International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 9-No.4 (2023) pp. 468-474
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Survey Data Engineering: Normalizing and Automating Semi-Structured Data Pipelines

Prateek Panigrahy*

KIIT University, Bhubaneswar, Odisha
* **Corresponding Author Email:** prateek.panigrahy@gmail.com -**ORCID:** 0009-0002-6992-5653

Article Info:

DOI: 10.22399/ijcesen.4895
Received : 05 November 2023
Accepted : 30 December 2023

Keywords

Survey Data Engineering;
Semi-Structured Data;
Data Pipeline Automation;
Data Normalization;
Schema Drift

Abstract:

The resulting enormous size of data gathering through surveys in the Web space has been a significant engineering problem in developing scalable and reliable data pipes. The heterogeneous forms, nested forms, language variations, and changing schema are comprised in survey data that is typically semi-structured. This is particularly necessary when there is a creation of data within various tools and in various languages because it is relevant to normalize this data to generate a standard structure through which the data can be analyzed to obtain actionable information. The paper explains the meaning and approach of survey data engineering as it relates to schema reconciliation, metadata parsing, and linguistic standardization as the most critical areas of normalization. Besides that, it also elaborates on how to automate end-to-end pipelines using modular architectures, real-time orchestration, and cloud-native technologies. The issues, such as schema drift, multilingual inconsistency, data quality issues, and compliance requirements, are also addressed. This paper is then concluded with the best practices to develop resilient, repeatable, and scalable survey data workflows since automation may be strategic in modern data-based environments.

1. Introduction

Survey-based data has become a useful asset in the new digital space in terms of decision-making regarding a broad field, including healthcare, marketing, social sciences, policymaking, and business analytics. Survey data can never be structured like a conventional structured database with respect to the semi-structured survey data, which are collected with the help of a wide variety of tools such as online forms, feedback portals, telephone interviews, and field-based data collection forms. These types of data are not uniform and multidimensional, and it is this aspect that makes it a special task for the engineers and data scientists to convert incongruent survey data to standard data that can be analyzed [1][2][3]. With the increase in the volume of data-based operations in organizations, there is an increased volume of semi-structured survey inputs. This adds up and renders the manual processing procedures ineffective, fallible, and unscaled. It has, therefore, been facing an acute need to automate the normalization and integration of semi-structured data on surveys to pipelines to have quality, validated, and reusable data [4][5]. It is an intricate task of data analysis, cleaning, transferring, and loading into the data analysis or data storage systems. Some of the problems that need advanced engineering practices to overcome include nested formats of the JSON, irregular metadata, multi-linguistic input, and the linguistic difference of schema to a location or to a collection instrument [6][7].Survey engineering Data normalization is the job of transforming data of different formats and putting them into a single model and preserving content and semantic relevance. The focus of this process lies with the application of automation that has minimized human intervention, field mapping consistency, validation rules, and lifecycle management of data. Making automated survey data pipelines is, however, not an easy thing; it requires the combination of not just computational intelligence, but also domain knowledge, and infrastructural flexibility [8][9]. It is against this backdrop that this paper aims to research the new subject of survey data engineering, in particular, perspective normalization operation and automation plan of the semi-structured data pipelines. Starting with the background knowledge of what survey

data is, one will continue with structural and semantic issues of normalization, the architectural models, and automation technologies that will help in coordinating the pipeline. Each of the sections is connected to the rest in a way that makes sense to the reader as they develop a flow of the engineering concepts and technological solutions that form the foundations of the current survey data management.

2. Understanding the Nature of Survey Data

The analysis that the data in the survey should be normalized and automated requires the analysis of the inherent nature of survey data and its semi-structured character. Survey data usually entails a combination of closed-ended preset answers, as well as open-ended text answers, where each has different structural conventions. It is exported to tabular forms of applications like Google Forms, Qualtrics, and SurveyMonkey, and metadata layers use a data format of either JSON or XML that can be nestable or various names in the field depending on the configuration [10][11]. This heterogeneity is caused by the combination of different aspects: the different kinds of survey questions (e.g., single-choice and multiple-choice, and Likert scale), regional adaptation, and a display format based on devices. Practically, even the result of the same survey template can be given in varying structural forms in various deployments, so that, unless previously normalized, the applications of generalized logic of parsing or transformation can be challenging [12][13].The field name can also be ambiguous, as well as in the surveys (e.g., Q1, Q2), which have no semantic meaning until it is decoded into question text. The free-text responses are even worse as they do not contain any structure, and they require some pre-processing in terms of tokenization, language recognition, and syntactic normalization. Also, time-series items, respondent metadata (e.g., location, type of device), and skip logic make the formulation of a consistent data model of raw survey exports more complex [14][15]. The effects of a loosely designed data pipeline system when conducting a survey are enormous. This may be because of incorrect mappings, context, or invalid transformations that all lead to degradation of the integrity of information-driven insights. This fact is what clarifies the need to have a systematic process of engineering raw survey data into clean, structured, and semantically useful data, which can be further consumed downstream by business intelligence software, machine learning software, or regulatory reports [16][17]. The knowledge about the various sources and forms of the survey data was the background that was required to discuss the process of normalization. The second section includes the challenge of software to convert heterogeneous survey data into a single interoperable format that has an analytic value and variability of the inputs as minimal as possible.

3. Normalization of Semi-Structured Survey Data

Normalization serves as a crucial step in transforming semi-structured forms into a homogeneous schema that can be efficiently processed, stored, and analyzed, as illustrated in Figure 1. In this context, it focuses on cleaning and correcting information while also addressing inconsistencies, integrating disparate data, and logically structuring it without compromising the semantic integrity of the original responses [18][19].Schema discovery can also provoke normalization, in which the engineers discover similarities and differences between datasets of the same or similar survey instruments. Schema inference, field clustering, metadata parsing using automated tools, and schema inference are some of the ways the candidate mappings are developed. Using these mappings, engineers can outline relationships between raw fields with their normalized counterparts they can express transformation regulations that resolve naming issues, type incompatibilities, and gaps in data [20][21]. Nested data is considered one of the hardest problems of normalization. The questions may be in the form of a matrix, repetition groups, or hierarchies (e.g., child-parent conditional questions) can also be presented in the form of a survey tool, and may be exported as an embedded array format or a JSON structure. That data flattening should be done in a way that maintains the association between native responses and metadata that is a reflection of the wording of the question, condition logic, and order of survey responses [22][23].The other driver is the significance of the other element of language normalization, which is necessary and adopted in other areas, that is, the multilingual surveys. The engineers should standardize the inputs by mapping out a variety of translations of the same question or the answer option to a canonical form. It can either be through natural language processing (NLP) models or rule-based mappings. Such kinds of deviations in encoding, punctuation, or spelling should be equally corrected, especially in free-text responses, such that appropriate downstream processing [24][25] may be performed. Value heterogeneity is also associated with normalization. The polar response may be Yes/No, Y/N, 1/0, or even local ones like si/no in Spanish. Such values

should be standardized to the coherent analytics in the form of a single ontology. Moreover, normalization pipelines must have the capability to coordinate on a time- timestamps of sources, different sources must reflect them in a consistent format and time zone representation [26][27].Version control plays a critical role during the normalization process. Survey designs rarely remain fixed or permanent; they evolve with the addition or modification of fields. As these changes occur, engineers must ensure that the original normalization logic remains consistent and stable, preventing disruptions caused by newly introduced logic. This involves an active pipeline element that would identify a schema change and alter the transformation logic to the change. These schema changes and the implications to previous datasets should be traced properly using documented and validated mechanisms [28][29]. As normalization rules are created, they are formalized into layers of transformation, and they are likely to be in Data Extract, Transform, Load (ETL) pipelines, or in DataOps today. These transformation stages are used to transform the semi-structured input into structured output with given fields, types, and valid values, and referential metadata. Automated survey data pipelines are based on such structured outputs and are discussed in the next section.

4. Automation in Survey Data Pipelines

After semi-structured survey data is normalized, the second important step is the automation of data pipelines that enable the former. Automation is essential for efficiency and scalability as well as the consistency, reproducibility, and auditability of data workflows. With increasing amounts of survey information being processed by organizations using different tools to conduct the survey and in different areas, the data is gathered, and manual interventions serve as a bottleneck and also as a point of risk in data integrity [30][1].The automation process starts with the ingestion stage, whereby the connectors and adapters are created to enable communication with diverse survey platforms via APIs, webhooks, or scheduled exports. These automated ingestion systems identify new entries, which are put forward, and the extraction of the same is done in real time or near real time and fed to the pipeline. This reduces the time delay between the data capture and the data availability, which is of great importance in time-sensitive algorithms like epidemiological supervision or political surveys [2][30]. After ingestion, the data is automatically transformed through transformation layers that apply preconfigured normalization rules that comprise schema alignment, data type coercion, label standardization, and lingo mapping. These transformation engines have typically been used on scalable processing engines like Apache Spark, Airflow, or cloud-native data processing engines like AWS Glue or Azure Data Factory. It should be noted that every stage of transformation is captured, versioned, and in many cases compared to validation suites so that any modification would not mean regression or data inconsistency in the final dataset [5]. Validation and error management mechanisms are also a part of automation. They are mechanisms subject to rules that raise red flags (abnormalities) of missing mandatory fields, invalid values, duplication of records, or time-series violations [7]. Automation can also be applied to a large-scale survey situation for data consolidation by accessing external data sets (e.g., demographic data, geographic mappings, or device metadata) and relating them with survey responses to create fuller and context-aware datasets. These joins will necessitate the identification, alignment, geo-coordinate standardization, and transformation of categorical labels to common taxonomies. The pipeline is fully automated to delve into the process to develop a smooth integration and analytical congruency.Also, automated pipelines control data outputs by scheduled exports, dashboards, and API-based delivery systems, enabling downstream systems or analysts to access the processed data in real-time. These outputs can be stored in relational databases, data lakes, or streaming platforms under the basis on the latency needs and volume. Metadata catalogs are kept in tandem with document data lineage, transformation logic, field descriptions, and versioning history, which is the guarantee of traceability, which is used during audit and compliance. Orchestration is one of the frequently neglected aspects of automation, including the control of pipeline dependencies, the sequence of execution, and failure recovery. Apache Airflow, Dagster, and Prefect are consulting tools used to organize the processes of data collection of surveys to enable engineers to describe complex dependency graphs, track execution metrics, and issue notifications on failures or hitting thresholds. Such orchestration layers are critical in operational resilience, more so in mission-critical deployment, where survey information is employed to make real-time decisions [4]. Automation allows scalability, efficiency, and consistency of the transformation of semi-structured survey data into high-quality datasets. Nonetheless, these pipelines require a planned architectural implementation, and this is discussed in the following section, which explores

some common pipeline models, tooling ecosystems, and deployment policies.

5. Architectural Models and Tools

Building on the principles of normalization and automation, the design of robust architectural models is essential for deploying end-to-end survey data engineering pipelines, as shown in Figure 2. These architectures must be scalable, modular, and resilient, capable of handling various data sources, formats, and processing demands. The architecture is not merely technical; it reflects a strategic alignment between data engineering, governance, and business objectives. At a high level, a typical architecture for a survey data pipeline includes the following layers: data ingestion, staging, processing, storage, and output delivery. Each of these layers is architected with redundancy, monitoring, and transformation logic. The ingestion layer leverages connectors built on REST APIs, FTP, cloud storage events, or message queues. These connectors standardize incoming data streams into staging areas, typically implemented through data lakes or NoSQL stores that accommodate semi-structured formats like JSON, XML, or CSV.The staging layer serves as a buffer for lightweight preprocessing such as schema validation, deduplication, and basic transformation before data moves to the processing layer, the pipeline's core. Here, transformation, normalization, enrichment, and validation are performed using tools like Apache Spark, dbt, or pandas, orchestrated by Apache Airflow or Dagster and deployed via Docker on Kubernetes for scalability [8][9]. Modern architectures emphasize modularity and microservices, where each function (e.g., language normalization, schema transformation, outlier detection) operates as an independent, stateless service with clear APIs. This enhances error handling, testability, and reusability. Event-driven frameworks like Apache Kafka or AWS EventBridge further enable real-time task triggering based on data availability.

On the storage side, architectures typically adopt a multi-tier model:

- A raw zone for storing original exports
- A processed zone for normalized and validated data
- A curated zone for analysis-ready datasets

6. Challenges and Best Practices

An effective survey data pipeline not only pertains to good architectural design but also to feasible strategies to proceed with the real issues of implementation. The most critical one is when schema variability where surveys can vary through the addition of more questions, inversion of sections, or alteration of logic such that it resulting in schema drift that can cause interference in data normalization. To deal with this, the idea of schema versioning and backward compatibility is proposed, so that it provides some structural flexibility, yet at the same time, consistency. The other thorn in the flesh is data quality; the biased responses, conflicting information, and other such problems require automated validation, anomaly detection, and human controls through quality dashboards and feedback. The fact that language peculiarities, character sets, and idiomatic phrases may skew the interpretation of multilingual, multicultural data further complicates the data processing of such data, and to raise the accuracy of the process in any language environment, it is preferable to alternate NLP pipelines, internationalization libraries, and human review in the loop [14][15]. In addition, filling survey data with external data produces schema anomalies and timing errors, which may breach referential integrity. Data contracts and transformation staging layers will ensure that the external data is in the preferred formats, annexing it before it is incorporated [16]. Anyway, good survey pipelines support that balance between automation and flexibility and incorporate monitoring, checking, and improvement procedures that provide reliability, accuracy, and the ability to scale as data format and numerous global inputs vary.Maintaining pipeline performance under load is a key challenge, as survey data systems must process real-time or batch workloads depending on business needs. Growing data volumes and complex transformations, especially with nested or multilingual fields, can create bottlenecks mitigated through horizontal scaling, partitioning, and lazy transformations [28][29]. Security and compliance are critical in regulated sectors like healthcare, finance, and public policy, requiring encryption, role-based access, audit logs, and CI/CD-integrated compliance checks [10][11]. Ensuring traceability and reproducibility demands logging, versioning, and metadata management using configuration files and workflow snapshots [22][23]. Finally, strong collaboration and ownership across teams supported by documentation and SLAs prevent fragmentation, ensuring scalable, secure, and reliable survey data pipelines. Beyond technical hurdles, the successful implementation of survey data pipelines is heavily influenced by organizational and operational maturity. This includes team capabilities, documentation standards, infrastructure readiness, and cross-

functional collaboration. The following table highlights key organizational factors that influence the success or failure of survey data pipeline projects, particularly in environments dealing with semi-structured survey data.

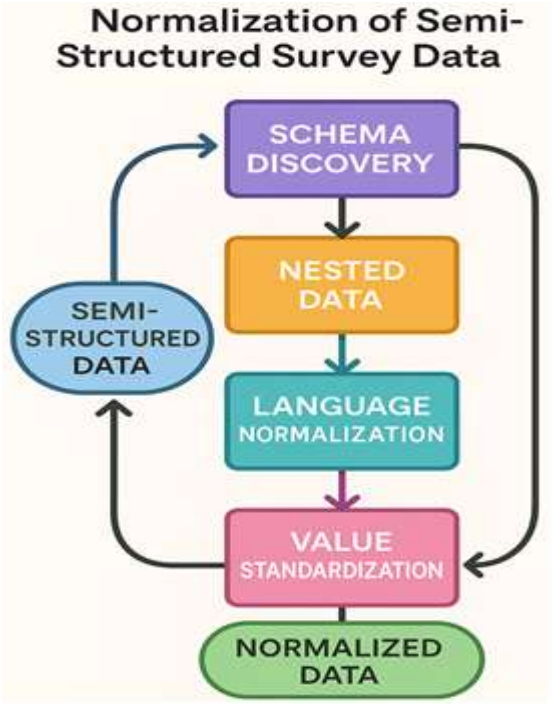

Figure 1: *Workflow for Normalizing Semi-Structured Survey Data*

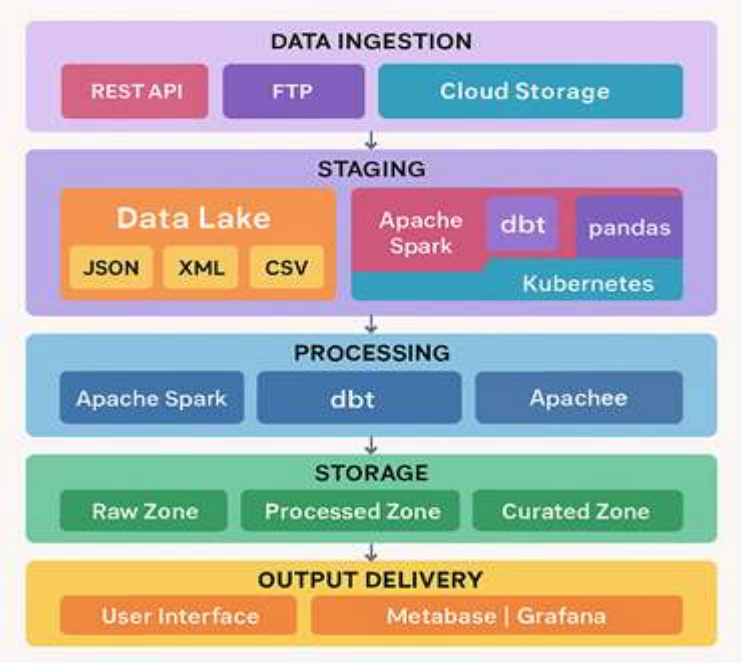

Figure 2: *Layered Architecture of a Survey Data Pipeline*

Table 1: *Organizational Readiness Factors Impacting Survey Data Pipeline Success*

Readiness Factor	Description	Impact on Pipeline Success	Common Gaps Identified
Data Stewardship	Clear ownership of data quality, metadata, and governance responsibilities	Ensures sustained data accuracy and traceability	Lack of defined data owners or stewards
Cross-functional Alignment	Collaboration between survey designers, engineers, analysts, and compliance	Reduces miscommunication and pipeline rework	Silos between data producers and consumers

Readiness Factor	Description	Impact on Pipeline Success	Common Gaps Identified
	teams		
Infrastructure Maturity	Availability of scalable storage, compute, and orchestration tools	Enables automation and high throughput	Reliance on manual tools and unscalable systems
Documentation Discipline	Standardization of schema definitions, transformation logic, and pipeline flows	Facilitates maintainability and reproducibility	Incomplete or outdated pipeline documentation
Change Management Process	Protocols for handling schema updates and version control	Minimizes disruption due to survey design changes	Ad hoc updates with no version tracking
Training and Skills Readiness	Engineering and data literacy across relevant teams	Improves pipeline design and monitoring	Inadequate training on modern tools and best practices

7. Conclusions

The paper has discussed the history of survey data engineering, its primary principles, viz., normalization and automation, as the foundations of the semi-structured data pipelines management. Populable and precise data processing is required because the survey has additional options for capturing human conduct and feelings. Normalization transforms free-text and multilingual and varying-schema data into frequent forms of analysis through mapping the schema, flattening data, and standardizing data. Ingestion, transformation, validation, and delivery can then be automated in real-time, which saves the workload and improves quality. The reliability, observability, and compliance are ensured by the tools such as Apache Airflow, Spark, Kubernetes, and data catalogs that are facilitated by the application of such architectural components as modular microservices, cloud-native deployments, secure data zones, etc. The versioning, anomaly detection, and reproducibility best practices are required to resolve challenges like schema drift, multilingual complexity, and performance bottlenecks. The application of AI-based modifications, self-mending pipes, and cross-lingual mapping will become more effective in the future, but normalization and automation will remain at the focal point during the value extraction of the survey data.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Yuan, G., Lu, J., Yan, Z., & Wu, S. (2023). A survey on mapping semi-structured data and graph data to relational data. *ACM Computing Surveys*, *55*(10), 1-38.

[2] Ahmad, H., Kermanshahani, S., Simonet, A., & Simonet, M. (2009, April). Data Warehouse-Based Approach to the Integration of Semi-structured Data. In *Asia-Pacific Web Conference* (pp. 88-99). Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] Singh, A. (2021). Data science and human behaviour interpretation and transformation. *Journal of Learning and Teaching in Digital Age*, *6*(1), 1-7.

[4] Chowdhury, R. H. (2021). Cloud-Based Data Engineering for Scalable Business Analytics Solutions: Designing Scalable Cloud Architectures to Enhance the Efficiency of Big Data Analytics in Enterprise Settings. *Journal of Technological Science & Engineering (JTSE)*, *2*(1), 21-33.

[5] Doleschal, J., Höllerich, N., Martens, W., & Neven, F. (2018, April). CHISEL: Sculpting tabular and non-tabular data on the web. In *Companion Proceedings of the The Web Conference 2018* (pp. 139-142).

[6] Bergstrom, L., Fluet, M., Rainey, M., Reppy, J., Rosen, S., & Shaw, A. (2013, February). Data-only

flattening for nested data parallelism. In *Proceedings of the 18th ACM SIGPLAN symposium on Principles and practice of parallel programming* (pp. 81-92).

[7] Wen, A., Fu, S., Moon, S., El Wazir, M., Rosenbaum, A., Kaggal, V. C., ... & Fan, J. (2019). Desiderata for delivering NLP to accelerate healthcare AI advancement and a Mayo Clinic NLP-as-a-service implementation. *NPJ digital medicine*, *2*(1), 130.

[8] Tantalaki, N., Souravlas, S., & Roumeliotis, M. (2020). A review on big data real-time stream processing and its scheduling techniques. *International Journal of Parallel, Emergent and Distributed Systems*, *35*(5), 571-601.

[9] Hendler, J. (2014). Data integration for heterogenous datasets. *Big data*, *2*(4), 205-215.

[10] Jiang, J. A., Wade, K., Fiesler, C., & Brubaker, J. R. (2021). Supporting serendipity: Opportunities and challenges for Human-AI Collaboration in qualitative analysis. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW1), 1-23.

[11] Saitwal, H., Qing, D., Jones, S., Bernstam, E. V., Chute, C. G., & Johnson, T. R. (2012). Cross-terminology mapping challenges: a demonstration using medication terminological systems. *Journal of biomedical informatics*, *45*(4), 613-625.

[12] Jones, C. S., Duncan, D. H., Morris, W. K., Robinson, D., & Vesk, P. A. (2022). Using data calibration to reconcile outputs from different survey methods in long-term or large-scale studies. *Environmental Monitoring and Assessment*, *194*(3), 185.

[13] Rama, K., Canhão, H., Carvalho, A. M., & Vinga, S. (2019). AliClu-Temporal sequence alignment for clustering longitudinal clinical data. *BMC Medical Informatics and Decision Making*, *19*(1), 289.

[14] Zavala-Rojas, D., Sorato, D., Hareide, L., & Hofland, K. (2022). The Multilingual Corpus of Survey Questionnaires: a tool for refining survey translation. *Meta*, *67*(1), 71-93.

[15] Facile, R., Muhlbradt, E. E., Gong, M., Li, Q., Popat, V., Pétavy, F., ... & Jauregui Wurst, B. (2022). Use of clinical data interchange standards consortium (CDISC) standards for real-world data: expert perspectives from a qualitative Delphi survey. *JMIR medical informatics*, *10*(1), e30363.

[16] Stein, B., & Morrison, A. (2014). The enterprise data lake: Better integration and deeper analytics. *PwC Technology Forecast: Rethinking integration*, *1*(1-9), 18.

[17] Ogunsola, K. O., Balogun, E. D., & Ogunmokun, A. S. (2022). Developing an automated ETL pipeline model for enhanced data quality and governance in analytics. *International Journal of Multidisciplinary Research and Growth Evaluation*, *3*(1), 791-796.

[18] De Jong, M., van Deursen, A., & Cleve, A. (2017, May). Zero-downtime SQL database schema evolution for continuous deployment. In *2017, IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)* (pp. 143-152). IEEE.

[19] Crumley Alvarez, P. A. (2004). *Managing large-scale systems with automated, centralized applications: using the Automated Submission System for law reviews* (Doctoral dissertation, Massachusetts Institute of Technology).

[20] He, X., Dong, H., Yang, W., & Li, W. (2023). Multi-source information fusion technology and its application in smart distribution power system. *Sustainability*, *15*(7), 6170.

[21] Hirschman, L., Grishman, R., & Sager, N. (1976, June). From text to structured information: automatic processing of medical reports. In *Proceedings of the June 7-10, 1976, national computer conference and exposition* (pp. 267-275).

[22] Kern, C., Klausch, T., & Kreuter, F. (2019, April). Tree-based machine learning methods for survey research. In *Survey research methods* (Vol. 13, No. 1, p. 73).

[23] Epoka, B. E. (2023). Literature Review of Qualitative Data with Natural Language Processing. *Journal of Robotics Spectrum*, *1*, 056-065.

[24] Cornelissen, B., Zaidman, A., Van Deursen, A., Moonen, L., & Koschke, R. (2009). A systematic survey of program comprehension through dynamic analysis. *IEEE Transactions on Software Engineering*, *35*(5), 684-702.

[25] Gunaratna, K., Lalithsena, S., & Sheth, A. (2014). Alignment and dataset identification for linked data in the Semantic Web. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(2), 139-151.

[26] Sánchez, L., Lanza, J., Santana, J. R., Sotres, P., González, V., Martín, L., ... & Crespi, N. (2023). Data enrichment toolchain: a data linking and enrichment platform for heterogeneous data. *IEEE Access*, *11*, 103079-103091.

[27] Konyushkova, K., Sznitman, R., & Fua, P. (2017). Learning active learning from data. *Advances in Neural Information Processing Systems*, *30*.

[28] Fiannaca, A. J., Kulkarni, C., Cai, C. J., & Terry, M. (2023, April). Programming without a programming language: Challenges and opportunities for designing developer tools for prompt programming. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1-7).

[29] Sun, P. J. (2019). Privacy protection and data security in cloud computing: a survey, challenges, and solutions. *IEEE Access*, *7*, 147420-147452.

[30] Chirumamilla, K. R. (2023). Low-Latency Data Pipelines Using Kafka and Snowflake. *JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE)*, *11*(1), 80-106.