance metrics to identify potential matches and discrepancies within the address components. The feedback loop is crucial here—by continually refining the parsing rules based on user input, the system becomes more accurate over time, even without the use of machine learning. This iterative process enables the system to handle a wide variety of address formats, including those that deviate from standard patterns, making it both flexible and reliable.

Active learning offers significant advantages over traditional machine learning models, particularly in scenarios that require the flexibility to make incremental improvements without the need to overhaul the entire dataset or system rules. This characteristic is especially beneficial in parsing systems, like those used for processing names and addresses, where variability and exceptions are common. In traditional deterministic or machine learning models, if an error is identified, addressing this error typically requires retraining the model or revising the entire rule set. This process can be time-consuming and resource-intensive, as it may involve recalibrating the model parameters or rewriting rules based on the new data.

Moreover, such adjustments risk affecting the performance on other parts of the dataset that were previously correct, a phenomenon known as the "stability-plasticity dilemma." In contrast, active learning allows for more targeted interventions. When an error is detected in active learning systems, only the specific part of the system—such as a particular mask or dictionary entry associated with the error—needs to be adjusted. This is done by either updating the existing entries or adding new ones to the dictionary to handle the exception. This selective updating preserves the integrity and accuracy of the other parts of the system, ensuring that previous learning and correct mappings are not disturbed by changes made to address new or outlier cases.

This method enhances the system's adaptability and efficiency, enabling it to improve continuously as it processes new information without requiring comprehensive retraining or global rule adjustments. Such a model is not only more scalable but also more maintainable, as it allows for fine-grained improvements that directly address specific issues or adapt to new types of data encountered in operational environments. Active learning thus provides a practical approach in dynamic settings where data can change frequently, or new patterns emerge over time, making it a superior choice for applications that benefit from ongoing learning and adaptation without the need for frequent, broad-scale modifications.

Table 5: Address Parsing Improvements with Active Learning vs. Rule-Based System.

Use Case	Input Example	Rule-Based Parsing Result	Active Learning Parsing Result
Puerto Rican Address Parsing	"1000 Avenida Juan Ponce de León, Apt 3C, San Juan, PR 00907"	Struggles with stName vs. suffix, misclassified "Avenida," and incorrectly parses apt. number.	Correctly identifies "Avenida Juan Ponce de León" as the street name and accurately parses "Apt 3C."
Puerto Rican Address with Directional Component	"123 Calle de la Rosa Oeste, Apt 2B, Guaynabo, PR 00966"	Will misinterpret "Oeste" (West) as part of the street name instead of a directional suffix.	Correctly identifies "Calle de la Rosa" as the street name and "Oeste" as the directional suffix, accurately parsing "Apt 2B."
Pre-directional and Post-directional Components	"123 North East Street Drive, Little Rock, AR 30003"	Will classify "North" as the only pre-directional and "East Street" as the street name, failing to recognize "Street" as a suffix.	Correctly identifies "North East" as a combined pre-directional ("NE"), "Street" as a suffix, and "Drive" as a post-directional.
Complex Urban Address	"456 West Market Street Plaza, Chicago, IL 60605"	Struggles to differentiate between "Market Street" and "Plaza," leading to incorrect parsing.	Successfully identifies "West Market Street" as the street name and "Plaza" as the post-directional suffix.
Puerto Rican Address with Multiple Components	"789 Calle San Francisco, Piso 4, Oficina 12, Old San Juan, PR 00901"	Will incorrectly parse "Piso 4" (Floor 4) and "Oficina 12" (Office 12), potentially merging these with other components.	Accurately identifies "Calle San Francisco" as the street name, and correctly parses "Piso 4" and "Oficina 12" as distinct occupancy identifiers.

ACKNOWLEDGEMENTS

This research was partially supported by the Department of Commerce US Census Cooperative Agreement CB21RMD0160002. We express our profound gratitude to Dr. John R. Talburt, the Project Coordinator. His invaluable guidance and support have been instrumental in the progression and success of this project. We also extend our thanks to Syed Yaser Mehdi for his comprehensive research on the literature review which has significantly enriched our understanding and findings. We are deeply grateful to the University of Arkansas at Little Rock for providing us with the necessary resources and an inspiring platform to conduct our research. Their commitment to fostering an environment of academic excellence and innovation has been crucial to our work. Additionally, we would like to acknowledge the US Census Bureau for sponsoring this project. Their collaboration and the data provided have been pivotal in shaping the research outcomes. This project not only benefited from the contributions mentioned above but also reflected the collaborative spirit and the shared goal of advancing knowledge in the field of Information Quality.

REFERENCES

Evaluation of Automation Techniques for Data Quality Assessment for Party and Product Master Data. Mohammed, Mahmood. University of Arkansas at Little Rock ProQuest Dissertations Publishing, 2022. 29206326.

Churches, T., Christen, P., Lim, K. *et al.* Preparation of name and address data for record linkage using hidden Markov models. *BMC Med Inform Decis Mak* 2, 9 (2002). https://doi.org/10.1186/1472-6947-2-9

Sorokine, A., Kaufman, J., Piburn, J., & Stewart, R. (2020). *Active Learning Approach to Record Linking in Large Geodatasets*. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Elhamifar, E., Sapiro, G., Yang, A., & Sasrty, S. S. (2013). A convex optimization framework for active learning. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 209-216).

Hogan, H., Cantwell, P. J., Devine, J., Mule, V. T., and Velkoff, V. "Quality and the 2010 Census." Population Research and Policy Review, 32(5):637–662 (2013).

Qian, K., Popa, L., & Sen, P. (2017, November). Active learning for large-scale entity resolution. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 1379-1388).

Li, X., Kardes, H., Wang, X., & Sun, A. (2014, November). Hmm-based address parsing with massive synthetic training data generation. In *Proceedings of the 4th International Workshop on Location and the Web* (pp. 33-36).

Li, X., Talburt, J. R., & Li, T. (2018, December). Scoring matrix for unstandardized data in entity resolution. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 1087-1092). IEEE.

Maddodi, S., Attigeri, G. V., & Karunakar, A. K. (2010, November). Data deduplication techniques and analysis. In *2010 3rd International Conference on Emerging Trends in Engineering and Technology* (pp. 664-668). IEEE.

Meduri, V. V., Popa, L., Sen, P., & Sarwat, M. (2020, June). A comprehensive benchmark framework for active learning methods in entity matching. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 1133-1147).

Nafa, Y., Chen, Q., Chen, Z., Lu, X., He, H., Duan, T., & Li, Z. (2022). Active deep learning on entity resolution by risk sampling. *Knowledge-Based Systems*, *236*, 107729.

Simonini, G., Zecchini, L., Bergamaschi, S., & Felix, N. (2022). Entity Resolution On-Demand.

Tu, J., Han, X., Fan, J., Tang, N., Chai, C., Li, G., & Du, X. (2022). DADER: hands-off entity resolution with domain adaptation. *Proceedings of the VLDB Endowment*, *15*(12), 3666-3669.

Ye, Y., & Talburt, J. R. (2019). Generating synthetic data to support entity resolution education and research. *Journal of Computing Sciences in Colleges*, *34*(7), 12-19.

M. Mohammed, J. R. Talburt, S. Dagtas and M. Hollingsworth, "*A Zero Trust Model Based Framework For Data Quality Assessment*," 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2021, pp. 305-307, doi: 10.1109/CSCI54926.2021.00123.