Enhancing Aircraft Type Matching Across Data Sources Using Entity Resolution

Dissertation Manuscript

Submitted to National University

School of Technology and Engineering

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

by

KARNA LYNN BRYAN

San Diego, California

March 2026

Abstract

This study addresses the challenge of merging aviation safety data from diverse sources that lack shared identifiers, which limits the ability to combine safety events and utilization information for risk assessment. Inconsistent aircraft type representations make it difficult to align records across systems at scale. The purpose of this constructive research study was to develop and evaluate an aircraft type entity matching capability that supports normalization to a common aircraft type reference representation. A manually curated gold dataset was constructed by mapping aircraft type taxonomies from the Federal Aviation Administration and the International Civil Aviation Organization. Using this labeled dataset, a deterministic baseline matcher was implemented using feature similarity scoring and thresholding. Then, a deep learning matcher was trained using a transformer-based pairwise classification approach that serializes record fields into a textual representation. Experiments evaluated negative sampling strategies, schema variation across sources, and multi-source union training. Performance was assessed on held-out test partitions and a top-1 linkage selection strategy. In-domain transfer was assessed on additional "dirty" aviation datasets using make-constrained candidate generation, manual audits, and a fully labeled Bureau of Transportation Statistics case to quantify transfer versus source-specific fine-tuning. Results showed that the deep learning approach outperformed the deterministic baseline, remained effective under schema variation, and showed promising transfer behavior. The findings indicate that deep learning approaches provide a practical foundation for aircraft type normalization across heterogeneous sources, enabling cross-source safety analyses that were previously difficult to conduct. Future work should prioritize automated blocking and standardization, broader labeling for additional in-domain sources, and systematic study of negative candidate selection under deployment-like candidate distributions.

Two GitHub repositories support this study. The repository https://github.com/karnabryan/aircraft_er contains all data preparation and the baseline feature-based data matcher. The repository https://github.com/karnabryan/ditto_aircraft_er is forked from megagonlabs/ditto and contains the Ditto deep learning entity resolution model. The forked repository contains no modifications to the Ditto framework, but all aircraft type specific model runs and metric calculations.

Acknowledgements

First and foremost, I would like to thank my family for supporting me on this journey. My three sons watched me study alongside them through high school and college. Thanks to them for embracing the additional independence that came with my crazy-busy student-worker status. In the end, we all benefited from the opportunity to each find our own way. Thank you to my husband for picking up the slack whenever possible and for putting a lot of local exploration on hold while I was busy studying. Thank you as well to my dad, sister, and long-time friends for frequently checking in on me and recognizing the magnitude of the commitment.

A very special thank you to my advisor, Dr. Irene Tsapara, for her support, especially during the critical phases of this process. She encouraged me to seek opportunities to publish my work, which greatly enhanced this experience. Thank you as well to my committee members, Dr. Aeron Zentner and Dr. Hamzah Al-Najada, and to Dr. Laura Harris and Dr. Lawrence Fulton, who were there with me at the start of this journey. I know you all had to learn more about how aircraft type is described than you ever planned to know, so thank you for indulging me in my "North Star" of understanding practical ways to bring together aviation safety data from a long list of sources.

A very big thank you to the Department of Transportation for providing assistance on this journey through the Volpe Fellows Program and the Federal Aviation Administration Degree Completion Program. Both programs provided tangible and moral encouragement, and I am grateful.

Finally, thank you to my "upbringing" at the NATO Center for Maritime Research and Experimentation and to all my incredible colleagues there. I am sure that I find myself here because of those memorable times and the fulfillment I found in that research community.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In 2018 and 2019, two Boeing 737 Max jets tragically crashed, resulting in the loss of nearly 350 lives and placing significant scrutiny on both Boeing and oversight of the company by the Federal Aviation Administration (FAA) (Herkert et al., 2020; Luo et al., 2020). The FAA is the national civil aviation authority for the United States (US) and implements regulations within the US according to the framework established by the International Civil Aviation Organization (ICAO) (De Florio, 2016). ICAO is an agency of the United Nations tasked with creating policies, standards, and regulations for the global airline industry which are in turn implemented by national and regional regulatory authorities. The responsibilities of aviation authorities include certifying new aircraft and aircraft components and monitoring the continued operational safety of these aircraft and related aircraft products (De Florio, 2016; Yu-ping et al., 2020). The FAA grounded all Boeing Max 8 aircraft both nationally and internationally days after the second high-profile crash and all Max 8 aircraft remained grounded until November 2020 (Englehardt et al., 2021). Despite FAA clearance, China's Civil Aviation Administration withheld service resumption until January 2023 (AirGuide Business, 2023).

In reaction to the 2018 and 2019 Max 8 events, US Congress passed the Aircraft Certification Reform and Accountability Act (2020). This act demanded heightened transparency and accountability from both regulatory bodies and manufacturers and called for enhancements to the FAA's oversight process (Aircraft Certification Reform and Accountability Act, 2020). The process used by the FAA Aircraft Certification Service (AIR) to assess aircraft-specific safety risk, known as Monitor Safety/Analyze Data (MSAD), was subjected to rigorous examination. The National Academies of Sciences, Engineering, and Medicine (2022) undertook a thorough review of the MSAD Transport Airplane Risk Assessment Methodology (TARAM).

This report identified numerous shortcomings and obstacles in the completeness, accessibility, and quality of the data fed into the TARAM process. The study proposed improvements to the process, including more systematic risk modeling, as well as the inclusion of techniques from reliability analysis and uncertainty analysis. The report also recommended enhancing the quality and accessibility of input data through agreements with airplane manufacturers, their suppliers, and aircraft operators, together with increased standardization of data exchange and routine assessments of data quality.

Less than a year after the last Max 8 had returned to service in 2023, an Alaska Airlines Boeing 737 Max 9 aircraft lost a cabin door plug mid-flight leaving a hole in the fuselage and forcing an immediate emergency landing (National Transportation Safety Board, 2024). While there were no associated fatalities or serious injuries, the Boeing 737 Max 9 fleet was immediately grounded, and attention quickly turned back to Boeing and regulation of the company (Walker & Chokshi, 2024). Media attention on the 737 Max refocused public attention on aviation safety and provided renewed public interest in aircraft type, especially on the Boeing Max series (Sachs et al., 2024).

This study focuses on the general role of aircraft type in aviation safety and risk analyses and the challenges in using aircraft type in safety analyses due to a lack of standardization in the way that data systems refer to aircraft type (Federal Aviation Administration, 2024b). The 737 Max example merely provides context into the role of regulators, the oversight process, and continued airworthiness in the aviation industry. Many processes for monitoring hazards and risks to ensure aviation safety are based on data aggregated by aircraft type, and regulators provide directives, such as airworthiness directives, special airworthiness information bulletins, and other continued operational safety guidance based on aircraft type (De Florio, 2016).

While many data systems compile aviation accidents and incidents, merely counting incidents and accidents does not provide a good sense of quantitative risk (Churchwell et al., 2018). Counts need to be normalized by a rate of utilization for the population being analyzed to get an accurate understanding of risk or other metrics for exposure (Aguiar et al., 2017; Badanik et al., 2021; Blom, 2010; Boyd, 2017; Marais & Robichaud, 2012). Aircraft type is commonly used to create a population for safety analysis (Churchwell et al., 2018; Smith, 2023) or compare safety between two different fleets of aircraft types (Wild, 2023). If safety events are in one information system but data on aircraft flight hours are in another system, or even unknown, this provides a challenge for quantitative safety-related trend analysis. Information from multiple databases is needed to calculate safety event rates and provide quantitative risk trends (Boyd, 2017; Kierszbaum & Lapasset, 2020). These rates, such as events per 100,000 flights, incorporate both raw event counts and normalization metrics like aircraft utilization or fleet size (Aguiar et al., 2017; Badanik et al., 2021; Boyd, 2017).

The inability to connect these data sources using common identifiers hinders the calculation of rates where one system counts the number of accidents and another system collects flight hours or aircraft counts (Buselli et al., 2021; Rose et al., 2020; Zhao et al., 2018). Aircraft type, or make-model, is included in most databases, but, because there is no standard representation for aircraft type and different systems use different nomenclature and levels of granularity, metrics combining more than one information source require significant cleaning by safety analysts. The ability to map aircraft type across data sources would enable new data-driven risk analyses that incorporate both accident counts and fleet utilization or fleet size on a much larger scale than is currently feasible through manual cleaning by subject matter experts.

To illustrate the complexity in the description of aircraft type across multiple systems, an example of the Bombardier Challenger business jet can be used to show how aircraft type definition can vary across multiple systems. The aircraft used in this example is commonly referred to as the Bombardier Challenger 300 or 350 by the manufacturer (Flight Safety Foundation, 2023) and the Commercial Aviation Safety Team (CAST)/ICAO Common Taxonomy Team (CICTT) taxonomy lists Bombardier Challenger 300 or 350 as the aircraft's marketing terms (ICAO, 2025b). However, the actual model and series used in the CICTT taxonomy is BD100 1A10.

In the U.S. Federal Aviation Administration Aircraft Registry, the aircraft is referred to as model BD-100-1A10. To demonstrate an additional complexity, there are two separate entries for the BD-100-1A10, one listing the manufacturer as "Bombardier Inc." and the other listing the manufacturer as "Bombardier Aerospace Inc." (Federal Aviation Administration, 2024a). Resolving these two different listings for manufacturer as the same entity is a classic data duplication problem (Christen, 2012). The US Aircraft Registry lists 754 distinct BD-100-1A10s from these two distinct entries, providing a physical count of the aircraft of this type that are based in or operate in the United States.

Moving to a different system, the Bureau of Transportation Statistics (BTS) lists this aircraft twice: once as "Bomardier Challenger 350" with a typo in the manufacturer's name, and again as "Bombardier BD-100-1A10 Challenger 300" (Bureau of Transportation Statistics, 2025). The BTS provides valuable data on the number of flight hours and distances traveled by aircraft type over the past five years, information that is not available in the previously mentioned data sources. Pilots and air traffic controllers use yet another standard based on four-character ICAO codes for aircraft types and the Challenger 300 and 350 are listed as a CL30 and

CL35, respectively (Federal Aviation Administration, 2026). While a subject matter expert may be able to group together these variations and understand that the Challenger 300, BD-100-1A10, and CL30 are the same aircraft type, baseline automated data matching algorithms are based on standardizing columns and punctuation and string similarity (Christen, 2012). These baseline methods may struggle with the Bombardier Challenger BD-100-1A10 example just described.

The CICTT taxonomy published by ICAO serves as a worldwide reference for identifying aircraft types and seeks to be the global taxonomy for aircraft type used by the industry (International Civil Aviation Organization, 2026). However, historically aviation data systems used their own taxonomies for aircraft type or left aircraft type as a free-form, user-defined field and many of these systems have not adopted the common global taxonomy (Federal Aviation Administration, 2024b). While the FAA Order 8000.71 directs that the ICAO taxonomy should be used within the agency when specifying aircraft types, an initial exploration of aircraft type data fields for common aviation sources published by the FAA shows that this adoption is not yet complete (Federal Aviation Administration, 2024a; Federal Aviation Administration, 2024b; Federal Aviation Administration, 2025a; Federal Aviation Administration, 2025b).

Entity resolution is a class of tools and techniques to match data across data sources where the data have no common keys or identifiers (Christen, 2012). Historically, entity resolution involved a set of data transformations to clean data fields by transformations such as removing punctuation, combining fields, and using a technique known as blocking to create a small subset of the data on which to make comparisons (Papadakis et al., 2023). These approaches can often handle minor misspellings and slight variations between descriptors, but do not solve the issue of how to align BD-100-1A10, CL30, and Challenger 300 because there is little similarity between these textual descriptors. Recent research in the ER domain has

incorporated large language models and deep learning techniques expanding the applicability of entity resolution approaches (Agarwal et al., 2022; Brunner & Stockinger, 2020; Ebraheem et al., 2018; Hegselmann et al., 2023). New techniques provide solutions for systems with the information needed to match entities in non-standard columns (Hegselmann et al., 2023), with errors (Agarwal et al., 2022), expressed using different terms (Ebraheem et al., 2018), and expressed with various levels of specificity (Backes et al., 2022).

To date, the use of entity resolution techniques in aviation and on aircraft types is a topic that has not been explored in academic literature. Table 1 provides the results of several search queries using the National University NavigatorSearch with peer-reviewed, full-text articles. Each query was explored once with no time constraint on publication date and once where publication date was constrained to the previous five years (as of May 2024). The searches were based on four main topics: entity resolution, using search terms "entity resolution or record linkage or entity matching;" aviation, using search terms "aviation or aircraft or airplane or airlines;" natural language processing (NLP), using search terms "natural language processing or nlp or sentiment analysis;" and, finally, aircraft type. The query was first generated for each of these single topics alone and then using various combinations.

Table 1

Search term results for single and aggregate topics related to entity resolution and aircraft type

Search terms	No date filter	Last five years
Single search terms		
entity resolution or record linkage or entity matching	25,256	6,535
aviation or aircraft or airplane or airlines	514,351	187,415
natural language processing or nlp or sentiment analysis	132,003	85,470
aircraft type	3,449	1,591
Aggregate search terms		

((entity resolution or record linkage or entity matching) and (aviation or aircraft or airplane or airlines))	22	5
((entity resolution or record linkage or entity matching) and (aircraft type))	0	0
((entity resolution or record linkage or entity matching) and (aircraft type)) adding SmartText[a]	2	1
((natural language processing or nlp or sentiment analysis) and (aviation or aircraft or airplane or airlines))	642	511
((natural language processing or nlp or sentiment analysis) and (aircraft type))	4	4

Note. [a] SmartText is automatically added by search when no results were provided

All individual topics have produced many publications without considering time constraints on the date of publication and for publications in the last five years. In the last five years, there are 187,415 publications in NavigatorSearch focused on aviation, 84,470 publications on NLP, 6,535 on entity resolution, and 1,591 discussing aircraft type. All selected topics are frequently discussed in academic publications. In looking at topics, papers that discuss NLP and aviation are relatively common with 511 papers in the last five years, but NLP and aircraft type is not frequently discussed with only four papers in the selected previous five-year time period. There are only five publications that combine entity resolution and aviation during the last five years and 22 with no date filter. A standard Boolean term search on entity resolution and aircraft type resulted in no publications, automatically triggering a switch to Navigator Search's SmartText Searching capability which searches beyond standard search fields. This resulted in 2 papers when no time constraint on the publication date was added, with one of these being in the selected previous five-year time period.

The recent paper uncovered by this SmartText Search identified a paper looking at spatial entity resolution using aviation data, where the authors were interested in finding similar geographic data entries based on multiple data sources (Khodizadeh-Naharidid et al., 2021). The focus of this paper is on geographic entities, such as airports or places, rather than on aircraft

type. The SmartText Search with no time constraints did reveal a study where entity resolution was used to match US-registered helicopters with helicopter-related accidents (Churchwell et al., 2018). The authors manually resolved the aircraft in this study to carry out their analysis.

Within the population of the twenty-two publications discussing both entity resolution and aviation data, many of the papers identified did not address aviation, but had authors from aeronautical departments as part of the affiliation. Other papers focused on identifying aircraft system hazards without focus on aircraft type, on spatial entity resolution for imagery collected on airplane mounted sensors, on aircraft noise issues for people living near airports, on human trafficking from air travel, on health issues for pilots and cabin safety crew, and on deep vein thrombosis. While each of these papers addresses some type of entity resolution problem in the aviation domain, these publications do not discuss the role of aircraft type and are not related to aviation safety analysis from an aircraft continued operational safety point of view. The application of entity resolution techniques to aircraft type is an undocumented gap in the academic literature.

Statement of the Problem

The problem addressed in this study is the challenge of merging aviation data from diverse sources that lack common keys, hindering safety analysts' ability to assess risk using multiple information sources. Recent progress in entity resolution using machine learning and large language models has been applied to several open-source datasets for product matching, publication matching, song matching, and other applications, yet, to date, based on a comprehensive literature search, there is no evidence that researchers have attempted to use these techniques to improve the resolutions of aircraft types between aviation databases (Agarwal, Singh, & Chaurasiya, 2022; Akritidis et al., 2020; Backes et al., 2022; Sun & Shen, 2022; Ziv &

Fire, 2022). Current aviation research explores the use of textual narratives for clustering and classification of safety reports and the use of fine-tuned language models specific to aviation to build aviation knowledge graphs, but most of these focus on extracting information from a single source of data, rather than connecting data across systems (Agarwal, Gite, et al., 2022; Ahadh et al., 2021; Amin et al., 2022; Buselli et al., 2021; Kierszbau et al., 2022; Yang & Huang, 2023). Without a greater ability to connect data from multiple sources, aviation safety analysts struggle to quantify the risk of similar incidents occurring on a fleet of that aircraft type and to obtain an accurate interpretation of fleet safety risks (National Academies of Sciences, Engineering, and Medicine, 2022).

Purpose of the Study

The purpose of this constructive research study is to develop an entity resolution model that can connect disparate aviation databases by matching across data sources using aircraft make-model-series specifications. The goal is to automate the process of record linkage across aviation databases that lack common fields, thereby enabling stakeholders to answer new analysis questions. Different systems use different specifications for aircraft type, as well as differences in the granularity of aircraft type categorization and data sources have not adopted a common taxonomy. Most analysis questions cannot be answered using a single data source but require combining multiple sources. Techniques to combine data sources using aircraft type would allow analysts to answer risk-related safety questions related to aircraft type at scale. The study will focus on aircraft that are either registered or fly within the United States, using publicly available data sources that include accident and incident reporting, aircraft registration, and aircraft usage statistics and will explore the use of an international standard taxonomy for aircraft type. The anticipated outcome is a method to blend aviation data from multiple sources

by aggregating by aircraft type, thereby filling a gap in the ability to use aircraft type to analyze data across multiple sources.

This study will use existing publicly available datasets as secondary sources for safety event data and aircraft fleet size and utilization information. Sources for safety events include the National Wildlife Strike Database (Federal Aviation Administration, 2024b), accident and incident information from National Transportation Safety Board (National Transportation Safety Board, 2025) and the Federal Aviation Administration (FAA) Accident and Incident Database (Federal Aviation Administration, 2024d). Data sources for aircraft fleet size and utilization will use data from the US Federal Aircraft Registry (Federal Aviation Administration, 2024a), the Bureau of Transportation Statistics (Bureau of Transportation Statistics, 2025), and the FAA Operator Aircraft tables (Federal Aviation Administration, 2024b). For aircraft type taxonomy, the study will use data from the FAA Aircraft Reference tables (Federal Aviation Administration, 2024a), the previously introduced ICAO CICTT taxonomy (ICAO, 2025b), and the ICAO "DOC 8643-Aircraft Type Designators" dataset (ICAO, 2025a).

Introduction to Conceptual Framework

The overarching data manipulation framework will follow the Cross-Industry Standard Process for Data Mining (CRISP-DM) process model comprised of business understanding, data preparation, modeling, evaluation, and deployment (Shearer, 2000). The CRISP-DM process is a structured method to approach data projects to make the process clear and understandable and to ensure that the project addresses the underlying business need. This project will collect and curate aircraft type data from multiple sources and create artifacts for a 'gold standard' dataset and entity resolution algorithms on aircraft type data to compare traditional approaches for data matching with newer techniques using machine learning and large language models.

Within the CRISP-DM, the theoretical framework inside the modeling stage will be that of the No Free Lunch Theory (NFLT) which states that for each new domain and specific business need, a different machine learning approach will likely be more effective based on how the machine learning model works with that specific data source (Sterkenburg & Grünwald, 2021). Within the evaluation step of the CRISP-DM, theory from statistical inference, namely frequentist inference, will be used to perform hypothesis testing on the F1 scores of various machine learning techniques to determine if one model does perform significantly better than other models on aviation specific data to match aircraft type (Demšar, 2006). Theory from the entity resolution framework will also be used (Christen, 2012), falling more under the domain of data integration than data science, but strongly supporting the requirements of this study.

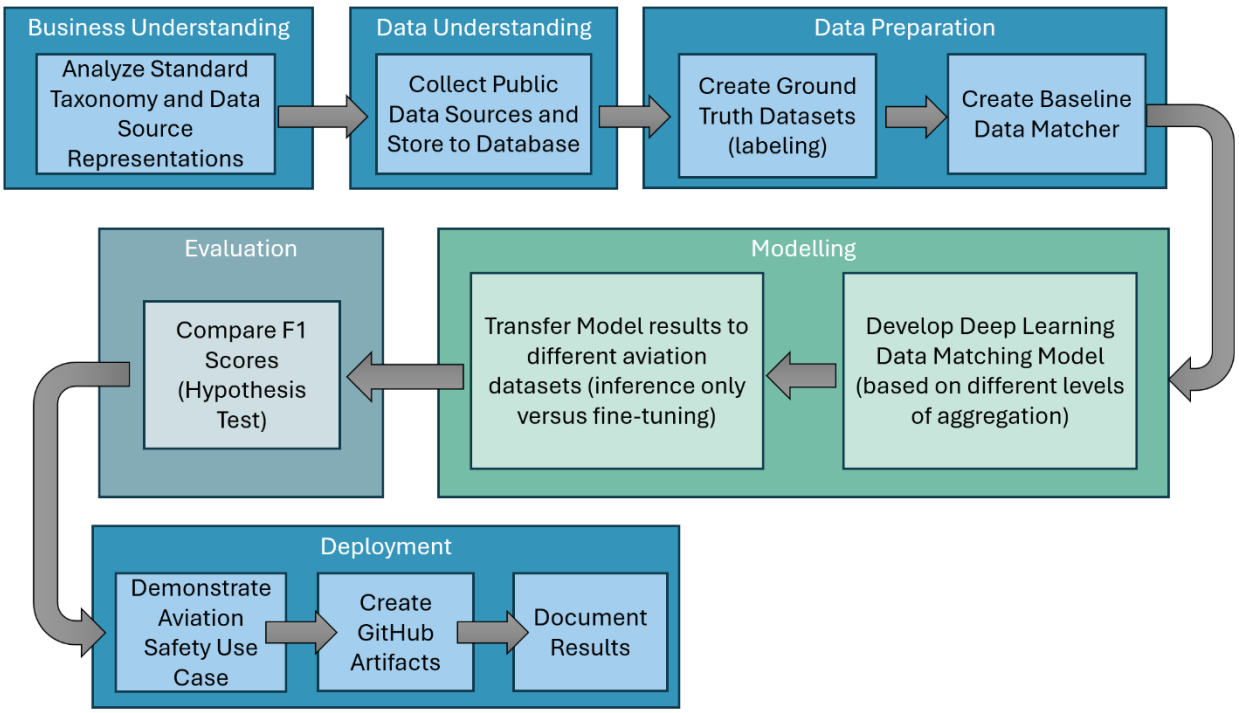
Introduction to Research Methodology and Design (Nature of the Study)

This project uses a constructive research methodology aimed at devising a novel solution for a practical issue (Brendel et al., 2022). This research methodology provides the flexibility to design models and artifacts targeted specifically to the aircraft type data matching problem. Figure 1 shows a process diagram for the proposed design to address this constructive research problem integrated into the CRISP-DM process methodology. The business understanding phase will evaluate the existing ICAO aircraft taxonomy for aircraft type definition and assess the business need aircraft type definition. The data understanding phase will examine each of the data sources and perform some initial exploratory data analysis on the aircraft type definitions in those sources. The data preparation phase will contain the critical step to manually create a gold dataset to be used to train and evaluate supervised learning approaches and provide the ability to compare these techniques with baseline entity resolution algorithms that do not use machine

learning. This phase will also create the baseline data matcher to compare with results based on deep learning strategies.

Figure 1

Process Model based on CRISP-DM Framework



The data modeling phase will first explore deep learning approaches to match aircraft type using a taxonomy. This exploration includes multiple feature set combinations, expressed as aggregated strings, as well as multiple matching hierarchies. In addition to a deep learning approach for matching two aircraft taxonomies, the study will look at techniques that can match additional aircraft data sources. This includes using the existing model trained on the taxonomies as well as a fine-tuned approach. The evaluation phase will use the labeled data to compare the different types of supervised learning approaches to determine if a given approach performs better than other approaches on aviation type matching and assess overall model performance.

Finally, the dissemination phase will focus on demonstrating the use of developed techniques on an aviation safety use case, publishing the gold dataset and techniques for use by other researchers, and documenting the results.

The modeling phase will examine newer-generation entity resolution algorithms using deep learning, namely Ditto (Li et al., 2021). These approaches are the current state of the art for entity resolution, blending the traditional field of entity resolution laid out by Christen (2012) with recent advances in deep learning. Research in this area is supported by several public gold standard datasets to test the performance of these approaches on different data challenges. Consistent with NFLT, different models and model configurations perform better with different data sources. This study aims to contribute a new aviation-specific dataset to the research community and to evaluate which model configurations show the best performance on the aircraft type data matching problem. The study will not explore automated blocking techniques used to group together similar records and reduce computational complexity by reducing the number of required pairwise comparisons (Christen, 2012).

Research Questions

This constructive research study will explore the following research questions that progressively develop concepts needed to align data by aircraft type:

RQ1

To what extent, if any, can a traditional entity resolution approach based on feature matching be used to match aircraft type between data systems?

RQ2

To what extent, if any, can deep learning techniques be used to match aircraft type between data systems?

RQ3

To what extent, if any, is in-domain data transfer possible using deep learning to match aircraft type between systems?

Hypotheses

While this study is constructive in nature, comparisons between the performance of traditional automated techniques developed in the data preparation stage can be compared with the approaches developed in RQ1 and RQ2. The following hypotheses will be tested:

$H1_0$

There is no significant difference in performance between traditional automated entity resolution techniques and one that utilizes a deep learning approach when measured by the F1-value.

$H1_a$

There is a significant difference in performance between traditional automated entity resolution techniques and one that utilizes a deep learning approach when measured by the F1-value.

Significance of the Study

The research study aims to utilize advancements in entity resolution on aviation data to align various sources by aircraft type. The goal is to improve quantitative risk assessment in the aviation domain by providing new techniques to perform data matching across data sources that lack common keys. The study will explore and formulate entity resolution methods tailored for aviation, generating a shareable dataset and code that enables the research community to benefit from the results of the study. By providing a publicly available dataset for entity resolution on aircraft type data, the study may also encourage other researchers to further develop this domain

by having another benchmark dataset for algorithmic development. By automating the data consolidation process across multiple systems, the hope is that the research can enable more widespread analysis using multiple systems.

While natural language processing (NLP) techniques have been applied to the area of aviation safety, these studies are all based on a single data source (Agarwal, Gite, et al., 2022; Ahadh et al., 2021; Amin et al., 2022; Buselli et al., 2021; Kierszbau et al., 2022; Yang & Huang, 2023). An entity resolution technique for aircraft type that supports the combination of data across multiple sources would support researchers in this area to perform similar safety studies on data integrated across multiple sources.

Current research activity in aviation safety compares safety across two or more aircraft types (Boyd, 2017; Wild, 2023) or analyzes safety events for a particular type or class of aircraft (Churchwell et al., 2018; Smith, 2023), but these studies rely on a domain subject matter expert to map aircraft type resolutions across data sources. A technique that could reliably generate these resolutions would enable researchers in aviation safety to use aircraft type in their studies without the overhead of manually correlating multiple data sources. This could enable safety analyses focused on aircraft type at a much larger scale than is currently feasible.

Definitions of Key Terms

Aircraft Type

Aircraft Type serves as a fundamental categorization of aircraft products and is commonly broken down into the terms “make,” “model,” and “series” (Badanik et al., 2021; Federal Aviation Administration, 2024a).

Aircraft Make

Aircraft Make is indicative of the company or manufacturer responsible for the production of the aircraft. Recognizable names such as Cessna, Boeing, Airbus, and Bombardier fall under this category (Federal Aviation Administration, 2024b).

Aircraft Model

Aircraft Model refers to the specific product line of an aircraft, describing a unique product design. Some examples of make and model are Cessna 172, Boeing 737, Airbus A320, and BD-100 (Federal Aviation Administration, 2024b).

Aircraft Series

Aircraft series is a more detailed representation of aircraft type. Aircraft series describes more specific characteristics of subtypes or configurations. Some examples of make-model-series are Cessna 172J, Boeing 737-800, Airbus A320neo, or BD-100-1A10 (Federal Aviation Administration, 2024b).

Blocking Techniques

Blocking techniques are used as part of the entity resolution process to group together similar records and reduce computational complexity by reducing the number of required pairwise comparisons (Christen, 2012; Javdani et al., 2019). Blocking techniques aim to reduce computational overhead by making only promising pairwise comparisons and skipping evaluations between pairs that would not likely match.

Entity Resolution (ER)

Entity resolution is a part of data integration that involves determining if two records refer to the same physical entity (Agarwal et al., 2022; Christen, 2012). Entity resolution can be used to remove duplicate records in databases or to merge two data sources with no common identifying fields.

Summary

This study proposes to develop artifacts to address the critical issue of integrating aviation data from various sources, where the data sources lack common keys for easy merging. Aviation safety analysts are currently hindered by the inability to merge these data easily and reliably across sources, as multiple sources are required for risk assessment calculations. The primary objective of the study will be to design an entity resolution model based on deep learning techniques that can effectively link different aviation databases by matching aircraft type. As part of the project, a gold standard dataset will be curated to enable the use of supervised learning techniques and promote additional research in this area.

The research will be guided by CRISP-DM model, providing a structured approach to organize and implement the process and ensure the project remains focused on the underlying business need and statement of the problem. In addition, the NFLT will be applied to evaluate various deep learning configurations for entity resolution in the context of matching by aircraft type. NFLT theorizes that no single supervised learning algorithm works best for every problem, underscoring the need to test different approaches for the development of this constructive research project. As the study uses data from multiple sources, the research will explore model configuration aspects supporting multi-source data.

New techniques to match aviation data across sources that lack common keys would improve quantitative risk calculations for aviation safety. These techniques can improve aviation safety by simplifying the process to understand accident and incident rates, combining safety event counts with aircraft utilization data. These techniques can support researchers studying how to apply natural language processing techniques to aviation safety data by enabling richer data sources from multiple systems. Researchers in entity resolution can also use the

development of a new gold standard labeled dataset to evaluate their algorithms against a new application area.

Chapter 2: Literature Review

This study addresses barriers to large-scale, multi-source aviation safety analytics resulting from inconsistencies in how data is expressed across systems, using aircraft type as an initial area of study. The purpose of this study is to demonstrate how an entity resolution framework applied to matching records by aircraft type can overcome barriers in heterogeneous data analysis arising from a lack of common identifiers. This literature review examines themes in aviation safety analysis and aircraft type identification, then reviews entity resolution as a framework for multi-source data matching.

Conceptual Framework

This study is organized using a popular structured method to approach applied data science projects, the Cross-Industry Standard Process for Data Mining (CRISP-DM) (Shearer, 2000). This process model is composed of five distinct phases: business understanding, data preparation, modeling, evaluation, and deployment. This study uses the CRISP-DM process as an overarching conceptual framework, while integrating methodologies from the data science and computer science communities as appropriate. The study will collect and curate aircraft type data from multiple sources and a labeled dataset to examine the effectiveness of various approaches to data matching by aircraft type across several different sources of aviation data.

Within this structure, the core technical methodology explored is that of entity resolution (ER). ER is a formal discipline of matching records that refer to the same real-world entity across different data sources (Christen, 2012). The ER domain has evolved significantly since its inception in 1946 and has recently integrated performant deep learning solutions for entity matching that have been demonstrated on a set of standard ER benchmarks (Papadakis et al., 2023). New domains can uncover new challenges for ER techniques and stimulate research in

the ER domain (Primpeli & Bizer, 2020). By generating a new, labeled dataset for aircraft type entities, ER approaches can be evaluated for their effectiveness in aligning aviation sources across aircraft type and potentially offer new learnings to those developing new ER methods.

This study first focuses on the aviation safety domain, providing the necessary background to understand some challenges faced by aviation safety engineers and analysts. The study first examines the role of quantitative risk analysis in continued operational safety, discusses taxonomies for aircraft type, and provides an overview of current research in the aviation domain using natural language processing and deep learning. Next, the review looks at the historical development of the entity resolution framework and discusses the current state of the art, providing the foundation for how entity resolution models can address current gaps in aviation data integration.

Aviation Continued Operational Safety and Aircraft Type

Aviation safety is a complex domain with high safety expectations and a variety of stakeholders (Buselli et al., 2021). The first section of this literature review focuses on a single aspect of aviation safety, the continued operational safety of aircraft, and discusses current studies on the supporting quantitative risk analysis process. This leads to a discussion of taxonomies for aircraft type, followed by an overview of current research in the aviation domain using natural language processing, machine learning, and deep learning.

Quantitative Risk Methodology for Continued Operational Safety

ICAO, the international organization responsible for regulating global aviation, mandates that commercial aircraft are built according to a type certificate, held by a manufacturer and regulated by the federal aviation authority in the country of that manufacturer (De Florio, 2016). For example, Boeing holds the type certificate for the 737, for which the FAA issues and

maintains the type certificate data sheet (TCDS) “A16WE” (United States Department of Transportation, 2026). The FAA’s regulation of US type certificates goes beyond merely approving a type design to also include continued operational safety (De Florio, 2016).

Continued operational safety is “the monitoring and management of safety risks associated with in-service aircraft, products, and articles to ensure that unsafe conditions are identified and addressed promptly” (Federal Aviation Administration, 2023). Aircraft operators are required to report service difficulty reports (SDRs) to the FAA, and aircraft manufacturers have additional reporting requirements under CFR Part 21.3 to inform the FAA of issues experienced during operations (De Florio, 2016). Together with aviation accident, incident, and occurrence reports, the FAA uses this information to identify unsafe conditions and develop safety interventions. These interventions include publishing Airworthiness Directives, Service Bulletins, and Safety Alerts for Operators, which contain mandatory or advisory corrective actions when unsafe conditions are identified (De Florio, 2016). The process used by the FAA to determine unsafe conditions is outlined in FAA Order 8110.107B, known as “Monitor Safety/Analyze Data (MSAD).” The MSAD order sets requirements for the FAA to identify potential safety issues through available information sources, perform qualitative triage of reported safety events, and, for more severe issues, conduct a quantitative risk assessment (Federal Aviation Administration, 2023).

This process involves identifying the applicable fleet of aircraft and estimating the frequency of hazard occurrence within that group (Federal Aviation Administration, 2023). Fleets may be categorized in several ways, including by aircraft type, by specific operator, by regulatory classification such as all commercial transport aircraft, or by a subset of aircraft identified by a manufacturer as sharing a problematic configuration or component. When

grouped by aircraft type, this generally refers to the aircraft listed under a specific TCDS and may apply to the entire type certificate or to a subset of aircraft based on model and series information. These fleet groupings link observed hazards from safety reporting to fleet utilization metrics such as hours flown and operating cycles, enabling a risk-informed assessment of continued operational safety (Federal Aviation Administration, 2023).

The qualitative risk analysis described FAA Order 8110.107B identifies a specialized risk assessment process for large transport aircraft referred to as the "Transport Airplane Risk Assessment Methodology (TARAM)" (Federal Aviation Administration, 2023). This process was updated in 2023 to incorporate the results of an audit by the National Academy of Sciences in 2022. The audit evaluated the effectiveness of the TARAM process and alignment with modern data integration capabilities (National Academies of Sciences, Engineering, and Medicine, 2022).

The 2022 evaluation endorsed the mathematical foundation of the TARAM quantitative methodology while identifying gaps in the implementation. These gaps included the availability of data to support the process, the inconsistent use of the process within different regions of the FAA, and the management of uncertainty in the quantitative methodology (National Academies of Sciences, Engineering, and Medicine, 2022). The audit found the process to remain more reactive than proactive, citing earlier findings from the "In-Time Aviation Safety Management" report, which called for the development of safety systems that could detect emerging risks and support predictive analytics by integrating digital data across systems (National Academies of Sciences, Engineering, and Medicine, 2022). The 2022 evaluation concluded that insufficient progress had been made toward the implementation of enhanced and integrated safety data analytics capability since the 2018 recommendations.

These audits and findings have historically provided technical guidance and helped define the scope of how continued operational safety is implemented within the US. Decades earlier, a 1998 National Research Council report titled *Improving the Continued Airworthiness of Civil Aircraft* emphasized the regulator's role in providing oversight to the full span of the aircraft life cycle. While the agency's role in certification was well established, the 1998 study highlighted a broader need for oversight of aircraft through the full operational lifetime, establishing a clear expectation for airworthiness oversight as it exists today.

This study is specifically focused on improving data integration for aircraft type to support the quantitative risk assessment process for aviation safety. According to FAA Order 8110.107B, aircraft type is a critical aggregator to group aircraft into fleets. This review now examines the definition of aircraft type in different aviation sources.

Taxonomies for Aircraft Type

Aircraft type supports the identification and monitoring of safety risk, as well as the development of targeted interventions to mitigate these risks. As noted in FAA Order 8110.107B, aircraft type fleets are used to determine the scope of a fleet impacted by a safety hazard and to assess hazard frequency and failure rates (Federal Aviation Administration, 2023). Aggregate fleet exposure to a hazard is important for aviation safety engineers to determine the urgency and expected financial impacts of safety mitigations, highlighting the need for scalable aircraft type definitions that refer to a precise set of impacted aircraft (Federal Aviation Administration, 2023). Accurate, consistent, and precise definitions of aircraft type ensure that safety interventions can be targeted to the correct population and risk assessments are based on reliable data inputs.

FAA Order 8000.71 provides the FAA's internal taxonomy for aircraft type based on a structured make-model-series (MMS) taxonomy. The aircraft make refers to the name of the original aircraft manufacturer, the model refers to the specific aircraft family specified on the type certificate, and the series is a specific variant of the model that refers to specific configurations of the model (Federal Aviation Administration, 2017). MMS entries are defined on the TCDS of an aircraft and provide a hierarchical structure for aggregating aircraft. To further control how aircraft are aggregated, a "master model" and "master series" may be used to group aircraft with similar characteristics (Federal Aviation Administration, 2017).

For example, the "DC-9-11" is classified under the make "Douglas," with the model "DC-9," and series "11." The master model for this aircraft type is also a "DC-9," while the master series is "10" as a number of DC-9s are grouped into a single master series to denote their similarity. Table 2 provides an example of MMS attributes for a small selection of aircraft types.

Table 2

Aircraft Make-Model-Series Example Records

Make	Model	Master Series	Series	ICAO Type Designator	Popular Name	Type Certificate
Airbus	A320	200	214	A320		EASA.A.064
Airbus	A320	200N	271N	A20N	ACJ320 NEO	EASA.A.064
Boeing	737	800	808	B738		A16WE
Boeing	737	8	8	B38M	MAX 8	A16WE
Boeing	787	9	9	B789	Dreamliner	T00021SE
Bombardier	BD100	1A10	1A10	CL30 / CL35	Challenger 300/ Challenger 350	A-234

Cessna	560		XL	C56X	Citation XLS	A22CE
Douglas	DC9	10	11	DC91		A6WE
Douglas	DC9	80	81	MD81	MD81	A6WE

FAA Order 8000.71 provides specific guidance that any taxonomy should be aligned with the taxonomy implemented by the International Commercial Aviation Safety Team (CAST)/ICAO Common Taxonomy Team (CICTT) (Federal Aviation Administration, 2017). The CICTT brings together a wide range of aviation experts from airlines, aircraft and engine manufacturers, pilot union, safety boards, and regulatory agencies from several participating countries. They create shared taxonomies to produce a unified set of definitions and references that can be used across the industry to create more shareable data and a unified understanding of aviation safety data concerns (International Civil Aviation Organization (ICAO), 2026).

The CICTT Aircraft Taxonomy uses a strict hierarchical structure in which each series belongs to a single master series, each master series maps to one model, and each model rolls up to a single master model. This one-to-one relationship at each level ensures consistent aggregation across varying levels of detail, supporting accurate grouping for analysis and reporting (Commercial Aviation Safety Team Common Taxonomy Team, 2019).

The taxonomy also includes other important metadata related to the taxonomy entry including popular name and ICAO Type Designator. The popular name is a name used by the manufacturer in its marketing language for the aircraft. For example, the official make-model-series for Bombardier aircraft BD-100-1A10 is known in marketing terms as the “Challenger 350” (Bombardier, 2023). While not all aircraft have a popular name, for some aircraft the popular name is used almost exclusively to refer to the aircraft type. Popular names do not

respect the taxonomy's hierarchy: several popular names can refer to the same aircraft series (e.g., the BD-100-1A10 can either be a "Challenger 300" or a "Challenger 350") and in other cases the same popular name can be used for multiple series and even multiple models.

Another important standardization code for aircraft type included in the CICTT taxonomy is the ICAO Type Designator code. The ICAO Type Designator code is maintained by ICAO in their publication "Document 8643 - Aircraft Type Designators" and is a short, four-letter alphanumeric code used as a global reference for an aircraft type (International Civil Aviation Organization, 2025a). The Federal Aviation Administration uses a version of the document 8643 data published as Order JO 7360.1K within the Air Traffic Organization (Federal Aviation Administration, 2026).

Air traffic control, surveillance systems, and other commercial providers of aviation data use this four-letter code to identify aircraft. These type designators are generally aligned with make-model-series, but, like popular names, there can be several aircraft series that map to a single four-letter code and the same code may occasionally be used across multiple models. ICAO type designators group aircraft by performance and handling characteristics that are relevant to air traffic management. FAA Order JO 7360.1K provides similar information to ICAO Doc 8643 using the same aircraft type designator codes (Federal Aviation Administration, 2026).

The CICTT aircraft taxonomy also includes additional metadata for each taxonomy entry including aircraft category, aircraft sub-category, landing gear category, max certified number of passengers, max certified takeoff weight, and military aircraft indicator. This metadata can be used to create higher level classes in a hierarchy such as fixed wing aircraft carrying over 100 passengers, single-seat aircraft, or unmanned helicopters. The taxonomy metadata provides a

useful tool to build higher level analytic groupings. Additionally, each taxonomy entry also has the ICAO Type Designator, and these have additional metadata such as engine class, wake turbulence class, and number of engines. This engine information is not included in the CICTT aircraft taxonomy as CICTT has an additional taxonomy for engine type (International Civil Aviation Organization, 2026).

The FAA Order 8000.71 on aircraft make-model-series provides implementation guidance on how the aircraft taxonomies need to be implemented, rather than documenting a prescribed taxonomy (Federal Aviation Administration, 2017). ICAO has published adoption procedures as implementing guidance for aviation data systems (ICAO, 2006). The adoption guidance encourages aviation system implementers to create mappings between existing MMS definitions and the CICTT aircraft taxonomy including all five levels of definition (make, master-model, model, master-series, and series). They claim that many existing aviation information systems only include aircraft make and model columns, where all information is compressed into these limited fields, possibly in free-text form, with many inconsistencies in how the data is recorded, with examples such as “CESSNA 172E,” “172 F”, and “172G” (ICAO, 2006). They argue that these inconsistencies create unintended and unnecessary constraints on the aviation community’s ability to integrate, analyze, and share data, and result in duplicate and ambiguous entries when describing similar aviation events.

The FAA’s Accident Information Data System (AIDS) published through the Aviation Safety Information and Sharing (ASIAS) has a working aircraft type taxonomy in a downloadable reference file (Federal Aviation Administration, 2024). This aircraft type reference file contains a single aircraft MMS code, together with separate fields for aircraft make, model, and a master identifier. The file also contains similar metadata to the CICTT aircraft taxonomy,

including popular name, type certificate, and a military aircraft indicator. While the schema of this file is similar to the CICTT Aircraft Taxonomy, individual data fields are represented differently, often using different naming conventions and capturing the series information only as a hyphenated portion of the MMS code field rather than a stand-alone value. An additional column provides the seven-digit aircraft type code used in the FAA's Aircraft Registry.

The FAA Registry contains a record of each aircraft registered in the United States and the FAA provides both a searchable version of these records along with a downloadable aircraft database (Federal Aviation Administration, 2023). There are over 300,000 aircraft in the downloadable database of current aircraft in the FAA Registry. Each of these has a seven-digit code that links the aircraft to a master aircraft type reference file.

The master aircraft reference file contains information on the make, model (including both model and series information), type of engine, number of engines, number of seats, aircraft weight class, and the type certificate data sheet (Federal Aviation Administration, 2023). By joining on the registry model code, the ASIAS taxonomy can be expanded to include this additional metadata, providing similar information to that of the CICTT aircraft taxonomy. While this taxonomy provides similar information to the CICTT taxonomy, there is no way to join the two taxonomies and individual data fields are not expressed in a standardized format.

Several accident and incident databases in the US collect information from a variety of stakeholders, but the collected aircraft type is not generally constrained to any specific format for make and model. While the ASIAS aircraft taxonomy file is included with the AIDS downloadable accident and incident files, the reported make and model fields in this system do not always use the make and model from the reference file (Federal Aviation Administration, 2024). The National Transportation Safety Board provides an online user interface for reporting,

with free-text entry for aircraft make and model (National Transportation Safety Board, 2025). The FAA's National Wildlife Strike Database collects information on wildlife strikes with a single free-form text entry for aircraft make/model (Federal Aviation Administration, 2023). These examples show that three separate accident/incident reporting systems which collect similar and sometimes overlapping information have no standardized approach to aircraft type or links to a taxonomy.

The CICTT Aircraft Taxonomy is an international standard for aircraft type that provides a structure to aggregate aircraft at different levels of granularity for analysis purposes. While the FAA has endorsed the use of this taxonomy through FAA Order 8000.71, examining three separate systems for aviation accidents and incidents and their downloadable records, data entry for aircraft type is generally free-text and standardization for aircraft type is not observable through historical records. These systems demonstrate gaps in the ability to analyze data according to standardized types, which constrains multi-source trend analysis and predictive analytics using aircraft type as a feature.

Overview of Current Academic Research in Aviation Safety Using Natural Language Processing and Knowledge Graphs

In Chapter 1 of this study, NavigatorSearch results on the keywords "aviation" and "aircraft type" and variants of the keyword for "entity resolution" failed to identify previous academic research in this area. However, using "aviation" with "natural language processing" did demonstrate significant activity. This subsection briefly reviews recent academic research in aviation safety using natural language processing (NLP) and knowledge graphs (KGs), highlighting any references to aircraft type.

NLP Research in Aviation. Amin et al. (2022) reviewed the literature for current applications of NLP to aviation, grouping articles related to incident classification, predictive maintenance, and air traffic communications. Aviation safety events have been used with text classification, topic modeling, and pattern recognition to identify safety trends using textual descriptions of accidents, incidents, and occurrences. Text-based aviation maintenance reports have been used with classification algorithms to categorize maintenance occurrences and to forecast component failures from technical logbook reports. Speech recognition algorithms on air traffic control communications with pilots have been combined with named entity recognition (NER) to identify risks in pilot-controller communications.

Yang & Huang (2023) performed a systematic review of NLP approaches to aviation safety using three academic databases, covering academic articles between 2010 and 2022. The authors focused their analysis on two overlapping areas, the use of NLP to analyze aviation accident and incident reports and speech recognition and automated communications analysis between pilots and air traffic control. The authors identified several classification models that use safety event reports to classify risk factors and predict accident severity. Methodologies have increased in complexity from support vector machines (SVMs), to long short-term memory (LSTM) models, to hybrid deep learning models that combine multiple deep learning architectures for improved performance. The authors also discussed the use of latent Dirichlet allocation (LDA) and structural topic modeling (STM) to classify textual reports. The authors found that NLP was an effective tool for predictive modeling in aviation, but gaps remained in data annotation and interdisciplinary collaboration between aviation experts and NLP researchers.

Nanyonga et al. (2025) also reviewed NLP for aviation safety, focusing on 34 studies including the phrases “aviation safety” and “natural language processing.” Like earlier reviews, the authors found similar themes. This included the use of LDA and STM for topic modeling to uncover underlying patterns and trends from textual reports. They also noted the use of deep learning models like LSTM and bidirectional LSTM (BLSTM) to classify incidents and identify causal factors. Additional themes uncovered were the use of semantic analysis and word embeddings to establish similarities in reports. The authors proposed practical uses for industry, including greater insights, prioritized interventions, more efficient safety reporting and analysis, predictive capabilities for risks and safety hazards, and proactive risk mitigation. The authors also analyzed data sources used for each of the studies, only identifying a single study that used more than one data source, indicating that current studies tend to use a single source of data for analysis. The authors noted gaps with multimodal data integration (e.g. text, audio, and video) and observed an insufficiency of labeled data to train algorithms.

The reviews demonstrate consistent technical themes including topic modeling and safety event classification, with varying coverage of fields like predictive maintenance. Data integration themes are not present in the current work on NLP in aviation.

Aviation Knowledge Graphs and Ontologies. Another common theme in the academic literature on aviation safety is with aviation knowledge graphs. A knowledge graph is a graph-based structure where real-world entities are represented as nodes and relationships between these real-world entities form edges between the nodes (Agarwal et al., 2022). KGs can integrate and relate large amounts of disparate and isolated pieces of information to support inference across all entities and the relationships between them.

Knowledge graphs can be built in a “top-down,” “bottom-up,” or “hybrid” fashion (Agarwal et al., 2022). Top-down approaches first define the graph’s ontology and then extract knowledge from the data (Lv et al., 2023). The graph’s ontology is the definition of the nodes and type of information stored in each node, providing structure to how the information is organized, the concepts and entities stored in the graph, and the structure of the relationships between the entities. Top-down approaches first define the graph’s ontology and then extract knowledge from the data to fit into this ontology (Lv et al., 2023).

In contrast, bottom-up approaches first extract knowledge from the data and use this to define the ontology (Buselli et al., 2021). Bottom-up approaches tend to be used for discovery-driven analysis, similar to the role of topic modeling in other aviation safety analyses. This approach uses the data itself to uncover concealed entities and relationships.

Most knowledge graphs follow a hybrid approach using a combination of top-down and bottom-up styles to consolidate the data and information (Agarwal et al., 2022). The hybrid approach can handle diverse data sources and discrepancies in schema to create some initial structure for the KG with the ability to provide feedback and update the schema iteratively.

Zhao et al. (2018) used a KG to organize historical data on aviation risk and accidents using the Aviation Safety Reporting System (ASRS) database. The ASRS database contains voluntary, confidential safety information provided by aviation professionals (National Aeronautics and Space Administration, 2026). The authors mapped the relational database elements to resource description framework (RDF) triples developed using Protégé (Zhao et al., 2018). The authors provided a roadmap for how knowledge graphs could be built directly from a relational database and demonstrated how multi-dimensional semantic queries provided richer information than traditional keyword searches.

Cheng et al. (2019) used information from the civil aviation security field to construct a top-down knowledge graph. The authors first established an ontology for the aviation security domain and developed a rule-based approach to extract entities and relationships from the data sources into the KG. The authors used the Neo4j graph database with Cypher queries. While the authors did not provide specific details about their underlying data, they demonstrated the concept for a civil aviation security KG for knowledge extraction.

Wang et al. (2020) published the development of an aviation KG spanning the domains of fault diagnosis and mission optimization specific to a military use case. The authors used flight control system manuals, historical fault case descriptions, and military airspace data to construct a framework including both top-down and bottom-up themes. After defining an initial schema for their ontology, the authors used the extracted data to provide continuous feedback and refinement to their schema. The authors used the language model ERNIE to extract knowledge from their existing data sources. The authors demonstrate how KGs can be used to provide an information and decision-making advantage in complex aviation environments by integrating data from multiple, disparate sources.

Buselli et al. (2021) used publicly available loss of separation (LoS) reports recording when two operating aircraft become dangerously close to create a bottom-up, discovery-driven KG. The authors used latent Dirichlet allocation (LDA) for topic modeling and used hierarchical clustering to identify common themes in the report. After this phase of data extraction, the authors mapped their discovered patterns to an existing safety taxonomy defined by EUROCONTROL. They demonstrated a novel method to extract and classify safety factors extracted from unstructured textual reports and classify these into an existing taxonomy. The

authors claim that their framework enabled more targeted prevention strategies by providing a structured and evidence-based representation to understand why certain safety events occurred.

Agarwal et al. (2022) built a hybrid KG to build a question-and-answer (QA) capability for safety experts to review accidents by asking complex questions from a large collection of detailed safety accident reports from the National Transportation Safety Board. Their approach used a top-down approach to create an ontology in Protégé using the ICAO Accident/Incident Data Reporting (ADREP) taxonomy. Note that this taxonomy is different from the CICTT aircraft taxonomy as the ADREP focuses on a holistic high-level description of aviation attributes. The authors then used a bottom-up approach to extract information from the NTSB safety reports. They then produced a QA capability by creating a system that retrieves candidate answers from both the KG using SPARQL and separately from raw text using BERT-QA and GPT3-QA, then ranking the answers to find the best answer. The authors claimed that the ability to combine KGs with deep learning models was superior to using either approach in isolation. The aviation KG portion of this work demonstrates how a single information source can be used with complex aviation taxonomy to provide a comprehensive graph structure for accident reports.

Lv et al. (2023) built a high-level enterprise knowledge graph for civil aviation customer service using a top-down methodology. The authors created a detailed expert-driven ontology in Protégé using a structure defined by the business need for a commercial aviation application. The business objectives included the perspective of competitor analysis and technical aircraft specifications. The KG used data from enterprise websites, commercial databases, websites, and expert domain knowledge to fill in the ontology. The authors claim to have created a formal,

expert-driven ontology that can create a competitive intelligence advantage for an aircraft manufacturer.

This literature review also identified some additional noteworthy publications focused solely on the development of aviation ontologies prior to work in aviation KGs. For example, Durak et al. (2018) developed an ontology called the "Avionics Analytics Ontology (AAO)" to support decision making for Air Traffic Management (ATM) and Unmanned Aircraft System Traffic Management (UTM). They created a structured knowledge base for aircraft operations including information on airspace, flights, weather, and aircraft systems. Ledvinka et al. (2019) created a holistic ontology for the Civil Aviation Authority in the Czech Republic. The authors detailed a system able to integrate disparate safety data in support of data-driven safety management and safety analytics.

While the referenced KG studies are all in the aviation domain, they represent applications for many different aviation stakeholders including safety analysts, national regulators, aircraft manufacturers, military decision makers, air traffic management, and aircraft maintenance programs. They also range from implementations where the data informs the ontology to those where the data is used to complete an existing ontology. These studies demonstrate the necessary step to align elements of a knowledge graph with an existing or derived taxonomy for entity attributes. Detailed information on aircraft type based on a hierarchical taxonomy would be complementary to each of these KG implementations and could possibly serve as a link between existing KGs.

Entity Resolution Background

ER, a longstanding challenge, has a rich history. This section reviews the progression of ER as a field from the early years into a well-established set of techniques based on feature-wise

comparisons with an aggregate score. The background also includes a section on open-source implementations and benchmarks for ER as a critical enabler for framework development.

Early Years for Record Linkage

In 1946, Dunn published the article "Record Linkage," laying out the conceptual idea of linking records across different datasets. Dunn envisioned the creation of a "book of life" for each individual, beginning with birth, ending in death, and filled with significant life milestones. Dunn's framework proposed linking census, health, and social records into a population-level tool that could provide a 'source of truth' for an individual's life.

This initial concept of tracking individuals across life events was further developed by Newcombe in his 1959 article "Automatic Linkage of Vital Records." Newcombe introduced statistical decision rules that used probabilities to match records and included initial strategies to account for spelling variations and typographical errors. He advocated for the use of automation and computers to support record linkage.

A more formal theory for probabilistic record linkage was published by Fellegi and Sunter in 1969. Records were compared and evaluated as matching, not matching, or potentially matching, based on a comparison vector, using a numeric threshold to determine a match or non-match. Likelihood ratios were used to determine matches, and this framework provided the theoretical underpinnings of entity resolution as it is known today. This became known as the Fellegi-Sunter framework and was used and improved by the U.S. Census Bureau beginning in the 1980s to manage a growing quantity of government and survey data (Winkler, 1993). In addition to improvements to the probabilistic decision rules, new techniques were developed to compare strings, including the Jaro-Winkler metric (Winkler, 1990). This is a measure of

similarity between strings that can account for character transposition and common prefixes, attempting to further overcome typographical errors in data recording and other inconsistencies.

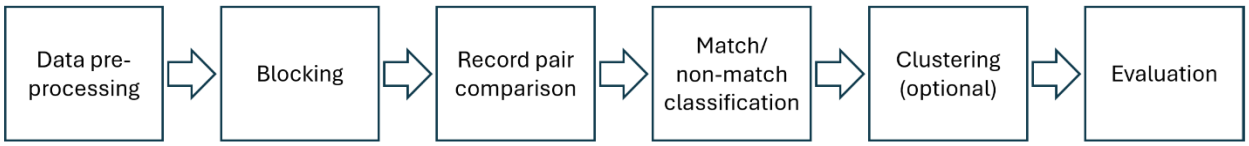
More advanced statistical techniques were explored in the 1990s to improve probabilistic data linkage using a mixture model of the true matches and false matches (Belin and Rubin, 1995). The Expectation-Maximization (EM) algorithm was used to estimate the latent parameters of the true-match versus false-match distribution, and the authors applied this to calibrate matching algorithms for the U.S. Census.

Current Foundations of Data Matching

In 2007, Elmagarmid et al. published "Duplicate Detection: A Survey," which presented a process flow for entity resolution that remains widely followed today. The authors outlined a canonical workflow including data preprocessing, blocking, similarity computation, classification, and clustering. An adapted view of this workflow is shown in Figure 2, which includes a later-adopted evaluation phase.

Figure 2

Standard Entity Resolution Framework



Elmagarmid et al. documented the prevailing rule-based and probabilistic approaches and extended this framework to incorporate early machine learning approaches for classification, as well as unsupervised clustering-based techniques for deduplicating records after the classification.

Also in 2007, Herzog et al. published a comprehensive text on record linkage techniques, which was conceptually aligned with the work of Elmagarmid et al. The authors provided in-depth coverage of established methods such as the Fellegi-Sunter probabilistic methods, the EM algorithm, and blocking techniques to reduce record pair comparisons. Their work emphasized data quality issues and added the evaluation step in Figure 2, focusing on measuring accuracy and understanding the impacts of linkage errors on downstream analysis.

In 2012, Christen published the text *Data Matching*, which expanded on these earlier works while maintaining the same general process flow. Christen introduced a stronger emphasis on scalability, system-level implementation, and software engineering. His text reinforced the modular nature of the workflow in Figure 2, demonstrating it as an adaptable tool for practical, real-world data matching implementations.

Together these works provide a robust conceptual framework for entity resolution, which continues to be used by researchers and practitioners. The following paragraphs offer a more detailed perspective on this workflow with a focus on advances from the broader community.

Data pre-processing. In the pre-processing phase, the initial steps involve addressing basic data quality and standardization issues, as well as generating new processed features that transform the original data (Christen, 2012). This often includes tasks such as parsing, standardization, and cleaning of data to improve its consistency and accuracy (Elmagarmid et al., 2007). In this stage, data cleaning may consist of correcting inconsistencies in data formats, spelling variations, or missing values with the goal of making the matching stage more effective (Köpcke et al., 2010). For example, there may be a “state” field with a combination of two-letter state codes, full names and abbreviations. Pre-processing steps could be used to transform all of these names into two-letter codes.

Blocking. The blocking phase is necessary to reduce the computational complexity of pairwise comparisons between records (Christen, 2012). Without blocking, each record would need to be compared with every other record, which becomes computationally unfeasible. Blocking selects groups of records that are likely to represent the same real-world entity for comparison using some knowledge of the data (Herzog et al., 2006). This is typically achieved by using "blocking keys" or "blocking functions" that group similar records together, with the goal of reducing the number of candidate pairs for comparison (Elmagarmid et al., 2007; Herzog et al., 2006). Blocking strategies should maximize the number of true matches selected for comparison while minimizing the number of record pairs that need to be compared (Christen, 2019). An example of a blocking strategy could be to only compare records from within the same state.

Record pair comparison. In the comparison phase, the features or attributes of each record pair are compared and assigned a value (Christen, 2012). Comparison functions are used to quantify the similarity or dissimilarity between two records for each feature or attribute (Elmagarmid et al., 2007). These comparison functions range from exact comparisons and numerical distance measures to more sophisticated string similarity metrics such as the Jaro-Winkler or Levenshtein distance (Christen, 2012). Within the data science community, the use of similarity measures designed to match strings despite slight differences in spelling or different ways of referring to the same thing is often referred to as fuzzy matching (Christen, 2012; Kaufman & Klevs, 2022; Winkler, 1990). While fuzzy matching can be used as a stand-alone term somewhat synonymous with the ER process, within the field of ER, these feature-level comparison functions are part of the comparison phase. The output of this phase is a comparison vector with the resulting similarity score for each feature (Köpcke et al., 2010).

Match/non-match classification. After determining the comparison vector for each record pair, the classification phase assigns a score based on aggregating the results of the feature comparison vector to determine if the records match or do not match (Herzog et al., 2006). Early versions of this comparison phase may create a simple additive score based on summing the (often binary) results of the comparison vector and using a numeric threshold for match or non-match (Christen, 2012). Building on this, machine learning techniques using classifiers like logistic regression, support vector machines, and decision trees began to be used to predict the match status based on the comparison vectors using a labeled training set (Getoor & Machanavajjhala, 2012). The output of the matching phase is a match or non-match decision or a match probability score representing the likelihood of a record pair match (Christen, 2012; Ilyas & Chu, 2019). Much of the recent research focus and interest from the data science community has involved this matching phase (Binette & Steorts, 2022).

Clustering. The process composed of pre-processing, blocking, comparison, classification, and evaluation is standard for what is known as "clean-clean ER." This assumes that in the two sets of records being compared there are no duplicate records, i.e., each individual source is "clean" (Christen, 2012). Many of these same steps are applied to deduplicate records from a data source that may have several records referring to this same entity, known as "dirty ER" (Christen, 2012). In the case that several records may refer to the same entity, a clustering phase is added as a final process to link together small clusters of one or more records with each cluster representing a unique real-world entity or record (Christen, 2012). Hierarchical clustering or graph-based approaches can be used to provide these record groupings to deduplicate matching records (Köpcke et al., 2010; Bhattacharya & Getoor, 2007).

Evaluation. The evaluation phase consists of comparing the results of the matching phase with a known ground truth to determine the model's accuracy (Christen, 2012). On very large datasets, the initial training and evaluation is often performed on a subset of the data to tune the model before deploying to the full-scale data sources (Christen, 2012; Konda et al., 2016). Common evaluation metrics include precision, accuracy, recall, and F1-score, consistent with other application areas for machine learning (Elmagarmid et al., 2007). These metrics can be used to refine previous phases in the process to improve the overall performance. Improvements may include new blocking strategies, additional comparison functions, or improvements to the classification models (Christen, 2019).

While the ER process provides a structured and widely adopted framework to match records across different sources, the workflow is reliant on handcrafted features that need to be developed, tuned, and tested for each new source (Christen, 2012; Elmagarmid et al., 2007). In addition to the manual creation of features, the process also requires calibrated rules for determining match/no-match status for given comparison vectors (Christen, 2012; Herzog et al., 2007). The dependence on manual feature generation and specialized rules for each new data source together with the emergence of new data science techniques for deep learning has led to significant new research in the ER domain (Binette & Steorts, 2022).

Implementations and Benchmarks for Entity Resolution

Figure 2's well-adapted and flexible framework has provided a solid basis for new research in this domain. This section addresses two other key enablers for the adoption of entity resolution techniques: implementation frameworks and benchmark datasets.

Implementation Frameworks. In 2008, Christen provided a full, open-source pipeline for data cleaning, blocking, matching, classification, and evaluation called "Freely Extensible

Biomedical Record Linkage (FEBRL).” FEBRL was a reference implementation allowing researchers and practitioners to experiment with different approaches in a reproducible environment. This shared implementation helped connect academic efforts with real-world applicability to accelerate adoption of novel approaches and reduce the barrier to entry for those new to the domain.

Based on its widespread adoption and utility far beyond its roots in the biomedical field, De Bruin (2019) introduced the $RecordLinkage$ Toolkit, a modernized Python library that builds on the conceptual foundations of FEBRL. The toolkit incorporates advances in data science and integrates with popular Python libraries like pandas and scikit-learn. The simple and modular toolkit can be used within a Jupyter notebook or Python script workflow, making it popular for prototyping, learning, and developing basic entity resolution applications.

Konda et al. (2016) released another powerful open-source project known as *Magellan* to address industry-driven scalability problems with implementation. Based on familiar workflows, the tool provided interfaces for labeling, debugging, and feature engineering and supported the needs of the data science community. Magellan focused on the entity resolution pipeline as a critical enabler for widespread adoption and accelerator for new algorithmic development. Similar to the $RecordLinkage$ Toolkit, Magellan hoped to bridge the gap between new academic approaches and real-world data integration challenges.

Recently, Efthymiou et al. (2023) produced another modular, open-source Python framework called pyJedAI. PyJedAI supports a configurable, end-to-end workflow for entity resolution with an implementation focused on scalability and automation. The authors note four generations of entity resolution based on the “4Vs” of data: veracity, volume, variety, and velocity. The veracity generation follows the workflow of Figure 2 with a focus on deterministic

matching between schema-aligned sources on clean, structured data. The volume generation focuses on how systems like Magellan provided a framework for large-scale processing based on parallelization and MapReduce-style processing. The variety generation provided schema-agnostic tools and includes active learning and graph-based techniques. Their fourth generation, velocity, is focused on current challenges in building streaming and incremental pipelines. Consistent with these generations, pyJedAI is focused on tools that reduce the need for expert knowledge of data, such as active and unsupervised learning as well as a performant implementation that can be run in Jupyter notebooks to prototype and in batch on production pipelines.

These open-source systems for ER have all focused on bridging the gap between academic research and real-world data integration issues. They have enabled the application of state-of-the-art pipelines for practitioners to support production deployments with scalable implementations. They have also reduced the barrier to entry for those new to concepts in the entity resolution framework by providing code that can be accessed through tools like Python and Jupyter following standard patterns.

Benchmark datasets. Benchmark datasets are essential for comparing and reproducing ER approaches. Elmagarmid et al. (2007) first recognized this need as part of their seminal paper. Christen (2008) provided synthetic datasets through FEBRL based on challenges in the biomedical research community. In 2010, the Leipzig University Database Group published four ER datasets (Köpcke et al., 2010). These included DBLP-ACM and DBLP-Scholar, which provided labeled ground truth for matches between different libraries of academic citations. They also included Amazon-Google and Abt-Buy, which matched commercial products between

online retailers. Since their publication, these datasets have provided a benchmark for the development of new ER techniques.

Magellan (Konda et al., 2016) promoted these datasets and included them as part of their open-source framework. By that time, a larger Walmart-Amazon dataset was added to the product matching category and both the Leipzig University Database Group and the Magellan team continued to refresh expanded repositories for benchmark datasets. These datasets emerged during a time when the use of supervised deep learning, which required large amounts of labeled data, was becoming more widespread. These benchmarks were a necessary precursor to future efforts in ER deep learning.

In 2020, Primpeli and Bizer evaluated benchmark datasets from these groups and the Web Data Commons and identified 21 benchmark datasets for profiling. The authors evaluated the datasets based on schema complexity, presence of long textual attributes, missing data, size of labeled training data, and inclusion of difficult cases. This review of the benchmark characteristics provides increased insight into model performance based on these attributes. The authors created partitions for fixed training and test splits and explored class imbalance by creating fixed ratios of positive and negative pairs to support greater reproducibility.

Recently, Papadakis et al. (2023) provided a new critique of current benchmarks and found that the datasets are often "too easy." The authors cited datasets that are too small, are "too clean," or too domain specific. They called for benchmarks with more records and greater variety, as well as multi-lingual datasets. They also cited the need for new benchmarks for streaming and incremental ER. The authors also recommended a potential new focus on platforms that can generate synthetic benchmark data based on configurable settings, community-driven datasets, and shared evaluation protocols.

Labeled benchmark datasets have fueled the development of ER algorithms and served as a foundation for evaluation between researchers and implementations. While progress in ER has been contingent upon these shared datasets, current benchmarks do not capture all the complexities of real-world data and new benchmark datasets can help researchers address remaining and emergent ER gaps (Papadakis et al., 2023).

Modern Approaches to Entity Resolution

Despite the maturity of traditional ER pipelines, these systems relied heavily on handcrafted comparison features and domain-specific tuning (Christen, 2012). This dependence imposed significant overhead for adapting to new datasets or schemas, as each required custom development for cleaning, pairwise similarity measures, and classifier calibration (Köpcke et al., 2010). As machine learning matured, the limitations of manual feature design became increasingly apparent. Supervised learning models like decision trees and SVMs had been adopted to classify record pairs, but they still required well-engineered feature vectors derived from string similarities or numerical distances computed at the attribute level (Konda, 2016).

The rise of deep learning brought a fundamental shift from hand-crafted features to feature representations learned through neural architectures (Goodfellow et al., 2016). Advances in language modeling, first with word embeddings (Bojanowski et al., 2017; Mikolov et al., 2013; Pennington et al., 2014) and then with transformer models (Devlin et al., 2019), changed the way that unstructured and semi-structured text could be represented. These advances have spawned significant new ER research where pairwise feature comparison and classification could be merged into a single process and new areas in representation learning could be explored (Barlaug & Gulla, 2021; Mudgal et al., 2018).

This section focuses on supervised deep learning approaches requiring labeled data. While unsupervised, semi-supervised, and active learning strategies have also seen significant development over the last several years (Barlaug & Gulla, 2021; Genossar et al., 2023; Zhang et al., 2022), the main scope of this review is on supervised deep learning. Progress in supervised learning is reported using the following themes: pre-trained word embeddings, transformer-based approaches, LLMs and sentence transformers, and other embedding approaches.

Pre-Trained Word Embeddings

Kooli et al. (2018) were among the first authors to incorporate word embeddings and combine them with neural architectures. The authors had two use cases for authorship affiliation in research publications and individuals that were part of various professional databases, although neither dataset was made public. They used the standard Wikipedia-trained fasttext library as well as a version trained on the French version of Wikipedia. The authors used both a multilayer perceptron (MLP) and LSTM architecture to produce a binary classifier to determine record pair matches. While the authors reported good performance using this methodology (F1 score of 97.1 on LSTM), they did not provide a repository of the model code or training data.

Also in 2018 Mudgal et al. published the DeepMatcher model which was quickly adopted by the ER community. DeepMatcher was a supervised ER framework using LSTM encoders and attention layers. This was the first deep learning model published on benchmark datasets, including Walmart-Amazon and Abt-Buy, and with a public code repository. DeepMatcher published F1 results for several neural architectures for a large set of public benchmark record pair sets, providing an initial baseline for deep learning approaches to ER. DeepMatcher was not developed for a specific class of data (e.g. matching publications, restaurants, or products), but

rather it showed performance across all use cases. ER approaches began to be developed to work generally and beyond a specific use case.

Around the same time, Ebraheem et al. (2018) introduced DeepER, a deep learning model designed to learn representations for entity resolution directly from raw attribute values. The model used GloVe for attribute-level word embeddings and employed attribute-specific LSTM encoders to generate field representations, which were then concatenated and passed through a feedforward network to predict matches. DeepER was evaluated against Magellan on benchmark datasets including Abt-Buy and Amazon-Google and demonstrated competitive performance compared to Magellan's traditional machine learning baselines. This work demonstrated the potential of using learned feature representations from raw text in place of custom similarity rules for each field.

Transformer-Based Approaches

The transformer architecture (Vaswani et al., 2017) was a significant development for natural language processing greatly improving the model's ability to understand contextual language and outperforming LSTM and CNN architectures. In 2019, Devlin et al. introduced Bidirectional Encoder Representations from Transformers (BERT), based on the transformer architecture and revolutionizing the use of pre-trained language models.

One of the first applications of a BERT-based model for ER was Logeswaran et al. (2019). The authors expressed a single entity as an aggregated string (e.g. "Bombardier BD-100-1A10 Challenger 300") and used flat aggregation with a separator token to encode a record pair (e.g. "Bombardier BD-100-1A10 Challenger 300 [SEP] Bombardier Challenger 350") and encoded the single string with the uncased BERT model as a [CLS] token representation. A final dense layer with a sigmoid classifier provided a match/no-match binary classification for the

string. This paper was not strictly focused on ER, but on general sentence-pair fine-tuning for multiple applications like paraphrase detection, semantic textual similarity, and question answering. This model was a zero-shot model demonstrating BERT's out-of-the-box capability to predict sentence pair matches.

Brunner & Stockinger (2020) were among the first to directly fine-tune BERT strictly for entity resolution. Their method also serialized entire record pairs into a format suitable for BERT's [CLS] classification head similar to the approach of Devlin et al. They also used a simple linear classifier on top of the output embedding as a binary sentence pair classification task. The authors used standard labeled product-matching datasets (e.g., Walmart-Amazon, Amazon-Google) already familiar to the ER community to train the model. They showed that this implementation could outperform Mudgal et al.'s DeepMatcher based on F1 score improvements.

In 2021, Li et al. extended the transformer paradigm for entity resolution by introducing a BERT-based model named Ditto specific for the ER task. Ditto used a similar implementation to the previous BERT-based encoders but added field-aware tokenization. Previous implementation flattened entity records into a single string, while Ditto introduced special tokens [COL] and [VAL] so that schema awareness could be preserved in the strings. For example, a Ditto string could look like "[COL] make [VAL] Bombardier [COL] model [VAL] BD-100 [COL] series [VAL] 1A10 [COL] name [VAL] Challenger 300 [SEP] [COL] make [VAL] Bombardier [COL] name [VAL] Challenger 350." While this string encoding captures all individual column names from the first and second data sources, the two data sources do not need to use a common schema as the model can learn representations through training. This alleviates the need for

schema alignment while providing some schema information to the model and provides a mechanism for the model to handle missing values.

This field tagging allowed Ditto to better distinguish between feature roles across entities. Ditto also introduced several new strategies for ER. The model could accept tags for certain key attributes such as [MANUFACTURER] to declare special tokens. It could also use existing labeled data to generate additional training data by swapping fields to create negative entries, truncating parts of entity records, and replacing synonyms. The model can also summarize long text strings. These provide additional noise in the labeled data so that the model can generalize across domain shifts and avoid overfitting.

Ditto was evaluated on multiple benchmark datasets such as Abt-Buy, Amazon-Google, and DBLP-Scholar, outperforming previous deep learning approaches including DeepMatcher and other BERT fine-tuning baselines, and quickly became a new reference model in the supervised ER literature. Ditto reported F1 score higher on all benchmark datasets with F1 scores as high as 0.987 when matching between two publication data sources DBLP-ACM.

In 2023, Li et al. published a follow-on publication with a refined architecture and enhancements to improve performance on noisy data. The 2023 approach added a cross-attribute attention mechanism that allowed the model to explicitly learn interactions between corresponding fields across entity pairs. The updated model was trained and evaluated on a broader range of benchmark datasets including datasets for “dirty” ER rather than just “clean-clean” ER. This more adaptive and robust transformer-based solution and open-source code base provided a strong implementation for ER practitioners. Ditto-light, a lightweight variant, used smaller sentence transformer models, including distilbert-base-uncased and all-MINILM-L6-v2. These models have a smaller encoder size meaning they can run much faster than the original

model and with less memory. Ditto-light had a reported decrease in accuracy of 1-2% but could run faster on a greater variety of hardware.

Paganelli et al. (2022) did an in-depth examination of the BERT-based entity matching models that are fine-tuned for the binary classification task of identifying matching/non-matching records. The authors examined which layers were adapted by the fine-tuning process and discovered that the fine-tuning primarily adapts the higher layers of BERT, showing that these layers are used to learn complex, context-aware patterns focusing on entity-pair comparisons. If the fine-tuning is focused mainly on lower layers, this would suggest that the learning is focused on the pairwise semantic similarity of tokens. However, by adapting the higher layers, this demonstrates that the fine-tuning attention mechanism is focused on a sophisticated understanding of how to align attributes of the same entity.

The authors also experimented with fine-tuning Sentence-BERT (SBERT) with a simple binary classifier. They showed that SBERT was not as effective as fine-tuning BERT directly for the ER task. The SBERT embeddings were not as effective at learning the relationships between attributes, providing further evidence that the direct fine-tuning enables the model to learn complex, nuanced relationships within the sentence pairs.

To complement these insights, Zeakis et al. (2023) provided an in-depth look at the empirical performance of various pre-trained embeddings for ER. The authors compared the results of pre-trained embeddings (e.g., Word2Vec, GloVe, fastText), versus BERT-based embeddings (like those used in Ditto), and S-BERT across standard benchmarks on the various ER tasks of blocking, unsupervised learning, and supervised learning.

For supervised learning, while the authors used deep learning to perform classification, they provided the binary classifier with each entity's embedding vector independently rather than

embedding the records as a single sentence-pair. The authors showed that the transformer-based embeddings were superior to word embeddings across the benchmarks, supporting the use of these embeddings for models such as Ditto, while not negating the value in the single sentence-pair embedding approach.

These studies focus on transformer-based representations for ER and show how fine-tuning changes model behavior. They represent a logical progression in the use of transformer models for ER, but they remain focused primarily on pairwise ER performance within a single dataset. A related line of work uses deep ER models, including BERT-based encoders, in transfer-learning settings where the central challenge is not only pairwise matching quality but also generalization across domains and sources. A familiar bottleneck in ER is the availability of sufficient labeled training data, so many of these developments focus on improving performance when labeled data in the target domain are limited.

Multi-Source and Cross-Domain Entity Resolution

Jin et al. (2021) extended this transfer-learning perspective to a multi-source entity linkage setting, where only a limited set of sources are well labeled and new sources continue to arrive with little or no annotation. Their framework, AdaMEL (Adaptive Matching and Entity Linkage), is designed to reduce overfitting to the data and label distributions of seen sources and improve generalization to unseen sources. The authors focus on learning transferable matching behavior through an attribute-level self-attention mechanism that models attribute importance during linkage. AdaMEL then uses domain adaptation with abundant unlabeled data from new sources, and it can also incorporate a small labeled support set from the target source in a semi-supervised setting. This work extends pairwise matching into a multi-source transfer-learning setting that addresses label scarcity and source heterogeneity.

A related challenge to formalize domain adaptation for deep entity resolution was proposed by Tu et al. (2022), where a model trained on a labeled source dataset must generalize to a target dataset with few or no labels. The authors introduced DADER (Domain Adaptation for Deep Entity Resolution), a general framework with a Feature Extractor, a Matcher, and a Feature Aligner. The Feature Aligner was added to reduce domain shift between source and target data. The framework supports an unsupervised approach with no target labels and a semi-supervised approach with a small number of target labels. The authors built on transformer-based ER to demonstrate a transfer-learning setting where domain adaptation can improve target domain matching performance when labeled data are limited.

Trabelsi et al. (2022) proposed a multi-source transfer framework for entity matching called DAME (Domain Adaptation for Matching Entities), also designed to improve generalization to a target domain with limited labeled data. The authors argue that separate training and fine-tuning for each dataset can lead to overfitting to a specific domain. DAME treats ER as a mixture-of-experts problem, where source-domain experts are combined with a global shared model trained across domains and a global model-guided attention mechanism. The paper evaluates both zero-shot transfer and target-domain fine-tuning for multi-source domain adaptation built on top of modern deep ER backbones.

A transfer learning framework TransER (Kirielle et al., 2022) was proposed for structured data relevant when source and target domains share the same feature space but differ in their data distributions. The authors frame this as homogeneous transfer learning, assuming the same attribute types and similarity functions across domains, while explicitly accounting for differences in class-conditional distributions for the different sources. TransER identifies high confidence source instances and uses a pseudo-label generator for target instances with a

classifier trained on the high-confidence pseudo labels. TransER targets cross-domain ER in settings where feature-based structured representations are consistent.

Multi-source ER methods are focused on improving generalization across domains and sources under the assumption that labels for the unseen target domain are scarce or non-existent. The next section considers frameworks built to exploit hierarchical relationships in the data structure requiring ER systems to model relationships across entities and support multiple levels of matching granularity.

Hierarchical and multi-granularity ER

Hierarchical and multi-granularity ER addresses the challenge of not only whether two records refer to the same entity, but also how the entity is represented and at what level it should be resolved. Traditional ER methods often assume that entities can be treated as flat records with a single binary match decision, but many real-world datasets contain nested structure, parent-child relationships, or category hierarchies that carry important matching information. In these cases, attribute similarity alone may be insufficient, because correct resolution may depend on structural context and on the intended level of specificity for the task. This has led to a line of work on ER methods that incorporate hierarchical structure directly, as well as approaches that support resolution at multiple levels of granularity rather than a single notion of entity equivalence.

Early work on hierarchical duplicate detection showed that this problem cannot be treated as a simple extension of flat-record matching. Leitão, Calado, and Herschel (2013) demonstrated that duplicate detection in hierarchical data encoded using XML can benefit from modeling both attribute values and the way those values are organized within the hierarchy. Their XMLDup approach used a Bayesian-network formulation to represent dependencies among elements and

attributes, allowing the model to incorporate structural context. This work established the idea that neighboring nodes and parent-child relationships can provide useful evidence for identifying duplicates when entities are related through a larger structure.

Fu et al. (2020) proposed the model Hierarchical Matching Network (HierMatcher), which uses a hierarchy in the matching architecture. The model performs comparisons at the token level, the attribute level, and the entity level instead of a single vector representation for each record pair. At the token level, the model compares records across attributes in case attributes are not cleanly aligned across schemas. At the attribute level, the model seeks to reduce the impact of noisy, redundant, missing, misplaced, or misspelled values by applying an attribute-aware attention mechanism. At the entity-level the model aggregates evidence from the lower levels to produce the match decision. This design is particularly relevant for heterogeneous sources because it treats schema mismatch and dirty data as part of the matching problem itself, rather than assuming that attributes are already aligned or that records are clean. Fu et al. also report strong performance across homogeneous, heterogeneous, and dirty ER benchmarks, showing that multi-level matching architectures can improve robustness when flat pairwise comparison is not sufficient.

Yao et al. (2022) extended this architectural view of hierarchy in a different direction by introducing HierGAT, a Hierarchical Graph Attention Transformer Network for ER. Rather than treating candidate pairs as independent binary decisions, the authors argue that ER often contains interdependence across decisions (e.g., where entities from the same source may be semantically related) as well as interdependence across attributes. HierGAT combines transformer-based contextualized text embeddings with a hierarchical graph attention model so the system can identify discriminative words within attributes and more informative attributes within the entity

pair, while also supporting more collective reasoning across candidate matches. HierGAT complements Fu et al.'s HierMatcher. HierMatcher uses a hierarchical matching architecture across token, attribute, and entity levels, whereas HierGAT uses graph-based dependency modeling and collective ER reasoning across candidate matches.

Genossar et al. (2023) introduced FlexER, a framework for Multiple Intents Entity Resolution that shifts the focus from a single binary decision for a match to task-dependent resolution objectives. The authors argue that many ER systems assume one equivalence criterion for all downstream uses, while real applications may require different intents (e.g., sometimes resolving records at the make-model level while other times at the make-model-series level). FlexER addresses this by formulating ER as a multi-label classification problem across intents and combining intent-specific tuple-pair representations in a multiplex graph processed by a graph neural network. FlexER is not a hierarchical ER model in the structural sense but provides a mechanism for learning and applying different levels of resolution strictness depending on analytic intent.

In addition to neural architectures, some work has focused on defining hierarchical ER tasks and evaluation protocols in applied domains. Kouki et al. (2019), for example, studied collective entity resolution in multi-relational familial networks, showing that leveraging relationship structure can improve resolution compared to treating record pairs as independent. Backes et al. (2022) make the granularity issue explicit in their work on hierarchical affiliation resolution. They define an unsupervised task in which affiliation strings are resolved to institutions across multiple hierarchy levels (e.g., department, faculty, university), and provide a conceptual framework, baseline methods, datasets, and evaluation metrics for studying the problem. Their results show that some subproblems, such as mapping affiliations to known

institutions or identifying lower-level institutions, are more tractable than full unsupervised top-level and hierarchical resolution. This work is useful in a hierarchical ER review because it shifts attention from model architecture alone to domain understanding and task formulation.

Complementing work on transformer-based and multi-source ER, hierarchical and multi-granularity ER explores how structural context and task-specific resolution levels affect matching performance. Hierarchical and multi-granularity ER is especially relevant to the aircraft type matching problem because records may need to be resolved at different levels of specificity, providing a potential solution for records expressed at different levels of granularity. The final section of this Modern Approaches to ER review considers recent developments using large language models.

Large Language Models

Large Language Models (LLMs) have introduced a new direction in entity matching research by shifting attention from specialized task-specific architectures toward more general-purpose generative models that can operate in zero-shot, few-shot, and in-context learning settings. Peeters, Steiner, and Bizer (2025) show that LLM-based matchers can perform competitively with fine-tuned pre-trained language model baselines with little or no task-specific training. The authors found that LLM-based entity matching is highly sensitive to prompt design and found that no single prompt performed best across all model and dataset combinations. Initial success suggests that LLMs may reduce the need for large, labeled datasets for ER, with the caveat that performance is currently tied to prompt formulation. The authors evaluated LLMs that were accessible through APIs, where usage may incur token-based fees. This has contributed to subsequent work on the cost-effectiveness of LLM-based entity matching.

This line of work has attracted significant attention because it suggests that EM systems may become less dependent on large, labeled datasets, even as the practical behavior of LLMs remains strongly tied to prompt formulation and deployment choices. Steiner, Peeters, and Bizer (2024) used smaller LLMs fine-tuned for ER. On these smaller LLM models, fine-tuning provided substantial performance gains, where the results for larger LLMs were less conclusive. The fine-tuned LLMs also reduced cross-domain transfer performance. The authors also explored structured LLM-generated explanations in the training set which also improved performance.

As many LLM approaches are based on hosted models with incurred token costs, recent work has also focused on cost-effectiveness and design choices rather than accuracy alone. The BATCHER framework by Fan et al. (2024) is a cost-effective in-context learning framework for ER. BATCHER studies record selection and question batching with the goal of balancing matching accuracy against monetary cost in batch prompting. Wang et al. (2025) proposed the ComEM framework that introduces the strategies “match,” “compare,” and “select” to improve cost-effectiveness. Similarly, Li et al. (2024) propose BoostER, an uncertainty-reduction framework that selectively queries an LLM on high-value matching questions to verify candidate links and iteratively refine the ER outcome, while using pair-selection and error-tolerant mechanisms to control API costs. These studies highlight that while LLM-enabled ER is growing in interest, considerations remain for token costs and deployment complexity.

A final related paper of interest is the Jellyfish model (Zhang et al., 2024) that groups several data preprocessing tasks into a single problem that can be solved using local instruction-tuned LLMs trained for a variety of data preprocessing tasks. This work shows a potential

alternative for production pipelines to include smaller specialized models fine-tuned for specific data preprocessing tasks.

This study did not include recent progress in LLMs due to these token charges and the desire to first develop baselines in traditional approaches. However, future directions for ER will likely include increased use of LLMs.

Modern Approaches to ER Summary

Recent advancements in ER have shifted the field from traditional rule-based and feature-engineered pipelines to deep learning models capable of learning rich, contextual representations of record pairs. Early progress with pre-trained word embeddings and LSTM architectures was followed by transformer-based models such as BERT, which improved ER performance by learning attribute relationships and contextual similarity directly from serialized record pairs. Within this line of work, Ditto introduced ER-specific tokenization and augmentation strategies that improved performance across standard benchmark datasets and established a strong transformer-based baseline for supervised ER.

Recent work has extended these gains beyond single-dataset pairwise matching by addressing cross-domain and multi-source settings, where transfer learning and domain adaptation are used to improve generalization when labeled data are limited. In parallel, hierarchical and multi-granularity ER has expanded the scope of the problem by showing that many applications require models that account for structural context or support different levels of resolution depending on the task. Most recently, LLMs have gained attention as a potential addition to ER pipelines, although their use can involve token-based usage costs and added deployment complexity. These trends show that modern ER is moving beyond pairwise matching alone to explore how to apply models in domains with little or no labeled data, how to

manage hierarchical information within the entities, and how to exploit recent developments with LLMs.

Summary

This literature review examined the intersection of aviation safety analysis, aircraft type taxonomy, and entity resolution to identify critical gaps in multi-source data integration for aviation data. It began by contextualizing the importance of aircraft type in continued operational safety, showing how the risk assessment process relies on precise fleet definitions informed by structured taxonomies. Despite international standards such as the CICTT taxonomy, inconsistencies in how aircraft type is expressed across data sources limit the integration and analysis of aviation safety records. While natural language processing and knowledge graphs have advanced in the aviation domain, few studies have addressed the challenge of aligning heterogeneous aircraft type data.

The review then examined the evolution of entity resolution, covering traditional and modern approaches as well as benchmark datasets and open-source implementations. Emerging deep learning models such as Ditto show significant promise for addressing schema mismatch and unstructured inputs. Current gaps in aviation-specific data integration motivate this study's aim to curate a ground truth aircraft type dataset and evaluate modern deep learning ER models on a new domain benchmark. By introducing this benchmark, the study aims to generate evidence on model performance in an aviation-specific setting and support further development for safety data integration and ER more generally.

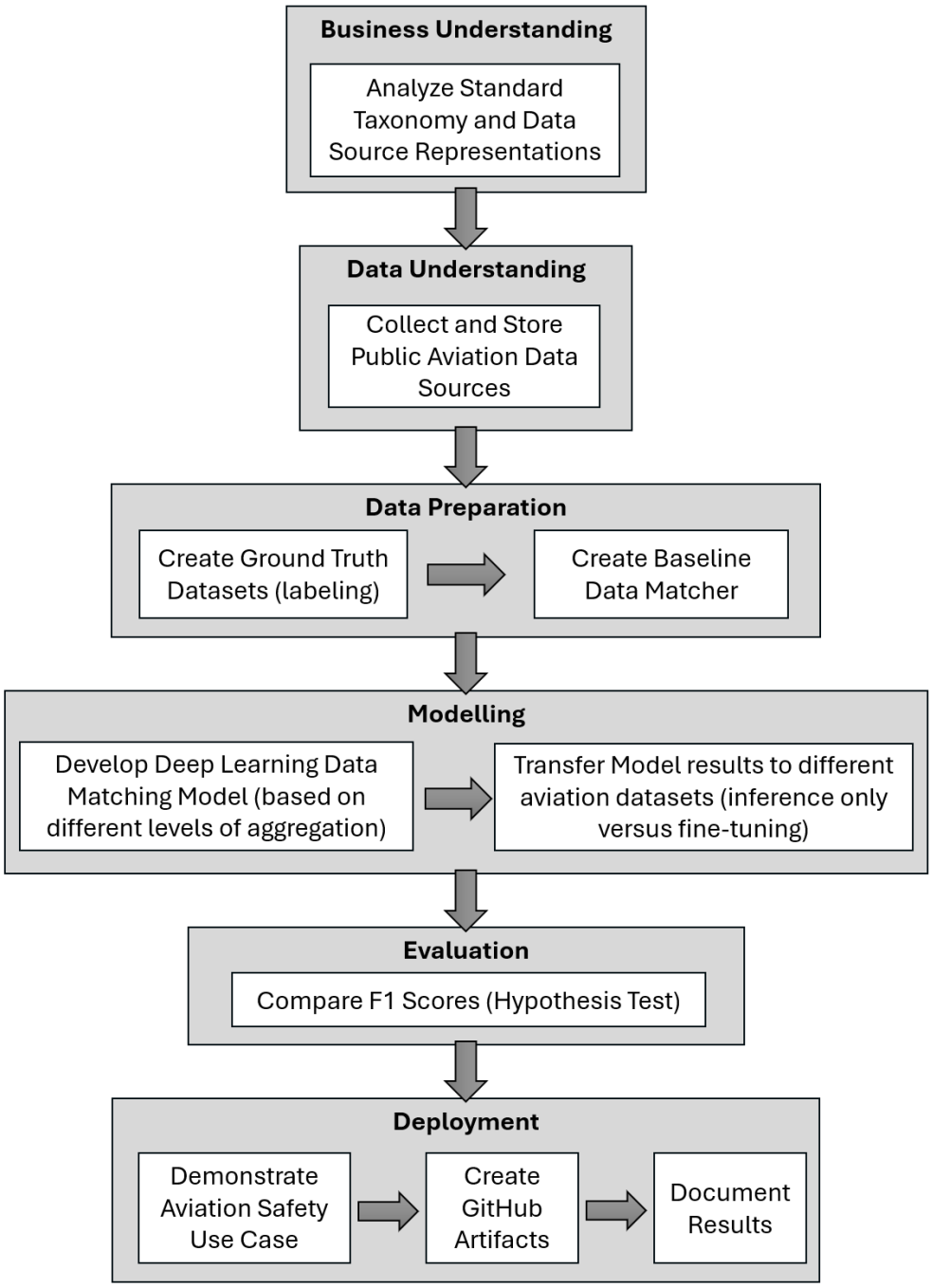
Chapter 3: Research Method

Linking records across different data sources without common identifiers presents a common challenge in data integration across various domains (Christen, 2012; Ilyas & Chu, 2019), and aviation is no exception (Blom, 2010; Boyd, 2017). The problem addressed in this study is the challenge of merging aviation data from diverse sources that lack common keys, hindering safety analysts' ability to assess risk using multiple information sources. The purpose of this constructive research study is to develop an entity resolution model that can connect disparate aviation databases by matching across data sources using aircraft make-model-series specifications.

This study adopts a constructive research methodology to create a tailored solution for matching aircraft types across disparate data sources. The study is based on the CRISP-DM for overarching structure, while using an entity resolution framework within this model to implement the best solution. Figure 3 provides the overall process flow of the study. During the business understanding, data understanding, and data preparation phases, a curated multi-source "gold" dataset will be constructed to guide the modeling process. The modeling phase will both examine the suitability of deep learning and transfer learning for aircraft type data matching and use these findings to construct a general-purpose model for aircraft type matching.

Figure 3

Process Model based on CRISP-DM Framework



The final stage of the study will be to demonstrate the maturity of the developed techniques through an aviation safety use case. This study is based on three different types of data sources: those with safety event information, those that provide estimates of flight hours or aircraft counts as utilization proxies, and taxonomy data sources. This phase will demonstrate how safety event data can be normalized using utilization proxies when aggregated through a taxonomy. The deployment phase will also produce relevant artifacts on GitHub, including the gold standard dataset. This new data source artifact will be a multi-source labeled dataset for entity resolution which can be utilized by researchers working in this domain.

Research Methodology and Design (Nature of the Study)

This study uses a constructive research methodology designed to develop an innovative solution to a real-world challenge (Brendel et al., 2022). Such a methodology offers the adaptability needed to create artifacts and models tailored to the specific requirements of the aircraft type data matching task. The CRISP-DM process model provides the structured approach used to create a model for matching aircraft type, while the entity resolution domain provides the technical underpinnings for the solution.

The constructive research methodology is appropriate because the overarching goal of this study is to create a set of artifacts, including a predictive model, which enables safety analysts to match aviation data sources by aircraft type. Alternative methodologies such as a purely quantitative study were considered. While a quantitative approach could support statistical analysis of predefined hypotheses using aviation datasets, it was not selected because it would not allow for the iterative development and refinement of artifacts that are central to this study. This solution is driven by the challenges faced by aviation safety analysts, and the more refined details of the solution will be determined based on the practical realities of aviation type data.

The entity resolution framework has been selected as an enabler for this solution because it is in widespread use for other data matching problems, such as matching products across two internet retailers.

The first step in the CRISP-DM model is the business understanding phase, where the different data sources will be evaluated to understand commonalities and differences in how aircraft types are described and used across sources. The study is based on nine publicly available data sources that fall into three categories for the purposes of this study: safety event data, utilization data, and taxonomies. These data sources will be further described below in the population and sample section.

The data understanding phase of the CRISP-DM assesses the compatibility of the various information sources and what additional metadata may be available across multiple sources. In the data understanding phase, each data source will be further analyzed with respect to aircraft type to understand whether consistent levels of granularity are used, whether data entry was free-form or based on standardized types, how columns align between sources, and what other relevant metadata may be useful for analysis. In this phase, the data sources will be analyzed to create a subset of the data for analysis. The data understanding will help guide model development by providing further insights into the data quality in each data source and highlight potential challenges.

The data preparation phase includes the critical step of building a “gold” reference dataset, which will be used to train and assess various entity resolution models. Of the nine datasets, three datasets can be linked through common data fields. The FAA taxonomy table contains links through one column to the FAA Operator utilization table, and through another column to the FAA aircraft registry. This means that links between these tables can be built

through database joins or similar techniques. After this, an intentionally selected purposive sample will be used to match the taxonomies with the AIDS, NTSB, NWSD, and BTS datasets so that these may also be used to evaluate model performance using metrics such as the F1 score. Not all datasets will be linked to the most granular level of the taxonomy if this level of definition is not available in the dataset.

The result of the data preparation phase will be a curated link across all nine data sources with at least two levels of granularity to align with those present in the data sources. This multi-source dataset will support the evaluation of schema-agnostic approaches, allowing a model trained on two data sources to be applied to a third. The models will not be given knowledge of the curated links between the sources, but this multi-source table will be used to calculate metrics that demonstrate model performance.

The modeling phase of the process includes two distinct phases. First, the study will train the Ditto model on the curated taxonomy-to-taxonomy dataset. This training will happen at multiple levels of granularity. Aggregate levels will not require additional labeling but will exploit the hierarchical nature of the taxonomies. The datasets all have many different feature columns which will be combined into a single string representation for each entity. The model will be tested using different combinations of columns. The second phase of modeling will take the model trained on the clean taxonomies and apply these to the other aviation datasets. These will be evaluated on a subset of labeled data for each of these datasets. The second phase will compare inference only with the model tuned on the taxonomies with a fine-tuned model built on this taxonomy-tuned model.

The evaluation phase will evaluate the performance of the various approaches using the labeled data. The F1 score will be used to evaluate performance. Because the “gold” data source

will be linked between all nine sources, it will be possible to train the data using the taxonomy sources, then run the model on a different source. This will help refine the model towards a general-purpose model for aircraft type matching that does not need to be re-trained for each new source. Consistent with the NFLT, different models will perform differently on aircraft type data than on other sources in the literature. The performance of these different approaches on aircraft type data will inform the general-purpose model to be included in the dissemination phase.

The final stage of the study will demonstrate the maturity of the developed techniques through an aviation safety use case, drawing on three types of data sources: safety event information, utilization proxies (e.g., flight hours, aircraft counts), and taxonomy data. This phase will illustrate how safety event data can be normalized using those utilization proxies when aggregated through a taxonomy, providing a demonstration of the general-purpose model on the safety event normalization case. The deployment phase will additionally disseminate artifacts on GitHub, including the multi-source gold standard labeled dataset for aircraft type and the general-purpose model. The published dataset can support other researchers in entity resolution, while the general-purpose model can be used for aviation safety research use cases.

Population and Sample

The general population addressed in this study is the set of all the aircraft type specifications used to describe aircraft registered and manufactured in the United States. The distinction between aircraft and aircraft type is that an aircraft is a specific real world entity with a registration number and serial number and an aircraft type is the specification of the aircraft. Aircraft type is formally described in a type certificate data sheet (TCDS) and has all of the characteristics of the aircraft (De Florio, 2016). A single aircraft type can have hundreds or even

thousands of aircraft, or it can be a specification for which only a single aircraft was built, or even one that was never built or no longer exists.

This study uses nine public aviation data sources that include aircraft type taxonomies, safety events, and aircraft utilization sources. A list of these sources with their respective row counts and unique aircraft types is shown in Table 3. While the theoretical general population is defined as the set of all aircraft type specifications used in the United States, this study defines its accessible population as the collective body of aircraft type data contained within the nine publicly available aviation data sources listed in Table 3. These data sources represent the population of aircraft types as recorded in U.S. civil aviation systems across multiple domains and use cases. While many of these data sources contain aircraft-level data, the population of interest for this study is defined at the aircraft type level, not the individual aircraft registration level. This approach aligns with the study's objective of evaluating entity resolution techniques for matching aircraft types across heterogeneous datasets to enable safety event normalization. While it would also be possible to use entity resolution techniques to match aircraft across data sources, this is not the scope of this study.

Table 3

Summary of Aircraft Type Data Sources by Category

Data source short name	**Data source full name**	**Number of table rows**	**Unique aircraft types**
	Taxonomies		
CICTT	ICAO Commercial Aviation Safety Team/ICAO Common Taxonomy Team (CICTT)	23,455	23,455
ICAO	ICAO publication *DOC 8643 - Aircraft Type Designators*	7,320	2,702

Data source short name	Data source full name	Number of table rows	Unique aircraft types
FAA	FAA Aircraft Reference Table for Aircraft Make Model Series (avinfo.gov)	8,027	8,027
	Safety Events		
AIDS	FAA Accident and Incident Data System	32,718	6,212
NTSB	National Transportation Safety Board	28,227	6,612
NWSD	National Wildlife Strike Database	303,632	601
	Utilization Sources		
OPER	FAA Operator-Aircraft table (avinfo.gov)	12,484	1,231
REG	FAA Registry	298,355	92,132
BTS	Bureau of Transportation Statistics - U.S. Domestic Commercial Flights	5,652,662	444

Note. Data sources are publicly available from the FAA, ICAO, NTSB, and the U.S. Bureau of Transportation Statistics. "Unique aircraft types" refers to distinct aircraft type representations found in each data source.

The taxonomy data sources include a row for each aircraft type description aligning with the study population. The CICTT taxonomy has 23,455 unique aircraft type entries. The *Doc 8643* publication has 7,320 rows but only 2,702 unique four-letter ICAO type designator codes. This is because several metadata fields can map to the same ICAO type designator. For example, for the ICAO type designator "CL35" contains rows for both the "BD-100 Challenger 350" and "BD-100 Challenger 3500." Because the CICTT taxonomy contains a column for the ICAO type designator, these two tables can be linked with this common field so that information in the *Doc 8643* taxonomy can be treated as an extension of the CICTT taxonomy. For this reason, the CICTT and ICAO taxonomies can be treated as a single table in this study. The FAA taxonomy

contains 8,027 rows and there are no fields that can be used to link the FAA taxonomy with the CICTT or ICAO taxonomies.

The AIDS, NTSB, and NWSD sources contain aviation safety accidents, incidents, and occurrence data collected by different agencies and for different purposes. Each of these have at least one field for aircraft type. The number of aircraft types observed in these sources is 6,212, 6,612, and 601 respectively. The reason that there are fewer aircraft type entries in the NWSD is because these aircraft types are reported at a higher level of aggregation. For example, they may be reported at the "make-model" level rather than a "make-model-series" level.

The FAA Operator-Aircraft table does contain the FAA Registry model manufacturer code which is also a field in the FAA taxonomy. Thus, this data source can be automatically linked with the FAA taxonomy. This table provides rows at the aircraft level for 12,848 aircraft, which, when condensed to the aircraft type level contains 1,231 aircraft types. While the FAA taxonomy contains over 8,000 entries for aircraft type, there are only around 1,200 aircraft types that are currently used in regulated commercial operations.

The FAA taxonomy also links to the FAA Registry via a manufacturer model code, but there are far more entries for manufacturer model specifications in the registry than in the taxonomy. This is because of many Registry aircraft types that are not tracked in the taxonomy, generally because the code only represents a single aircraft which is not operated commercially. For non-type certified aircraft such as those built by a hobbyist, the owner's name is often used as the manufacturer. This means that there are many additional aircraft type specifications in the registry which do not provide a meaningful way to aggregate aircraft because the type is only relevant for a single, specific aircraft.

The FAA Registry contains 298,355 different aircraft and has a separate table with 92,132 aircraft type descriptions. The registry aircraft type descriptions contain a single column for both model and series and do not enforce standardization. For example, there are entries for "BD-100-1A10" with both "Bombardier Aerospace Inc" and "Bombardier Inc" listed as the manufacturer. While there is a hyphen to delineate between the "model" and "series", there is also a hyphen in the model "BD-100" which contains abbreviated manufacturer information. There is also an example of a "Challenger 601-3R" and a "CL-600-2B16(CL601-3R)" demonstrating the challenges of using any regular expressions or automation to extract model and series information from these entries.

In the ER framework, "clean-clean ER" traditionally refers to the matching of two "clean" datasets where duplicate records are not expected in each dataset (Dong & Srivastava, 2015). This means that there should be a one-to-one relationship between potentially matching entries in the two datasets, even if a significant portion of each dataset may not overlap with the other (Christen, 2012). "dirty ER" refers to deduplication of a dataset where several rows may contain the same entity (e.g. "Bombardier Aerospace Inc BD-100-1A10" and "Bombardier Inc BD-100-1A10"). The "dirty ER" problem includes the optional clustering step to merge duplicate records (Elmagarmid et al., 2007).

The sample that will be used for this study uses combination of a non-statistical based purposive and convenience sampling (Etikan et al., 2016). Purposive sampling is used to study a specific population with the ability to purposefully focus on a specific segment of a population. Convenience sampling is used when a researcher does not randomly sample a population but uses the first available or conveniently available subjects. Each of these approaches provides challenges for generalizability in statistical hypothesis testing. However, given that this

constructive research methodology is focused on model development and demonstration, the sufficiency of the sample size for training and evaluating machine learning models is justified by its alignment with benchmark dataset sizes in the ER field, rather than a traditional power analysis focused on statistical inference to a general population.

For the modeling phase of this study, the first phase will be to train Ditto on a “clean-clean” dataset, matching the CICTT-ICAO taxonomy to the FAA taxonomy. An aircraft type taxonomy conceptually has one row per aircraft type to a specified level of granularity. The convenience sample for matching records will be taken from the manually mapped “gold” dataset that matches these two taxonomies where the aircraft type is in both datasets and a matching relationship can be established through manually data linking.

The number of matched rows established between the two datasets for manual mapping for the gold dataset is roughly 2,700 entries. As the CICTT taxonomy has over 23,000 rows and the FAA taxonomy has over 8,000 rows, this sub-selection was created from a purposive and convenience approach. The initial purposive sample for the "clean-clean" gold standard was constrained by the need for verifiable matches and constraints due to the feasibility of additional manual labeling.

The entries used to determine the mapped pairs were down-selected to include larger aircraft that are currently used where there are currently registered aircraft in the FAA registry. This excludes many aircraft no longer in service or aircraft that do not fly commercially or are not type certificated. While there are some non-type certificated aircraft in the sample and there are some types that do not have registered aircraft, the sample was selected to purposefully select those aircraft types that do have large numbers of registered aircraft and do fly commercially.

The sample also has an aspect of convenience sampling because the sample is based on the most prominent aircraft type when performing a manual selection process.

The full train/validate/test datasets will contain splits from all of these matching records plus sampled non-matching records at a roughly 10:1 ratio. This means that for every matched record ten non-matching records will be selected randomly from the remaining record pairs. A Python script will be used to generate the non-matching records and to divide the records into train, validation, and test subsets.

Benchmark datasets used for ER have a varying number of matching records (Primpelli et al., 2020). Most of the original Magellan datasets published around 2016 have between 68 and 130 matches and around 300 non-matches in the labeled dataset. The datasets in the Leipzig Database Group have been expanded to include between 1,000 and 5,000 matches. As ER algorithms have been demonstrated using these datasets with no more than 1,000 matching records, it is assumed that the ~2,700 records identified as a match between the two taxonomies will be sufficient to demonstrate good performance.

Because a taxonomy also provides a hierarchy for aggregation, aggregations of a taxonomy can have fewer rows that roll up in a more aggregated view. These aggregated levels can be used to run the model to match for an aggregated level (e.g. matching for make-model instead of make-model-series), but the mappings are already available from the de-aggregated data since the same aggregation is used on both taxonomies.

Of the datasets in Table 3 those that do not have fields that link them to one of the two taxonomies are AIDS, NTSB, NWSD, and BTS. Of these datasets, a smaller sample will be manually mapped in order to derive appropriate statistics as matches between these datasets and taxonomy entries are predicted. These datasets can also be used to fine-tune the model for these

specific datasets. The labeling for these datasets will be a convenience approach and focused on an aggregate level. As there are many records with a significant level of "dirtiness" manual labeling of all of these ancillary data sources would not be feasible.

Instrumentation

For this study, the primary instrument is an ER software model that can take multiple aviation data sources and make a binary classification decision on pairwise records for a match or non-match. In this case, the open-source Ditto model has been selected due to its acceptance in the literature as a state-of-the-art model and its extreme adaptability. Ditto processes entity information using a generic string representation (e.g. "COL col_1 VAL val_1 COL col_2 VAL val_2 ... COL col_n VAL val_n"), which allows the model to adapt to any combination of input columns.

This adaptability means that no schema alignment is strictly required, and two sources do not necessarily have to use the same input columns. It also means that Ditto can be used as an instrument to compare the results of different input feature sets by changing the entity input strings to determine which features are most effective. Ditto will be used in different configurations by varying input features and at different levels of hierarchy using the appropriate training labels for each hierarchy. This means that F1 scores can be compared across the different experiments to determine an optimal configuration for the aviation domain.

Ditto is based on a pre-trained transformer model using a BERT-based encoder with a final binary classification output layer (Li et al., 2021). Ditto treats ER as a sequence-pair classification problem by encoding both entities into a single input string. Using labeled data, Ditto is able to learn if two entities are the same through their concatenated textual encodings. Ditto has been used with multi-source data and at different levels of granularity making it well-suited to examine different configurations for ER on aircraft type.

The primary computing environment is a local workstation with an Intel® Core™ i9-13900KF (13th Gen) processor, 32GB DDR5 RGB 5200MHz of Random Access Memory (RAM), and a NVIDIA GeForce RTX 4070 12GB Graphics Processing Unit (GPU). All Python-based workflows, including library installations, script execution, and model training, are conducted within a Windows Subsystem for Linux (WSL) instance. This environment provides greater compatibility with several deep learning libraries compared to native Windows. Visual Studio Code (VS Code) will be used as the primary integrated development environment (IDE) for code editing and debugging. The open-source Python programming language will be used to develop the overarching pipeline. This choice is based on the scope of existing specialized libraries and widespread adoption within the data science community.

Beyond Ditto, several other open-source Python libraries are utilized to support various stages of the ER workflow, including data preprocessing, exploration, and alternative model comparisons. Open-source core data analysis and manipulation libraries that will be used are pandas, NumPy, and SciPy. Machine learning libraries that will be used are PyTorch, scikit-learn, and transformers. Natural language processing libraries include gensim, sentencepiece, spacy, nltk, and regex. Additional utility tools include tqdm and jsonlines.

To build the baseline data matcher that will be used as a reference, the open-source *RecordLinkage* project (MIT license) will be used. Other open-source libraries that may be considered as an alternate are pyJedAI, Magellan, and dedupe. *RecordLinkage* was selected because of its transparency with the creation of feature-wise matching rules and classifiers and its collection of fuzzy-matching rules already included in the library. Dedupe is available as a similar type of Python package but with slightly less functionality. PyJedAI and Magellan are both robust Python wrappers for performing entity matching, but their more automated code

structure makes them less straightforward for implementing simple and transparent rule-based comparisons.

For web scraping and automated data downloads, both R and Python will be used. R will be used in RStudio with knit notebooks for code that was developed prior to this study. Python with Selenium and BeautifulSoup will be used for more recently developed code.

A significant amount of data exploration and initial data cleaning is required to understand which datasets have similar features and matching columns, as well as an understanding of how many rows are in each dataset. Much of this initial data exploration will be done in a combination of Tableau Desktop and Tableau Prep. While proprietary software, both have a free student edition that can be utilized for this study.

All data preparation scripts, model training runs, and evaluation workflows used in this study are available in the project repositories (Bryan, 2026a; Bryan, 2026b), and detailed implementation steps, file layouts, and execution instructions are documented in Appendix A. The repositories include configuration files and scripts used to generate candidate pairs, construct train/validation/test splits, train Ditto checkpoints, and compute all reported metrics. Together, these resources provide a complete record of the as-executed workflow sufficient to reproduce the reported results under the same datasets and evaluation framings.

All open-source software and libraries utilized in this research are employed in accordance with their respective licenses, and no specific permissions beyond standard public access are required.

Operational Definitions of Variables

This section describes the dependent and independent variables for the study. The main dependent variable for the study is the F1 score together with precision and recall. The

independent variables in the study are the input feature sets, the level of hierarchy, the data sources, hyperparameters, and the baseline ER capability developed using a simple matching scheme.

F1 Score, Precision, and Recall

To measure the effectiveness of the central instrument, the Ditto ER model, the primary metric for this study and the dependent variable is the F1 score. This metric is commonly used in machine learning classification tasks, particularly where there is a cost imbalance associated with false positives and false negatives (Sokolova & Lapalme, 2009). In entity resolution, where the goal is to maximize correct matches while minimizing incorrect ones, the F1 score proves particularly effective due to its inherent ability to balance these two competing goals.

The F1 score is derived from two fundamental metrics, precision and recall. Precision is defined as the proportion of the model's predicted matches that are indeed correct matches. Recall is the proportion of all actual correct matches that are successfully identified by the model. Together, these metrics balance the model's ability to identify true matches against the risk of either over-prediction from false positives, or under-prediction from false negatives.

To define these metrics more formally, we first categorize the outcomes of the Ditto model's binary classification decisions against the "gold standard" ground truth. These are all count variables, representing ratio-level measurements:

- True Positives (TP): The number of record pairs that are correctly identified by the ER model as a match and are, in fact, true matches in the "gold standard."
- False Positives (FP): The number of record pairs that are incorrectly identified by the ER model as a match but are actually non-matches in the "gold standard" (a Type I error).

- False Negatives (FN): The number of record pairs that are incorrectly identified by the ER model as a non-match but are actually true matches in the "gold standard" (a Type II error).

Then, precision, the proportion of predicted matches that are correct, and recall, the proportion of actual correct matches that are identified, are calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

The F1 score is then computed as the harmonic mean of precision and recall:

$$\text{F1} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{(\text{Precision} + \text{Recall})}$$

The harmonic mean is especially valuable for averaging ratios because it gives more weight to smaller values and less to larger ones. This characteristic ensures that the F1 score effectively balances precision and recall, penalizing models that perform extremely well on one metric while performing poorly on the other, thus preventing extreme biases in the matching performance.

Input Feature Set

Input feature sets are a key category of independent variables. The only input feature to the Ditto model is a concatenated string of the entity-pair sequence, so the exact features presented to the model will be constructed in a pre-processing step that manipulates the contents of this concatenated string. Each model run will depend on its feature set and labels and referred to with an index (e.g., M1, M2, M3) for model runs based on different feature sets. This feature set includes the use of different combinations of columns from the original datasets, such as make, model, series, or combined fields representing model-series as a single feature. It also

includes various additional fields like type certificate, ICAO type designator, popular name, number of engines, number of seats, and other available metadata.

The availability of this metadata will vary across input data source. The main two taxonomy sources have a relatively consistent set of features that will be used to train the initial taxonomy-to-taxonomy model. Part of this training can be "learning harder" by training on examples which mask certain known features in the feature string (Li et al., 2023). Each selection of input feature combinations selected for model runs will be documented and labeled as a unique model run. An exhaustive set of these input feature parameters by source are provided in Annex A.

Level of Hierarchy

Levels of hierarchy represent another independent factor. This study evaluates matching at different levels of granularity, including comparisons at make-model and make-model-series level. Two other aggregations will include grouping by master make model and by using the master series instead of the series. These levels of aggregation in the taxonomy were designed for the purpose of different analytic groupings. As the goal of this study is to aggregate aircraft by analytic groupings based on aircraft type, these different levels in the taxonomy will be explored.

These different levels of hierarchy will be evaluated as separate models and displayed in a grid against the feature string models. While each hierarchy has its own match/non-match label, only one labeled dataset is required as the labels for each level of hierarchy can be inferred from simply aggregating the taxonomy to the appropriate level. The level of hierarchy is designed to create separate models that are specialized to a particular level of aggregation.

Data Source

To evaluate RQ2, the performance is based only on the highly curated and fully labeled taxonomy-to-taxonomy dataset. Only the two main combined taxonomy data sources will be used for RQ2. For RQ3, the study will address how the model performs against dirty datasets that do not have all the features of the input dataset. This will be achieved in two ways. First, by inference based on the model trained on the taxonomy-to-taxonomy dataset and secondly by fine-tuning the model for these datasets with dirtier data.

The performance of the original taxonomy-to-taxonomy model features sets may differ for the dirty in-domain datasets than for the original model evaluation. This means that the evaluation of a model may depend both on its performance against the gold standard dataset as well as how it performs against an in-domain dataset that was not originally used to train the model.

Hyperparameters

Some Ditto hyperparameters may also be manipulated and could be treated as another independent variable. These include batch size, maximum token length, learning rate, number of epochs, and transformer settings such as the pre-trained language model (e.g., RoBERTa, distilBert), data augmentation, and mixed-precision training. While hyperparameter tuning is not the main emphasis of this study as it is anticipated that a standard set of hyperparameters will be sufficient to run the Ditto model, if needed these will be systematically adjusted. These parameters are defined in the Python codebase and can be adjusted to evaluate their impact on model performance.

Baseline ER Capability

In addition to Ditto, the baseline ER capability using the $RecordLinkage$ library will be evaluated with the same F1 metrics as a baseline. This baseline model will require its own set of features as the model uses a more standard framework than Ditto's which concatenates all information into a single string that can be encapsulated by the family of BERT embeddings. As the goal of the study is to evaluate the capability of the BERT-based embeddings and deep learning, the use of the $RecordLinkage$ library will not include highly tuned and detailed custom rules incorporating many special cases. This baseline is used merely as a reference point for the use of the deep learning algorithms in a new domain.

All of these settings are implemented and controlled through Python scripts, which serve as the experimental instrument. This instrument automates model training, evaluation, and logging of configuration-specific performance metrics.

Study Procedures

The study will be conducted in a series of structured, reproducible steps that align with the CRISP-DM framework and follow the process model from Figure 3. This section further refines the Data Preparation, Modeling, and Evaluation steps into a more refined logical progression.

The following procedures are designed to produce a curated dataset, train and evaluate a deep learning model, and test its ability to generalize to previously unseen aviation datasets. The sequence below outlines the detailed procedures for preparing the data, training models, and conducting inference and fine-tuning, directly supporting RQ1 through RQ3.

Step1: Analyzing Aircraft Taxonomies

This step involves the qualitative analysis of various data sources using aircraft type and documentation surrounding aircraft type. This step supports the literature review on aircraft type and guides the next step on the collection of public data sources for aircraft type.

Step2: Collecting and Storing Data

This step involves the collection and storage of the nine selected public datasets. This step is addressed through a combination of code written in R and Python. Unless the data source is a single downloadable file, each data source is systematically downloaded or scraped using reproducible code so that any new data updates can be incorporated by re-running the existing code. Frequent updates are not needed as aircraft type data is relatively constant over time. The data will be backed up using a GitHub repository and stored as files rather than in a relational database structure.

Step 3: Curating the Gold Dataset

This is the first step in the Data Preparation phase of the CRISP-DM and involves the creation of a high-quality labeled dataset for training and evaluation. This curated dataset serves as the ground truth against which model performance will be evaluated, meeting the standard of a “gold” dataset through exhaustive manual validation and expert judgment. This step will be an iterative task of analysis and curation of the combined datasets.

As a first step, analysis in Tableau Desktop and Tableau Prep will be used to determine how data sets overlap and align and which share common fields. Tableau is useful to quickly explore counts in different cross tabulations of the data and understand the columns needed to get to the most granular row of data. Tableau supports a quick understanding of data quality and row counts for aggregated data.

Tableau Prep will be used to join data sources that have common identifiers. The tool will be used to experiment with different join types to understand where different sources overlap. It will also be used to create comma delimited files for manual data labeling.

The manually matched "gold standard" dataset will be produced in Excel by linking identifiers from the two primary aircraft taxonomies: the FAA taxonomy and the combined ICAO CICTT/Doc8643 taxonomy. These two sources are treated as “clean” datasets, meaning each entity is assumed to be distinct and not duplicated within a source. Manual annotation will be guided by aircraft type metadata and supported by additional visual inspection tools. This process will use metadata from the two taxonomies to ensure a good match. For example, the type certificate data sheet should match between the two sources even if the main analysis is on matching the manufacturer, model, and series. Figure 4 shows an example of the matching process.

Figure 4

Creation of the gold dataset using CICTT and FAA Taxonomies

CICTT Taxonomy

	inde	cictt_string	cictt Organization Common Name	cictt Acft Master Model Name	cictt Acft Make Name	cictt Acft Model Nam	cictt Acft Series Name	cictt Acft Popular Name	cictt Type Certificate Name	cictt Icao Acft Type Designat	Certified Number Passer	Certified Takeoff Weight
29	28	91	AERO COMMANDER	AERO COMMANDER-680	AERO COMMAI	680	FLP	PRESSURIZED GRAND COMM	2A4	AC6L	10	8500
30	30	92	AERO COMMANDER	AERO COMMANDER-680	AERO COMMAI	680	T	TURBO COMMANDER	2A4	AC80	10	8950
31	31	93	AERO COMMANDER	AERO COMMANDER-680	AERO COMMAI	680	UNDESIGNATEI	SUPER COMMANDER	2A4	AC68	6	7000
32	32	94	GULFSTREAM	AERO COMMANDER-680	AERO COMMAI	690	C	JETPROP 840	2A4	AC90	10	10325
34	34	95	GULFSTREAM	AERO COMMANDER-680	AERO COMMAI	690	D	JETPROP 900	2A4	AC90	10	10700
35	35	96	GULFSTREAM	AERO COMMANDER-680	AERO COMMAI	695	A	COMMANDER 1200	2A4	AC95	10	11200
37	37	99	AERO COMMANDER	AERO COMMANDER-680	AERO COMMAI	720	NO SERIES EXIS	ALTI CRUISER	2A4	AC72	5	7500
44	44	122	AEROCAR	AEROCAR-I	AEROCAR	I	NO SERIES EXIS	No Popular Name	4A16	CAR	1	2100
45	45	131	AERONCA	AERONCA-11AC	AERONCA	11AC	NO SERIES EXIS	CHIEF	A761	AR11	1	1250
46	46	132	AERONCA	AERONCA-11AC	AERONCA	11BC	NO SERIES EXIS	CHIEF	A761	AR11	1	1250
47	47	133	AERONCA	AERONCA-11AC	AERONCA	11CC	NO SERIES EXIS	SUPER CHIEF	A796	AR11	1	1350
48	48	134	AERONCA	AERONCA-15AC	AERONCA	15AC	NO SERIES EXIS	SEDAN	A802	AR15	3	2050
49	49	138	AERONCA	AERONCA-50TC	AERONCA	60TF	NO SERIES EXIS	TANDEM	A728	AR6T	1	1150
50	50	140	AERONCA	AERONCA-K	AERONCA	65CA	NO SERIES EXIS	SUPER CHIEF	A675	AR65	1	1250
51	51	141	AERONCA	AERONCA-50TC	AERONCA	65TAC	NO SERIES EXIS	DEFENDER	A728	AR6T	1	1200
52	52	142	AERONCA	AERONCA-50TC	AERONCA	65TAF	NO SERIES EXIS	DEFENDER	A728	AR6T	1	1200
53	53	143	AERONCA	AERONCA-50TC	AERONCA	65TAL	NO SERIES EXIS	DEFENDER	A728	AR6T	1	1200
54	54	144	AERONCA	AERONCA-50TC	AERONCA	65TC	NO SERIES EXIS	TANDEM	A728	AR6T	1	1150
55	55	145	AERONCA	AERONCA-50TC	AERONCA	65TF	NO SERIES EXIS	TANDEM	A728	AR6T	1	1150
56	56	146	AERONCA	AERONCA-50TC	AERONCA	65TL	NO SERIES EXIS	TANDEM	A728	AR6T	1	1150
57	57	147	AERONCA	AERONCA-7AC	AERONCA	7AC	NO SERIES EXIS	CHAMPION	A759	CH7A	1	1220
58	58	147	AERONCA	AERONCA-7AC	AERONCA	7AC	NO SERIES EXIS	CHAMPION	A759	CH7A	1	1220

FAA Taxonomy

	inde	cictt_string	ACFT_MMS_CODE	MFR_SHORT_MAK E	ACFT_MODEL_COI	MASTER_ID	POPULAR_NA ME	TYPE_CERT_ NUMBER	REG_NO- SEATS	REG_AC- WEIGH	REG_SPE ED
29	28	91	GA-680-FLP	GULSTM	GA-680	GA-680		2A4	11	CLASS 1	188
30	30	92	GA-680-T	GULSTM	GA-680	GA-680-TP	TURBO	2A4	11	CLASS 1	188
31	31	93	GA-680-680	GULSTM	GA-680	GA-680	SUPER	2A4	7	CLASS 1	188
32	32	94	GA-690-C	GULSTM	GA-690	GA-690-TP	840 JETPROP	2A4	11	CLASS 1	0
34	34	95	GA-690-D	GULSTM	GA-690	GA-690-TP	900 JETPROP	2A4	11	CLASS 1	0
35	35	96	GA-695-A	GULSTM	GA-695	GA-695	1000 JETPROP	2A4	11	CLASS 1	0
37	37	99	GA-720-720	GULSTM	GA-720	GA-720	ALTI-CRUISER	2A4	6	CLASS 1	187
44	44	122	AROCAR-1-1	AROCAR	AROCAR-1	AROCAR-1		4A16	2	CLASS 1	87
45	45	131	AR-11-AC	ARONCA	AR-11	AR-11	CHIEF	A761	2	CLASS 1	75
46	46	132	AR-11-BC	ARONCA	AR-11	AR-11	CHIEF	A761	2	CLASS 1	75
47	47	133	AR-11-CC	ARONCA	AR-11	AR-11	SUPER CHIEF	A796	2	CLASS 1	75
48	48	134	AR-15-AC	ARONCA	AR-15	AR-15		A802	4	CLASS 1	85
49	49	138	AR-65-60TF	ARONCA	AR-65	AR-65		A728	2	CLASS 1	75
50	50	140	AR-65-CA	ARONCA	AR-65	AR-65		A-675	2	CLASS 1	75
51	51	141	AR-65-TAC	ARONCA	AR-65	AR-65		A728	2	CLASS 1	71
52	52	142	AR-65-TAF	ARONCA	AR-65	AR-65		A728	2	CLASS 1	71
53	53	143	AR-65-TAL	ARONCA	AR-65	AR-65		A728	2	CLASS 1	71
54	54	144	AR-65-TC	ARONCA	AR-65	AR-65		A728	2	CLASS 1	75
55	55	145	AR-65-TF	ARONCA	AR-65	AR-65		A728	2	CLASS 1	75
56	56	146	AR-65-TL	ARONCA	AR-65	AR-65		A728	2	CLASS 1	75
57	57	147	BL-7-AC	BLANCA	BL-7	BL-7	CHAMP	A759	2	CLASS 1	90
58	58	147	AR-7-AC	ARONCA	AR-7	AR-7	CHAMPION	A759	1	CLASS 1	0

The original plan for the matching process was to create a custom Streamlit application using Python to support the manual matching. However, the process uses many columns and having the full set of Excel capabilities to sort and create ad-hoc formulas was thought to be more efficient. The process involves careful scanning of the columns included in Figure 4, as

well as additional columns, to determine the matches. The first two columns of the figure show the mapping columns between the two sources.

All collected datasets and associated preprocessing scripts will be stored in a GitHub repository with version control. This will ensure that any changes to datasets or code are documented, supporting reproducibility and transparency of the study workflow.

Step 4: Creating a Baseline ER Model

To address RQ1, as a second step in the Data Preparation phase a baseline matcher will be constructed using the open source *RecordLinkage* library in Python. The *RecordLinkage* matcher serves as a feature-based baseline model. Its results will form the basis for RQ1 and the hypothesis test H10/H1a, potentially establishing the relative benefit of advanced models like Ditto.

This matcher will use rule-based comparisons for each feature. For example, the model field will be compared using a string comparison metric such as Jaro-Winkler or Levenshtein distance to determine the feature level match. Then an aggregate score across all columns will be used to classify as a match or non-match. The feature level matches are generally expressed as a simple 1 or 0 score for feature level match or no-match and the aggregate is the sum of these individual results. A predefined threshold for the sum is generally used to determine the final classification. The results will serve as a point of comparison for the deep learning approaches explored later in the study. This baseline will be tested only on the gold taxonomy-to-taxonomy dataset.

Step 5: Preparing the Input Feature Sets

This is the first step of the Modeling phase in the CRISP-DM and where this process flow will add additional steps to the overarching process flow diagram for clarity. A set of candidate

input feature configurations will be defined and labeled as individual model runs (e.g., M1, M2, M3). These features represent different combinations of source fields such as make, model, series, popular name, ICAO designator, type certificate, and other relevant aircraft type metadata. Each feature configuration will be carefully documented and aligned with Ditto's input string format. A complete listing of features and model variants is provided in Annex A.

Step 6: Defining Levels of Hierarchy

This is another step in the deep learning Modeling phase of the CRISP-DM. Aircraft types exist at multiple levels of granularity, and these different levels will be explicitly defined and used to evaluate Ditto's performance. These will be labeled as H1 (fine-grained, make-model-series), H2 (make-model), and H3 (aggregated level, such as master make-model or master series). Labels for each level of hierarchy will be derived from taxonomies through deterministic roll-ups.

These hierarchical levels allow the study to evaluate model performance for the varying levels of aggregation. This is important because not all data sources are described at the same level of granularity and the data matching to a taxonomy is not possible if the data is not described at the same level of detail as the taxonomy. Analysis goals can also be a determining factor for desired level of granularity. The study will build different models at different aggregation levels so that the matching level can be selected a priori depending on the data source characteristics and analysis goals.

Step 7: Generating Ditto-Compatible Inputs

Custom Python scripts will be written to generate training and evaluation files for Ditto in the required COL VAL format. For each model (e.g., M1, M2) determined in step 5 and each level of hierarchy (e.g., H1, H2, H3) determined in step 6, a corresponding dataset will be

generated using the labeled pairs from the gold dataset. The models are determined by the set of features used to generate the aggregated text input and the hierarchy is used to determine the binary classification label. These datasets will be stored with appropriate identifiers.

During this same step, where the matching entities are formatted using the appropriate aggregate text string, non-matching entities will also be generated by sampling from all other entities. The non-matching entries will be sampled at a roughly 10:1 ratio where 10 non-matching pairs will be generated for each matching pair.

For each combination of model and hierarchy, the combined matching and not matching data will be divided into three comma-delimited files for a train, test, and validation sets. The splits will be roughly 80%, 10%, and 10% but this can be adjusted if needed.

Step 8: Modeling with Taxonomy Data

Ditto will be trained on each model and hierarchy configuration from steps 6 and 7 based on the appropriate aggregated text string and hierarchy level from the gold taxonomy dataset. To manage this, a Python script will be used to call the Ditto model with the appropriate data inputs and store the trained model outputs. A performance matrix will be created where each row represents a model configuration (e.g., M1, M2) and each column represents a hierarchy level (e.g., H1, H2, H3). Precision, recall, and F1 scores will be reported for each cell in this matrix, forming the primary evaluation for RQ2.

Step 9: Sampling and Labeling of Additional Aviation Datasets

To address RQ3, a separate set of samples will be selected from the four remaining aviation datasets: NTSB, BTS, AIDS, and NWSD. These datasets are more "dirty," lack schema alignment, and do not link directly to the taxonomies. A small number of records from each will

be manually labeled at an aggregate hierarchy level (e.g., H3) using the same annotation approach as the gold dataset.

These samples may be treated as one combined dataset for inference tasks. This would represent the model's ability to classify matches on a generic aviation domain set. Additionally, while the registry aircraft type is included in the gold dataset, registry only columns could be used as a different input source without the need for additional labeling.

Step 10: Transferring In-Domain via Inference

Using the best-performing Ditto model trained on the gold dataset at the same level of hierarchy (e.g., H3), inference will be performed on the newly labeled samples from step 7. This test evaluates Ditto's generalization capacity on in-domain data not part of the training set and supports RQ3. Performance will again be measured using F1 scores and compared with fine-tuned results in step 12.

Step 11: Fine-Tuning on In-Domain Datasets

In this phase, the previously trained Ditto model will be fine-tuned on the small subset of the labeled data from Step 9. This step will use transfer learning with frozen layers and a new classification head. This will allow a comparison between inference-only models and models adapted to a specific in-domain dataset, testing the benefit of task adaptation to the new data source.

Step 12: Evaluating Performance

The evaluation phase will focus on results from both the taxonomy-to-taxonomy models and from the multi-source models. These are described in depth in the data source section below. For the multi-source models, F1 scores from the inference-based results in step 10 and the fine-tuned results in step 11 will be compared to evaluate the effectiveness of domain adaptation.

These comparisons will be used to test RQ3 and assess the extent to which transfer learning is effective for aircraft type data.

Step 13: Deploying the Results

This step will demonstrate how the ER model can be applied across multiple aviation datasets. Using the model to link aircraft type across multiple data sets enables the normalization of aviation safety event data using utilization proxies (e.g., aircraft counts, flight hours). This use case validates the practical utility of the aircraft type ER model and illustrates its potential contribution to aviation safety analysis.

In addition, this step includes the creation and dissemination of open-source artifacts via GitHub, including the curated gold dataset, supporting Python code, and, where appropriate, trained model weights. All artifacts will be documented to ensure reproducibility and facilitate use by other practitioners and researchers.

These steps will be used as a structured sequence of procedures for the study plan. This structure allows for a robust and reproducible evaluation of ER approaches to deep learning entity resolution approaches. Using the CRISP-DM framework and aligning tasks directly with the research questions, the study follows a concrete methodology while at the same time addressing the practical challenges of aircraft type matching in aviation data.

Data Analysis

The data analysis strategy for this study will be to evaluate and compare the performance of different ER approaches across multiple experimental configurations. This includes a baseline feature-based matching method and deep learning models using Ditto applied to a curated aircraft type dataset. The Ditto model is evaluated with multiple input feature sets, at different levels of granularity, and across various aviation data sources. The analysis supports the three

research questions and tests the null hypothesis H_0 against the alternative H_{1a} using F1 scores as the primary dependent variable.

Baseline matcher (RQ1)

To address RQ1, the baseline $RecordLinkage$ matcher will be evaluated on the gold taxonomy-to-taxonomy dataset. This model uses rule-based feature comparisons and a threshold-based decision rule. The results will serve as a point of comparison for the deep learning approaches in RQ2. No statistical inference will be applied to this baseline as it is intended purely as reference.

Deep learning models (RQ2)

To address RQ2, each configuration of the Ditto model for input feature sets (e.g., M1, M2…) and levels of hierarchy (e.g., H1, H2, H3) will be trained and evaluated independently on the curated gold dataset. The labeled data will be divided into train, validation, and test sets using a fixed split of 80% training, 10% validation, and 10% testing.

The precision, recall, and F1 score will be calculated using the held-out test set for each model configuration. These results will be analyzed in tabular form to compare different feature sets at the different granularity levels and to evaluate overall model effectiveness on the aircraft type matching task. The best performing feature set for each level of hierarchy will be displayed in bold face to support comparisons across the different hierarchies. Based on these results, an optimal feature set will be selected.

To statistically assess differences across the various deep learning configurations, this study adopts the non-parametric Friedman test recommended by Demšar (2006) as an alternative to repeated-measure ANOVA. If the Friedman test with an α-value of 0.05 indicates statistically

significant differences, a post-hoc Nemenyi test will be used to identify which model pairs differ significantly. These tests will be implemented in R or Python.

Multi-Source Data Matching (RQ3)

To address RQ3, the best-performing Ditto model from RQ2 (for the appropriate hierarchy level) will be applied to additional "dirty" aviation datasets that do not align schema-wise with the taxonomies (e.g., AIDS, NTSB, NWSD, BTS). Inference-only performance will be compared against a fine-tuned version of the model trained on a small manually labeled sample from each dataset. These comparisons will also include Registry-only features and an optional combined dataset representing generic aviation data.

Similar to the results of RQ2, F1 scores will be calculated for each dataset and the results will be displayed in tabular form. No formal hypothesis testing will be conducted for RQ3. The comparisons will be descriptive and used to inform domain transfer strategies and gauge the suitability of a generic aircraft type matcher that works across multiple data sources.

Hypothesis Testing

The hypothesis test for this study measures the baseline data matcher against the top performing Ditto configuration. The null hypothesis is that there is no difference in performance, measured by the F1 score between the two models.

A two-tailed independent samples t-test (or Wilcoxon signed-rank test if assumptions of normality are not met) will be used to evaluate the difference in F1 scores across multiple experimental runs. This single test will support the evaluation of the hypothesis. R or Python will be used for statistical testing.

Assumptions

Assumptions are foundational beliefs held to be true that guide the research process (Creswell & Creswell, 2018). This study assumes that aircraft type is a meaningful unit of aggregation for aviation safety data and that linking aircraft type across datasets enables valid risk-based analyses. It further assumes that aviation safety analysts face challenges in aligning aircraft type across heterogeneous data sources and that improving this alignment can enhance the quality of aviation safety analyses.

Additionally, the study assumes that existing ICAO and FAA taxonomies are accurate, representative, and define hierarchies that are useful to practitioners. Another key assumption is that it is possible to accurately match taxonomy entries between the ICAO CICTT taxonomy and the FAA taxonomy and that the labeling process used to match these sources produced minimal errors that do not impact the findings of the study. Lastly, it is assumed that growing interest in aviation knowledge graphs reflects a need for the alignment of aviation data sources and that this study can meaningfully contribute to that goal.

Limitations

Limitations are potential constraints or weaknesses in the study which are out of the researcher's control but may limit generalizability (Creswell & Creswell, 2018). In selecting a deep learning model using a BERT-based embedding, this study is limited by the ability of textual embeddings to capture semantic meaning from aircraft type data. By using a single aggregation string to represent all data features, the ability of a deep learning model to learn individual features from these embeddings and relationships between these features is a limitation. Based on its previous success across various domains it is believed that the Ditto

model can perform well on a variety of data. This study will explore this existing paradigm rather than selecting a strategy to train an aviation specific language model.

Another limitation is that some data may be incorrect, ambiguous, or recorded without sufficient specificity. It may be impossible to map some dirty or incomplete fields to a taxonomy. Creating multiple models for different levels of granularity may partially mitigate this limitation but not all entries will align to these predefined levels, and some datasets may contain mixed levels of granularity. Future research may be needed to address gaps in mapping to mixed levels of granularity.

Delimitations

Delimitations are intentional decisions and boundaries to narrow the scope of the study (Creswell & Creswell, 2018). This study assumes that aircraft type is a useful unit of analysis. Entity resolution algorithms could be applied to individual aircraft using registration or serial numbers. This was not selected because often data is deidentified in public sources.

The study also focused on cases where a clear mapping exists between taxonomies and used a purposive sample that focused on larger and in service aircraft. Military aircraft and light sport aircraft were not excluded but were not intentionally selected for the gold data set. For this reason, matching may not generalize to smaller aircraft, aircraft no longer in service, and new aircraft types like electric vertical takeoff and landing (eVTOL) and unmanned aircraft and drones.

In the ER framing, blocking is a step to reduce the number of candidate pairs to avoid exhaustive comparisons and compare only the pairs most likely to match (Javdani et al., 2019). From a modeling point of view the study intentionally focused on the data matching problem and ignored blocking and clustering techniques. The number of different aircraft types is thought to

be manageable enough to ignore the blocking phase. This decision limits applicability to use cases involving extremely large or highly redundant datasets, where clustering might be required to resolve many-to-many matches.

The primary theme for the conceptual framework, problem statement, and purpose statement focus on the matching aspects of aircraft type. While blocking and deduplication are important aspects of the general ER pipeline, the salient problem addressed in this study is the data matching problem. The research questions have focused on the suitability of ER for data matching and the study of deep learning approaches, as well as multi-source ER. These questions and the constructive nature of the study also lead this study towards a focus on data matching rather than on large-scale implementation considerations.

Ethical Considerations (Secondary data)

This study received approval from the University's Institutional Review Board (IRB) prior to any data collection or analysis activities. The research exclusively uses publicly available secondary data sources related to aviation safety events, aircraft utilization, and taxonomy data, and therefore presents no more than minimal risk to individuals.

Because no personally identifiable information (PII) is included in these data sources, and because no data is collected directly from human subjects, there are no direct privacy concerns. One reason for selecting aircraft type as a focus of study was to ensure that the study did not focus on individual aircraft that could be associated with individuals. The datasets are collected from publicly available sources and are available for research. This study does not attempt to re-identify individual aircraft or link data to specific individuals or operators.

The files will all be stored on a personal computer with password protection. The code and analysis will be maintained in a private repository on GitHub until artifacts are reviewed for

public dissemination. No sensitive information will be included in these publicly disseminated files.

The researcher has professional experience in both aviation safety and data science and brings domain expertise to the study design and findings. While this background is valuable for understanding potential issues with aircraft type data, there is a potential for unintentional bias based on previous experiences. As a mitigation, the modeling used for this analysis will be reproducible and quantifiable with well-established metrics such as F1 scores. The manual data labeling will be based on existing data features and consistent with existing type certificate documentation.

Summary

This study follows a constructive research methodology to match heterogeneous aviation data sources by aircraft type. This chapter outlines the research methodology and procedures to conduct the study. The study follows a CRISP-DM process framework to iteratively design a solution grounded on theoretical foundations in ER. Recent progressions in deep learning have been applied successfully in ER and Ditto has emerged as state-of-the-art for deep learning-based data matching.

This chapter shows how Ditto, Python, and other tools will be used collectively to apply ER techniques to the aircraft type data matching problem. The study process involves creating a labeled gold dataset by matching two aircraft type taxonomies and developing a baseline data matcher using a traditional ER pipeline. Ditto is used as the central instrument to match records across various input feature set configurations and hierarchy levels using the gold dataset. The same model is then applied to additional aviation sources to evaluate the generalizability of the approach.

The chapter also addressed ethical considerations, assumptions, limitations, and delimitations relevant to the study, and a detailed data analysis plan describing how the results will be evaluated and interpreted. The next chapter will present the findings of the study, including performance metrics across all configurations, comparisons between methods, and evidence in support of or against the proposed hypotheses.

Chapter 4: Findings

The problem addressed in this study is the challenge of merging aviation data from diverse sources that lack common keys, hindering safety analysts' ability to assess risk using multiple information sources. The purpose of this constructive research study is to develop an entity resolution model that can connect disparate aviation databases by matching across data sources using aircraft make-model-series specifications.

The research questions addressed in this study are:

RQ1. To what extent, if any, can a traditional entity resolution approach based on feature matching be used to match aircraft type between data systems?

RQ2. To what extent, if any, can deep learning techniques be used to match aircraft type between data systems?

RQ3. To what extent, if any, is in-domain transfer possible using deep learning to match aircraft type between systems?

Although the study is framed as constructive research, the following hypothesis comparison was also assessed:

$H1_0$. There is no significant difference in performance between the traditional baseline and the deep learning approach.

$H1_a$. There is a significant difference in performance between the traditional baseline and the deep learning approach.

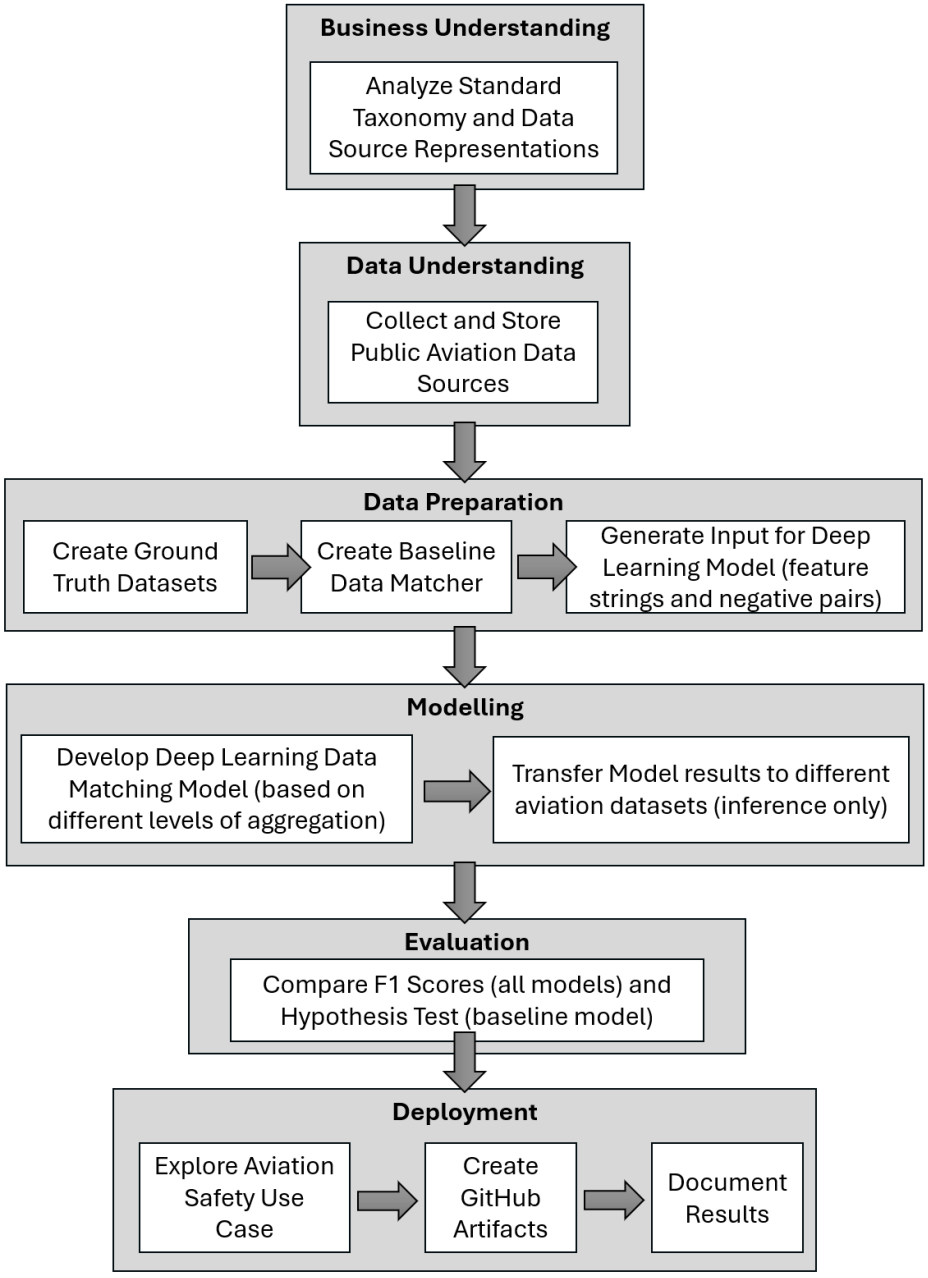
This chapter reports the steps executed to build the curated datasets, develop and evaluate baseline and deep learning matching approaches, and assess transfer to additional aviation datasets. Two GitHub repositories support this study: /karnabryan/aircraft_er and /karnabryan/ditto_aircraft_er (Bryan, 2026a; Bryan 2026b).

Data Preprocessing and Modeling Process Diagram

Figure 5 provides the as-executed workflow for this study within the CRISP-DM framework. The first phase analyzed aircraft type fields across the selected public data sources and identified opportunities to align sources when possible. This phase also produced a labeled "gold" dataset that manually matched the FAA and CICTT taxonomies, which do not share any common keys. This gold dataset was used to create a baseline matcher using a standard entity resolution process and to create train/validation/test datasets for the deep learning model Ditto. After training Ditto on the taxonomy-to-taxonomy task, additional models were trained and evaluated on other aviation data sources, and a multi-source model was created by combining the training and test datasets from the individual source training and test datasets to assess generalization across the input formats. The final step compared metrics across model runs to evaluate individual model generalization performance, documented findings, and published artifacts on GitHub.

Figure 5

Process Model based on CRISP-DM Framework



Although the study followed the CRISP-DM structure proposed in Chapter 3, several implementation decisions changed what was ultimately evaluated in Chapter 4. The items below are retained in the main chapter because they directly affect interpretation of the reported results, comparisons across tables, and the hypothesis test. A complete list of the deviations are documented in Appendix A.

The first refinement was to focus the hierarchy of the study to just make-model-series (MMS) and make-model (MM). While Chapter 3 proposed additional levels of hierarchy, the master-model and master-series levels in the FAA taxonomy were inconsistently and incompletely defined. As a result, the focus was concentrated on MMS and MM, the two levels that could be defined consistently. From a modeling point of view, the as-executed workflow added union-trained models at both the MMS and MM levels by combining source-specific training datasets using split-preserving construction. This addition enabled a direct comparison between source-specific model checkpoints and a single union-trained checkpoint evaluated on the same source test partitions.

Among the in-domain transfer datasets, BTS was the only source fully labeled within the scope of the study. This enables quantitative top-1 evaluation for BTS, including comparison of transfer-style inference and a BTS-trained model. In contrast, the other in-domain datasets (e.g., AIDS, NTSB, NWSD) were not fully labeled and are therefore evaluated using manual audit summaries, which are descriptive rather than full-dataset performance estimates. For the in-domain datasets, exhaustive candidate generation was not computationally feasible. To make evaluation feasible, candidate pairs were constrained using a manually standardized make field so that each source record was compared only to CICTT candidates sharing the same make. The

in-domain transfer findings should be interpreted as conditional on this executed candidate-generation strategy.

The formal hypothesis test was changed to McNemar's test applied to paired top-1 predictions from the baseline and deep learning models on the same held-out records. This change better matches the data structure because the models are evaluated on the same cases (paired outcomes), and the primary inferential question is whether the models differ in error behavior on those shared observations. Precision, recall, and F1 remain the primary descriptive metrics across experiments, while McNemar's test provides the formal statistical comparison for the baseline-versus-Ditto hypothesis.

Data Preprocessing

This section describes the preprocessing steps executed to prepare the study datasets for baseline and deep-learning entity resolution. It summarizes the business and data understanding analyses that shaped feature and hierarchy decisions and details the construction of the FAA-to-CICTT gold dataset, the $RecordLinkage$ baseline matcher inputs, and the Ditto training and evaluation files. These steps produce a consistent set of MMS- and MM-level datasets, including source-specific and union configurations, as well as evaluation-only in-domain datasets used to assess transfer learning on the union configuration. Details of the data preprocessing are described in more detail in Appendix A.

Business Understanding (As Executed)

Aviation safety analyses often require combining records from multiple public datasets (e.g., operational activity, aircraft registry information, incident/accident narratives, and taxonomy references). In practice, these sources rarely share a universal aircraft type identifier, and aircraft type is represented using different conventions depending on the system's purpose.

The central business problem in this study is therefore not simply "matching strings," but enabling cross-source analysis by identifying which sources can be linked deterministically and which require entity resolution to reconcile aircraft type specifications. Table 4 summarizes how aircraft type is represented across the study's data sources and whether each source provides standardized linkage keys. From Table 4, a key insight is that the Registry code links the FAA source to the OPER and REG sources through a common key and the ICAO type designator directly links the CICTT and ICAO sources. Thus, by creating the manual mapping between the FAA and CICTT sources then using these join keys, direct data manipulation can be used to create ground truth files for all five of these sources. The gold dataset labeling only needed to focus on linking the structured CICTT and FAA tables, referred to as the taxonomy-to-taxonomy mapping in the remainder of this study.

Table 4

Aircraft Type Representations and Deterministic Linkage Keys by Data Source

Data source	Structured MMS fields (make/ model/series)	ICAO type designator	Registry code	Linkage implication
CICTT	Yes	✓		Deterministic to ICAO; manually map to FAA for gold dataset
ICAO	Partial	✓		Deterministic to CICTT
FAA	Yes		✓	Deterministic to REG/OPER using registry code; manually map to CICTT for gold dataset
AIDS	Partial			No direct linkage keys; single "model" field includes series
NTSB	Yes			No direct linkage keys
NWSD	Free text			No direct linkage keys; "dirty" aircraft type specification
OPER	Partial		✓	Deterministic to FAA/REG via registry code

Data source	Structured MMS fields (make/ model/series)	ICAO type designator	Registry code	Linkage implication
REG	Partial		✓	Deterministic to FAA/OPER using registry code; single "model" field includes series
BTS	Free text			No direct linkages; "dirty" aircraft type specification

Note. ✓ indicates the field is present as a standardized mapping key. "Structured MMS fields" indicates the presence of discrete make, model, and series attributes. "Partial" indicates that MMS information is present but collapsed (e.g., series embedded in model). "Free text" indicates aircraft type is provided as a single unconstrained text string rather than standardized fields.

Data Understanding (As Executed)

Following the business understanding phase, the study conducted a deeper dive into actual content of the columns represented in each of the study data sets to gain an understanding of the level of standardization in these fields. For context into the concrete challenges for data matching observed in aircraft type data, shows variations in aircraft type strings for a single make model across nine aviation data sources. These highlight variations between Bombardier and Airbus Canada for the make, along with abbreviations, misspellings, and other variations seen in the data. The model is most often coded as BD500 but is sometimes also encoded as A220 or CS100 or CS300. The two series values 1A10 and 1A11 are contained in some of the datasets and not others representing a difference in the reported level of granularity. Information from the model and series fields are sometimes repeated across the columns. Finally, the four-letter ICAO designators BCS1 and BCS3 show little resemblance to the other fields, where some types are easier to guess (e.g. B38M represents a Boeing 737 Max 8). Those sources with a single free-text entry field for aircraft type are particularly inconsistent.

Table 5 shows variations in aircraft type fields for a single make model across nine aviation data sources. These highlight variations between Bombardier and Airbus Canada for the

make field, along with abbreviations, misspellings, and other variations seen in the data. The model field is most often coded as BD500 but is sometimes also encoded as A220 or CS100 or CS300. The two series values 1A10 and 1A11 are contained in some of the datasets and not others representing a difference in the reported level of granularity. Information from the model and series fields are sometimes repeated across the columns. Finally, the four-letter ICAO designators BCS1 and BCS3 show little resemblance to the other fields, where some designator types are easier to guess (e.g. B38M represents a Boeing 737 Max 8). Those sources with a single free-text entry field for aircraft type show the greatest inconsistency. Matching records across sources requires context beyond variations in strings, including an understanding that an BD500, A200, and CS100 can all mean the same thing.

Table 5

Aligned aircraft type strings for the BD500 make-model across nine aviation data sources

Source	Make	Model/ Description	Series	Popular name	ICAO type designator
CICTT	BOMBARDIER, AIRBUS CANADA	BD500	1A10	CS100, A220-100, ACJ TWO TWENTY	BCS1
	BOMBARDIER, AIRBUS CANADA	BD500	1A11	CS300, A220-300	BCS3
ICAO	BOMBARDIER	BD-500 CSeries CS100			BCS1
	BOMBARDIER	BD-500 CSeries CS300			BCS3
FAA	BOMBDR	BD-500	BD-500-1A10		
	BOMBDR	BD-500	BD-500-1A11		
AIDS	AIRBUS CANADA LP, BOMBARDIER, BOMBARIER		BD-500-1A11		
NTSB	BOMBARDIER	BD500	1A11		
NWSD		A-220			
OPER		BD-500	BD-500-1A10		

Source	Make	Model/ Description	Series	Popular name	ICAO type designator
REG	AIRBUS CANADA LP	BD-500	BD-500-1A11		
		BD-500-1A11			
BTS		Bombardier C Series Cs100			

Note. Values are shown as recorded in each source to illustrate cross-source variation in how the same aircraft make and model is represented. Attributes not shown are not missing values, but fields not contained in the dataset. When only one aircraft type description field is available, this is shown in the model column in this table.

Data Preparation: Create Ground Truth Datasets (As Executed)

The primary objective of the ground-truth construction step was to create a labeled dataset aligning an FAA aircraft type reference representation to the CICTT taxonomy in the absence of shared keys. This "gold" dataset provides supervised labels for model development and evaluation and serves as the bridge between the two deterministic linkage groups identified in the business understanding phase (those linked by the ICAO designator and those linked by the registry code). More detailed implementation steps are provided in the Data Preprocessing section of Appendix A.

The primary labeled dataset in this study is a manually curated FAA-to-CICTT aircraft type mapping used as the gold standard for supervised training and evaluation. This gold dataset was created to bridge the two taxonomy systems that do not share a common identifier and to provide the reference labels needed for both the traditional baseline matcher and the Ditto models. To construct the gold dataset, the FAA aircraft type reference was first consolidated with FAA-adjacent sources (OPER and REG) using the registry code and FAA aircraft type keys described in Table 4. This produced an FAA-aligned table that provided additional contextual

fields for manual labeling and reduced the labeling scope to aircraft types observed in the registry and operations-linked population. During execution, the originally planned FAA reference file version was no longer publicly available, so an equivalent FAA reference file distributed with the AIDS dataset was used instead. The substituted file retained the same schema but contained slightly fewer rows than the originally planned source.

After consolidation and filtering, the FAA-aligned labeling table contained 3,759 aircraft type rows. These rows were manually mapped to CICTT identifiers using FAA make, model, and series fields, with type certificate and popular name fields used for disambiguation when needed. Of the 3,759 rows, 3,678 were assigned to a non-null CICTT identifier and retained as labeled positive FAA to CICTT matches. The remaining 81 rows were excluded from the MMS gold dataset because they were ambiguous or could not be assigned a consistent make-mode-series mapping.

This gold dataset supports the supervised training and evaluation of the baseline taxonomy-to-taxonomy models, construction of MMS and MM labeled datasets, and construction of the labeled dataset for linked sources through direct joins on the registry code and ICAO type designator per the relationships described in Table 4.

Data Preparation: Create Baseline Data Matcher (As Executed)

A deterministic feature-based baseline matcher was implemented using the *RecordLinkage* library to provide a transparent comparison point for the deep learning approach. The baseline matcher operates on the FAA to CICTT gold dataset and uses equivalent fields from both taxonomies to compute similarity scores.

From the gold dataset, reference table was prepared for schema aligned CICTT and FAA columns for make, model, series, type certificate, and popular name. Because multiple FAA rows

can map to the same CICTT identifier, the CICTT side was de-duplicated at the entity level for the baseline matching task. This produced 3,456 unique CICTT entities and 3,678 FAA rows for the taxonomy-to-taxonomy baseline evaluation.

All matching fields were standardized prior to scoring using a consistent text-cleaning procedure (e.g., lowercasing, normalization of separators, and removal of non-informative formatting variation). Candidate generation for the baseline matcher used exhaustive indexing without blocking, comparing every CICTT record to every FAA record. This produced the full candidate universe used for the pairwise baseline evaluation presented in the Results section.

The baseline matcher first computed binary field level similarity indicators based on both exact matches and text similarity scores. The field level indicators were then summed to create a total score used for threshold-based pairwise classification. In addition to pairwise evaluation, a top-1 linkage strategy was prepared by selecting the highest-scoring candidate per source record with deterministic tie-breaking. The exhaustive pairwise and top-1 linkage results are both reported the Results section because they answer different aspects of RQ1.

Data Preparation: Create Ditto Input Files (As Executed)

To train Ditto, the labeled FAA to CICTT data were converted into Ditto's required pairwise text format. Each candidate pair consists of a serialized left record and a serialized right record, together with a binary match/non-match label. Records are serialized into Ditto's generic COL/VAL representation allowing heterogeneous schemas to be compared without requiring the columns to align. For the baseline taxonomy-to-taxonomy MMS task, the left record was the CICTT entity and the right record was the FAA row. Model inputs used the same five aircraft type attributes (make, model, series, popular name, and type certificate).

Positive examples were created directly from the 3,678 labeled FAA-to-CICTT matches. Negative examples were generated by pairing each positive with non-matching candidates. The baseline Ditto configuration used a 1:10 positive-to-negative ratio with randomly sampled negatives. Additional "learn-harder" (LH) configurations were also prepared for the taxonomy-to-taxonomy MMS task by increasing the proportion of closely related negatives, such as those sharing the same make model with a different series or records with a matching value for series. These variations were included to test the effect of negative sampling on false positives. These LH variants were created only for the baseline taxonomy-to-taxonomy MMS experiments.

For supervised Ditto experiments, datasets were split into train/validation/test partitions using an 80/10/10 split. In addition to the standard train, validation, and test files, all-pairs and ID-preserving versions of the files were produced to support downstream error analysis, top-1 linkage evaluation, and hypothesis-testing preparation. These files were used to evaluate Ditto under both held-out test-split conditions and alternate candidate-generation scenarios.

Performance of the $RecordLinkage$ baseline was evaluated on an exhaustive set of candidate pairs, but using Ditto evaluation on the same set of exhaustive candidate pairs provided to be computationally challenging. Further, evaluation of the Ditto model beyond the held-out test set changes the meaning of the metrics. To better understand the true performance of the negative candidate generation strategies, evaluation-only stress-test files were created for the baseline taxonomy-to-taxonomy task. These three sets (referred to as Canadair, Random-10, Random-100) contain a small set of known positive matches paired with exhaustive non-matching candidates and were used to assess the true false-positive behavior for the highly imbalanced candidate pools. A full exhaustive CICTT to FAA candidate pair dataset was

prepared for inference-only evaluation to verify the results on the smaller exhaustive datasets using a revised script for candidate pair generation.

For the formal hypothesis test, a separate exhaustive candidate evaluation file was constructed for the held-out Ditto test rows only so that paired top-1 predictions from the baseline and Ditto models could be compared on the same strictly held-out cases using McNemar's test. Results for the exhaustive candidate generation with the held-out test set are presented as part of the hypothesis testing only.

Beyond the baseline taxonomy-to-taxonomy MMS task, additional Ditto datasets were prepared to evaluate schema variation, hierarchy variation, and multi-source training. At the MMS level, labeled source-pair datasets were prepared for CICTT to Registry and FAA to Registry in addition to the baseline CICTT to FAA dataset. A union MMS dataset was then created by combining source-specific train, validation, and test files within split (merging train with train, validation with validation, test with test) before shuffling. This split-preserving construction was used to avoid train/test leakage and to support direct source-specific versus union-model comparisons.

At the MM level, the MMS gold mappings were collapsed to make-model entities to create labeled MM datasets for multiple source schemas. These source schemas were CICTT, FAA, Registry, and DOC8643 (including variants). A union MM dataset was constructed using the same split-preserving approach and used to evaluate whether a single model could support multi-source inference while preserving source-level performance.

For in-domain transfer (RQ3), evaluation files were prepared for AIDS, NTSB, NWSD, and BTS using MM-level candidate generation because series information is not consistently available across these sources. BTS was the only in-domain source fully labeled within scope

and therefore supports quantitative top-1 evaluation. AIDS, NTSB, and NWSD were prepared as evaluation-only inference files and support descriptive manual-audit summaries.

Exhaustive candidate generation was not computationally feasible for the larger in-domain sources. To make evaluation feasible, candidate pairs were constrained using a manually standardized make field, so each source record was compared only to CICTT candidates sharing the same make. This make-constrained candidate-generation strategy defines the executed evaluation universe for the reported in-domain transfer results and should be considered when interpreting RQ3 findings.

The data preparation process produced a set of labeled and evaluation-only datasets supporting baseline $RecordLinkage$ pairwise and top-1 evaluation, Ditto MMS and MM supervised training and evaluation, source-specific and union-model comparisons, and in-domain transfer assessment under make-constrained candidate generation. These prepared datasets define the evaluation scenarios reported in the results section and provide the basis for the comparative analysis across RQ1, RQ2, and RQ3.

Modeling

This section describes the modeling runs executed using the Ditto deep learning entity resolution framework in support of RQ2 and RQ3. Ditto formulates entity resolution as a supervised pairwise classification task. Each record is serialized into a text representation using a COL/VAL convention, and training data are provided as paired left and right serialized records with a binary match or non-match label. After training, the model is used for inference to produce a predictions file containing a match decision and an associated confidence score for each candidate pair. Ditto also supports training a model on one dataset and applying that model

to candidate pairs from a different dataset for inference, which enables transfer evaluation across aviation sources with heterogeneous aircraft type representations.

All model runs used Ditto's standard training pipeline with a transformer language model backbone. Runs were executed using a consistent configuration across datasets to support comparability across experiments. All runs used DistilBERT with batch size 64, maximum sequence length 64, and learning rate 3e-5. Mixed precision fp16 was enabled, and models were fine-tuned for 40 epochs on a GPU. Preliminary sensitivity checks were conducted on a small subset of runs to confirm that performance was stable under minor configuration changes. Experimental emphasis was instead placed on dataset construction decisions specific to the aircraft type matching task, including negative sampling design, schema variation between sources, and the union training strategy.

Table 6 summarizes the Ditto modeling tasks executed for RQ2 and RQ3, including the aircraft type level, training configuration, and evaluation framing used to report results. The text below explains how Table 6 is linked to the two modeling activities in CRISP-DM process model (Figure 5). Modeling was conducted at two levels of aggregation. The Make-Model-Series (MMS) level required agreement at the make, model, and series level as defined by the labeled datasets. The Make-Model (MM) level required agreement at make and model only, supporting sources where series is unavailable, inconsistently encoded, or embedded within other fields. Table 6 provides an overview of the different training configurations and evaluation framings for groups of model runs organized by task. Implementation details about specifics of the tasks, training configuration, and evaluation framing can be found in Appendix A. A high-level perspective of how the model run groups organized by the CRISP-DM modeling activities follows below.

Table 6

Ditto model run tasks and evaluation framings by research question (RQ2-RQ3)

RQ	Task	Level	Training configuration	Evaluation framing
RQ2				
	MMS taxonomy-to-taxonomy baseline (CICTT to FAA)	MMS	Baseline 1:10 + LH variants (learn-harder negative sampling)	Held-out test splits; eval-only stress tests; full exhaustive candidate inference; top-1 linkage from exhaustive inference
	MMS schema variation (CICTT to Registry, FAA to Registry)	MMS	Source-specific train/valid/test (no LH)	Source-specific held-out test split
	MMS union model (CICTT to FAA + CICTT to Registry)	MMS	Split-preserving union training (single checkpoint; no LH)	Union held-out test split and per-source held-out test evaluation
	MM source-specific models (CICTT, FAA, Registry, DOC8643 variants)	MM	Source-specific train/valid/test (no LH)	Source-specific held-out test split
	Union MM (aggregation of all MM source-specific models)	MM	Split-preserving union training (single checkpoint; no LH)	Union held-out test split and per-source held-out test evaluation
RQ3				
	BTS transfer-style inference (CICTT to BTS)	MM	No BTS training (inference using union MM checkpoint)	Top-1 evaluation under two candidate-generation settings: exhaustive and make-constrained (make blocking)
	BTS-trained in-domain model (CICTT to BTS)	MM	BTS-specific train/valid/test (no LH)	Source-specific held-out test split
	Transfer to other in-domain sources (CICTT to AIDS / NTSB / NWSD)	MM	No training (inference using union MM checkpoint)	No full labels; manual audit of 100 top-1 links per source (make-constrained candidate generation)

Note. "Held-out test split" refers to evaluation on labeled train/validation/test partitions using the executed negative sampling design. "Eval-only stress tests" and "full exhaustive candidate inference" refer to highly imbalanced candidate pools used to stress false-positive behavior under exhaustive-style linkage. "Top-1 linkage" selects at most one proposed alignment per left-hand record based on the highest-confidence predicted match. "Union" datasets

were constructed using split-preserving union construction (train with train, validation with validation, test with test) and evaluated on both the union test partition and the original source test partitions.

Modeling: Deep Learning Model Results (As Executed)

The first half of Table 6 summarizes the Ditto deep learning modeling tasks executed in support of RQ2. The first set of model runs established a baseline using the CICTT to FAA taxonomy-to-taxonomy model. Several experiments were added to these runs to evaluate the role of negative candidate generation strategies on model performance. The negative sampling strategy varied across a set of configurations, as described in the data preparation section. Multiple evaluation framings were also used to stress-test these negative sampling strategies.

Following the baseline taxonomy-to-taxonomy runs are the MMS schema variation model runs for CICTT to Registry and FAA to Registry. These tasks evaluate whether Ditto can learn robust matching behavior when the right-hand record schema differs from the full taxonomy feature set and provides fewer structured attributes. In these configurations, the right-hand record serialization reflects the Registry schema and contains only manufacturer and model fields, with series information embedded in the model string. These models were trained and evaluated using a standard held-out train/validation/test split without learn-harder (LH) negative sampling. The purpose of these runs is to isolate the effect of schema differences on MMS matching performance while keeping the training configuration consistent across the source-pair tasks.

The last MMS level task was to explore a union model task that pools the CICTT to FAA and CICTT to Registry dataset using split-preserving union construction, merging training files with training files, validation files with validation files, and test files with test files prior to shuffling. This design supports multi-source training while avoiding train/test leakage across the underlying datasets. The union MMS checkpoint was evaluated on the union test partition and

also used for inference on the original source test partitions to assess whether exposure to multiple right-hand schemas preserved performance relative to the source-specific MMS checkpoints.

The final two RQ2 rows in Table 6 shift modeling from MMS to the Make-model (MM) level. At the MM level, a match is defined by agreement at make and model only, supporting sources where series information is unavailable, inconsistently encoded, or embedded within other fields. MM experiments include a set of source-specific models trained on MM-labeled datasets across multiple right-hand representations, including structured taxonomies and DOC8643 variants designed to test different schema-information conditions (e.g., code-only versus description-based representations). These runs use the same held-out train/validation/test approach (without LH) to support direct comparison across MM datasets.

As with MMS union model, the MM union model was trained using split-preserving union construction to produce a single checkpoint intended to support inference across heterogeneous right-hand formats. Union evaluation includes performance on the union test partition as well as evaluation of each original single-source MM test partition on the union checkpoint. This design provides a direct check of whether union training supports a multi-source "all-purpose" checkpoint while preserving source-level performance on the constituent MM tasks.

Modeling: In-Domain Transfer Learning (As Executed) (RQ3)

The second half of Table 6 summarizes Ditto tasks executed for RQ3, which evaluates in-domain transfer to the remaining heterogeneous aviation sources from Table 4. These sources include AIDS, NTSB, NWSD, and BTS, which represent partially structured fields and unconstrained free-text aircraft type descriptions with examples provided in Table 5. Because

full ground truth labeling was not feasible for most in-domain datasets within scope, transfer evaluation relies primarily on inference and structured review, with BTS included as the labeled in-domain case supporting quantitative evaluation.

The first task was BTS transfer-style inference (CICTT to BTS) using the MM union checkpoint. Both an exhaustive candidate pool and a pool constrained to those candidate records with the same make were evaluated for BTS. Model outputs were summarized using a top-1 linkage strategy, selecting the highest-confidence candidate per BTS record among pairs classified as matches, which provides a practical way to produce one proposed alignment per left-hand record for downstream inspection and comparison to ground truth.

This was compared to a BTS-trained in-domain model, fine-tuned using labeled BTS train/validation partitions and evaluated on the BTS held-out test split. This BTS-specific model provides a direct comparison point for the transfer setting, enabling evaluation of whether performance improves when the model is trained on the target source representation rather than applied under transfer.

The final task demonstrated transfer to other in-domain sources (CICTT to AIDS/NTSB/NWSD) using inference-only evaluation files and the union MM checkpoint. For these sources, exhaustive candidate generation was not computationally feasible and full ground truth labeling was not available. As a result, evaluation used make-constrained candidate generation and top-1 linkage selection, with results summarized through a manual audit of 100 sampled top-1 links per source. These findings are reported as descriptive quality checks rather than as full dataset performance metrics and should be interpreted as conditional on the executed candidate-generation strategy.

The empirical contribution of this chapter is a reproducible comparison of a transparent baseline and a transformer-based ER model across schema-aligned, schema-varied, union-trained, hierarchical (MMS/MM), and in-domain transfer evaluations for aviation aircraft-type normalization. The next section reports results for the traditional $RecordLinkage$ baseline (RQ1) the deep learning modeling tasks summarized in Table 6 (RQ2 and RQ3). Together, these results provide the empirical basis for the evaluation of findings across RQ1-RQ3 and the baseline-versus-Ditto hypothesis test.

Results

This section reports model performance results for the baseline traditional matcher and Ditto-based deep learning models. Results are organized to support the three research questions. For RQ1, baseline $RecordLinkage$ results are reported for the FAA-to-CICTT taxonomy-to-taxonomy task using a threshold sweep and a top-1 approach. For RQ2, Ditto results are reported for the baseline FAA-to-CICTT task (RQ2a), MMS schema variation and MMS union (RQ2b), and MM datasets and MM union (RQ2c). For RQ3, transfer results are summarized for in-domain evaluation-only datasets and the fully labeled BTS case.

Evaluation (As Executed)

Unless otherwise noted, performance is reported using pairwise precision, recall, and F1 derived from confusion-matrix counts. For labeled datasets, metrics reflect evaluation on the held-out test partitions. For evaluation-only stress tests and transfer assessments, metrics reflect evaluation on the constructed candidate-pair files used for inference.

Results for RQ1: Traditional Entity Resolution Performance ($RecordLinkage$ Baseline). Table 7 reports performance of the baseline $RecordLinkage$ model across all observed threshold values. A threshold of $t = 0$ means that all candidate pairs are classified as

matches and none are classified as non-matches. Lower thresholds yield very high recall but extremely low precision because large portions of the candidate universe are labeled as matches. For smaller thresholds, recall is high because most true matches are identified as matches, whereas for larger thresholds, recall is low because very few true matches are identified as true positives. Precision is low for lower thresholds due to large numbers of non-matches classified as matches, and it increases for higher thresholds as false positives decrease. Because of the high class imbalance, false positives dominate the F1 calculation, resulting in relatively poor F1 performance across all threshold values. The best overall balance under this pairwise evaluation is achieved at $t = 6$, with precision, 0.456, recall 0.381, and F1 0.415.

Table 7

Baseline `RecordLinkage` Matcher Performance (Exhaustive Pairwise Evaluation)

Threshold (t)	TP	FP	FN	Precision	Recall	F1
0	3678	12707490	0	0.000	1.000	0.001
1	3527	468082	151	0.007	0.959	0.015
2	3249	199377	429	0.016	0.883	0.031
3	2722	97443	956	0.027	0.740	0.052
4	2313	91010	1365	0.025	0.629	0.048
5	1700	2975	1978	0.364	0.462	0.407
6	**1403**	**1677**	**2275**	**0.456**	**0.381**	**0.415**
7	358	134	3320	0.728	0.097	0.172
8	230	0	3448	1.000	0.063	0.118
9	7	0	3671	1.000	0.002	0.004

Note. Metrics reflect ***pairwise*** evaluation over the full candidate set produced by exhaustive comparison. TP = true positives; FP = false positives.

The pairwise evaluation reported in Table 7 is based on an exhaustive pairwise comparison of over 12 million record pairs and is a case of extreme class imbalance. This level of class imbalance is typical of exhaustive candidate generation in entity resolution evaluations. The aviation use case motivating this study is focused on a more downstream task: given an aircraft type in one aviation safety data system, determine the best aircraft type match to a record in a different aviation system.

Because the downstream task is to select the single best match for each source record, a top-1 evaluation was used as the primary framing. This top-1 framing treats entity resolution as a one-to-one linkage decision layer over pairwise scores, selecting (for each left-hand record) at most one right-hand candidate. This is done by selecting the highest-scoring candidate among those classified as matches, consistent with recent work that distinguishes one-to-one matching from pairwise classification and emphasizes the role of post-processing in producing a single best link (Papadakis, Efthymiou, et al., 2023). Top-1 framing also reduces the interpretive distortion created by exhaustive candidate universes, where pairwise precision can be dominated by extreme class imbalance and may not reflect the practical quality of the one-best linkage decision. This aligns with recent critiques of benchmark protocols and evaluation settings that can be "too easy" relative to deployment-style candidate distributions (Papadakis, Kirielle, et al., 2023). When no candidate is classified as a match for a left record, no link is produced and the case is counted as a false negative in top-1 evaluation.

The top-1 results include a tie-breaking criterion when multiple candidate pairs exceeded the threshold and tied on score. When multiple candidates tied for the same score, tie-break rules were applied in the following order: exact type certificate match, exact model match, exact series match, and exact make match. Table 8 reports top-1 performance across thresholds. Using F1 to

balance precision and recall, the best overall top-1 performance was observed at $t = 3$ (precision = 0.759, recall = 0.620, F1 = 0.682). Top-1 evaluation yields substantially higher precision than pairwise evaluation because it limits each left record to at most one predicted link.

Table 8

Baseline Matcher Performance (Top-1 Best Match per FAA Row)

Threshold (t)	TP	FP	FN	Precision	Recall	F1
0	2401	1277	1277	0.653	0.653	0.653
1	2401	1275	1277	0.653	0.653	0.653
2	2399	1216	1279	0.664	0.652	0.658
3	**2279**	**724**	**1399**	**0.759**	**0.620**	**0.682**
4	2049	483	1629	0.809	0.557	0.660
5	1629	184	2049	0.899	0.443	0.593
6	1369	85	2309	0.942	0.372	0.534
7	354	12	3324	0.967	0.096	0.175
8	230	0	3448	1.000	0.063	0.118
9	7	0	3671	1.000	0.002	0.004

Note. Metrics reflect ***top-1*** evaluation using the best match per FAA row after exhaustive pairwise comparisons. TP = true positives; FP = false positives. **(TP + FN= 3678)**

Answer to RQ1. A traditional feature-matching approach can recover a substantial share of FAA-to-CICTT links when evaluated under a top-1 linkage framing, achieving a best F1 score of 0.682 at threshold $t = 3$. For taxonomy-to-taxonomy, schema-aligned aircraft type matching, traditional ER is useful for surfacing many easier linkages but does not provide reliable large-scale normalization without additional blocking techniques, more refined feature matching rules, and/or considerable manual review.

Results for RQ2a: MMS taxonomy-to-taxonomy baseline and LH experiments.

Table 9 details the performance of Ditto MMS configurations across five negative sampling

strategies. Each configuration was evaluated on the held-out test split generated for that configuration. On the CICTT-to-FAA held-out test partition, the Baseline 1:10 configuration achieved the highest F1 score of 0.989, with all runs having an F1 no lower than 0.883 across held-out test splits. Recall was high for all held-out test sets as well as evaluation-only stress tests, while the negative sampling configuration noticeably affected false positive counts and precision. The most extreme case of a 1:90 ratio of positive to negative pairs produced the highest F1 among the reported MMS configurations for the fully exhaustive evaluation.

Table 9

MMS baseline Ditto results across negative sampling configurations (test + evaluation-only stress tests)

Dataset	Evaluation	TP	FN	FP	TN	Precision	Recall	F1
Baseline 1:10	**Baseline 1:10**	**366**	**2**	**6**	**3,671**	**0.984**	**0.995**	**0.989**
	Eval Canadair	9	0	67	29,348	0.118	**1.000**	0.212
	Eval Random (10)	10	0	69	31,141	0.127	**1.000**	0.225
	Eval Random (100)	99	1	756	366,944	0.116	**0.990**	0.207
	Exhaustive	3,258	5	23,407	10,620,499	0.122	**0.998**	0.218
Baseline LH 1:3	Baseline LH 1:3	360	8	37	448	0.907	**0.978**	0.941
	Eval Canadair	9	0	103	29,312	0.080	**1.000**	0.149
	Eval Random (10)	10	0	3,080	28,130	0.003	**1.000**	0.006
	Eval Random (100)	100	0	27,640	340,060	0.004	**1.000**	0.007
	Exhaustive	3,258	5	760,019	9,883,887	0.004	**0.998**	0.009
Baseline LH 1:10	Baseline LH 1:10	359	9	45	1,381	0.889	**0.976**	0.930
	Eval Canadair	9	0	61	29,354	0.129	**1.000**	0.228
	Eval Random (10)	10	0	359	30,851	0.027	**1.000**	0.053
	Eval Random (100)	100	0	4,038	363,662	0.024	**1.000**	0.047
	Exhaustive	3,254	9	130,945	10,512,961	0.024	**0.997**	0.047
Baseline LH 1:50	Baseline LH 1:50	354	14	57	9,348	0.861	**0.962**	0.909
	Eval Canadair	9	0	27	29,388	0.250	**1.000**	0.400
	Eval Random (10)	10	0	73	31,137	0.120	**1.000**	0.215
	Eval Random (100)	99	1	533	367,167	0.157	**0.990**	0.270
	Exhaustive	3,230	33	15,112	10,628,794	0.176	**0.990**	0.299
Baseline LH 1:90	Baseline LH 1:90	343	25	66	20,240	0.839	**0.932**	0.883
	Eval Canadair	9	0	8	29,407	0.529	**1.000**	0.692
	Eval Random (10)	10	0	28	31,182	0.263	**1.000**	0.417

Eval Random (100)	99	1	205	367,495	0.326	**0.990**	0.490
Exhaustive	3,208	55	5,984	10,637,922	0.349	**0.983**	0.515

Note. Metrics are reported using pairwise precision, recall, and F1 computed from TP, FP, FN, and TN counts. "Baseline 1:10" rows reflect evaluation on the held-out test split for each corresponding training configuration. "Eval Canadair," "Eval Random (10)," and "Eval Random (100)" are evaluation-only stress-test files constructed by pairing a small set of known matched records with an exhaustive set of non-matching candidates, producing highly imbalanced candidate sets. "Exhaustive" reflects inference over the full CICTT-to-FAA candidate universe. Because each evaluation setting contains a different number of candidate pairs, TP/FP/FN/TN counts are not directly comparable across rows.

For comparisons with the $recordlinkage$ baseline used for RQ1, a top-1 linkage evaluation was computed for the Ditto taxonomy-to-taxonomy models using the exhaustive candidate pair evaluation. The exhaustive evaluation cases removed all many-to-one relationships evaluating a total of 3,263 possible matches. This metric better reflects the downstream linkage use case and provides a more common ground to compare RQ1 results with RQ2 results. For each CICTT identifier, the highest-confidence FAA candidate among those predicted as a match was selected as the proposed link and compared to the gold mapping. In these runs, every CICTT identifier had at least one predicted match, so exactly one proposed link was produced per left record. This method produces at most one proposed link per CICTT identifier and avoids interpreting millions of non-links as false positives.

Table 10

Ditto Taxonomy-to-Taxonomy MMS Top-1 Results across Negative Sampling Configurations

Dataset	TP	FP	FN	Precision	Recall	F1
Baseline 1:10	**2886**	**377**	**377**	**0.884**	**0.884**	**0.884**
Baseline LH 1:3	1510	1753	1753	0.463	0.463	0.463
Baseline LH 1:10	2016	1247	1247	0.618	0.618	0.618
Baseline LH 1:50	2823	440	440	0.865	0.865	0.865
Baseline LH 1:90	2917	346	346	0.894	0.894	0.894

Note. Metrics reflect top-1 entity-level evaluation with at most one predicted link per left-hand CICTT identifier (N = 3,263). Because one link is always selected for each left record, precision and recall reduce to TP/N, and F1 equals the same value. TP, FP, and FN are counted per left record rather than per candidate pair.

RQ2a Error Analysis for Top-1 Deep Learning Baseline. Table 11 provides examples of records correctly matched by Ditto but not by the feature-based baseline. The model is able to learn patterns like “NORTH AMERICAN” = “AJ” and “DASSAULT” = “AMD” showing that the context is being developed not only for abbreviations but also relationships between aircraft designs that have been sold from one manufacturer to another. The model also shows the ability to manage the fields from multiple columns simultaneously. The model shows the ability to, for example, match information from the CICTT model with the FAA series. This would never be possible with column-based feature matching.

Table 11

Qualitative Audit of Records Identified only by the Deep Learning Model

CICTT Make	FAA Make	CICTT Model	FAA Model	CICTT Series	FAA Series
NORTH AMERICAN	AJ	AJ1	1	AJ1	1
KUBICEK	BB	BB30	30	XR	XR
CESSNA	CE	R172	172	G	R172G
LOCKHEED	L	749	49	79	74979
STINSON	SR	SR10	10	B	B
YAKOVLEV	YAK	YAK18	YAK	T	18T
DASSAULT	AMD	MYSTERE FALCON 50	50	MYSTERE FALCON 50	50
FAIRCHILD	AMERAP	PILGRIM FC2	PILGRM	PILGRIM FC2	FC2
HAWKER	HS	HAWKER 850	125	XP	850XP

Across error audits, the most common failure modes were near-family confusions (adjacent variants within the same aircraft family), sparse or ambiguous series tokens (including single-letter series), and labeling/taxonomy quality issues (e.g., legacy mislabels or corrupted entries). These patterns suggest that residual errors are driven less by simple lexical variation and

more by ambiguity in how type granularity and manufacturer lineage are represented across sources.

Additional useful observations were made in the error auditing. For example, the records that were matched in the feature-based model but not with Ditto very often had single letters to represent series. The tokenization of single letters is a known challenge for transformer-based models. Importantly, some "incorrect matches" were identified as improperly manually labeled data. CICTT identifier 7939 should have been labeled as 7946 and CICTT identifier 6582 should have been labeled 6601. This error analysis supports a further refined gold dataset.

Results for RQ2b: MMS schema variation and union MMS. Table 12 reports MMS results for schema variation and union training. The CICTT-to-Registry and FAA-to-Registry runs produced F1 values of 0.974 and 0.977, respectively. The MMS union model achieved F1 = 0.985 on the union test partition. When the union model was evaluated separately on the original source test partitions, the F1 values were 0.993 on the CICTT-to-FAA test split and 0.977 on the CICTT-to-Registry test split. The CICTT-to-FAA rows reproduce the Baseline 1:10 test-split results from Table 9 and add the corresponding union-model evaluation on the same CICTT-to-FAA test split to support a direct comparison.

Table 12

MMS Registry and MMS union model results (overall + per-source evaluation)

Model	Evaluation	TP	FN	FP	TN	Precision	Recall	F1
Union MMS	**Union MMS**	**727**	**9**	**13**	**7341**	**0.982**	**0.988**	**0.985**
CICTT to FAA	CICTT to FAA	366	2	6	3671	0.984	0.995	0.989
	Union MMS	365	3	2	3675	0.995	0.992	0.993
CICTT to Registry	CICTT to Registry	359	9	10	3667	0.973	0.976	0.974
	Union MMS	362	6	11	3666	0.971	0.984	0.977

FAA to Registry	FAA to Registry	360	8	9	3668	0.976	0.978	0.977

Note. Metrics reflect pairwise evaluation on held-out test partitions. Rows where Model and Evaluation match report the source-specific model checkpoint evaluated on that source's test split. Rows with Evaluation = Union MMS report the union-trained checkpoint evaluated on the same source test split. The first row reports union-trained performance on the union test partition.

Results for RQ2c: MM models and MM union. Table 13 reports results for MM matching across source-specific models and the union-trained model. The first row summarizes the MM union model evaluated on the union test partition. For each single-source dataset, the table reports two evaluations, performance of the source-specific model checkpoint evaluated on that source's held-out test partition and performance of the union-trained checkpoint evaluated on the same source test partition. This side-by-side comparison provides a direct check of whether union training preserves source-specific performance while enabling a single model to support inference across heterogeneous right-hand schemas.

Table 13

MM results for source-specific models and union-trained evaluation across MM datasets

Model	Evaluation	TP	FN	FP	TN	Precision	Recall	F1
Union	**Union**	**2822**	**95**	**69**	**28800**	**0.976**	**0.967**	**0.972**
CICTT	CICTT	400	0	0	3970	1.000	1.000	**1.000**
	Union	400	0	0	3970	1.000	1.000	**1.000**
FAA	FAA	360	7	5	3637	0.986	0.981	**0.984**
	Union	354	13	4	3638	0.989	0.965	**0.977**
REGISTRY	REGISTRY	364	3	4	3638	0.989	0.992	**0.990**
	Union	361	6	9	3633	0.976	0.984	**0.980**
DOC8643	DOC8643	812	20	21	8196	0.975	0.976	**0.975**
	Union	816	16	18	8199	0.978	0.981	**0.980**
DOC8643 code	DOC8643 code	95	24	33	1148	0.742	0.798	**0.769**
	Union	96	23	16	1165	0.857	0.807	**0.831**
DOC8643 descriptions	DOC8643 descriptions	793	39	39	8178	0.953	0.953	**0.953**
	Union	795	37	22	8195	0.973	0.956	**0.964**
DOC8643 with drops	DOC8643 with drops	806	26	14	6564	0.983	0.969	**0.976**

Note. Rows where Evaluation matches the Model label report the performance of the source-specific MM model checkpoint evaluated on that source's held-out test partition. Rows with Evaluation = Union report performance of the MM union-trained checkpoint evaluated on the same source's held-out test partition. The first row (Model = Union) reports performance of the union-trained checkpoint on the union test partition.

Across datasets, the MM union model achieved precision 0.976, recall 0.967, and F1 0.972 on the union test partition. Source-specific MM models achieved high performance for the structured CICTT, FAA, Registry, and DOC8643 description-based configurations. Performance varied when the input schema was reduced to the DOC8643 code-only representation, reflecting the limited information content available in that setting. When evaluated on the individual source test partitions, the union-trained checkpoint produced performance that was generally comparable to the corresponding source-specific models, indicating that union training did not meaningfully degrade MM matching accuracy for the evaluated sources.

Answer to RQ2. Deep learning entity resolution using Ditto can match aircraft type between data systems with high accuracy on held-out labeled test partitions across both MMS and MM tasks, and it remains effective under schema variation and multi-source union training. However, evaluation under exhaustive or highly imbalanced candidate pools shows that performance is sensitive to negative candidate composition, with false positives increasing substantially under exhaustive-style evaluations. Overall, Ditto provides a practical aircraft type matching approach that generalizes across heterogeneous schemas more effectively than traditional feature matching, particularly when trained in union configurations, while still requiring careful candidate-generation and dataset construction to control false positives at scale.

Results for RQ3: In-Domain Transfer.

Table 14 reports quantitative results for BTS inference under transfer-style inference settings as well as a BTS-specific model fine-tuned on BTS for comparative purposes. Because transfer-style BTS inference was executed over an evaluation-only candidate-pair file rather than

a labeled test split, performance for the transfer setting was computed using a top-1 linkage strategy rather than pairwise classification metrics. Ditto inference produced a score for each BTS-CICTT candidate pair.

For evaluation, BTS records were treated as the left-hand entities, and for each BTS record the single highest predicted matching confidence score CICTT candidate was selected as the proposed link. This yields at most one proposed outcome per BTS record. This produced proposed links for most BTS records (N=240), with some records receiving no link when the top-ranked candidate was predicted as a non-match. The selected top-1 links were then merged to the BTS ground truth mapping to determine whether the proposed CICTT identifier was correct. The BTS-specific model achieved precision 0.974, recall 0.974, and F1 0.974 on the BTS test split. Under transfer-style inference, the model achieved F1 0.782 on the BTS exhaustive evaluation candidate set and F1 0.818 under make-constrained candidate generation. BTS eval-only reflects inference over the exhaustive BTS-CICTT candidate set, whereas BTS block eval-only reflects make-constrained candidate generation.

Table 14

BTS Model and Transfer Top-1 Results

Model	TP	FP	FN	Precision	Recall	F1
BTS eval-only	154	82	4	0.653	0.975	0.782
BTS block eval-only	166	40	34	0.806	0.830	0.818
BTS model	228	6	6	0.974	0.974	0.974

Note. Metrics reflect top-1 entity-level evaluation with BTS as the left-hand entity set (N = 240 unique BTS records). For each BTS record, the highest-confidence CICTT candidate among pairs predicted as matches was selected. If no candidate pair was predicted as a match, no link was produced. TP, FP, and FN are counted per BTS record, not per candidate pair.

Make-constrained candidate generation improved BTS transfer-style top-1 F1 from 0.782 to 0.818 (+0.036), while BTS-specific fine-tuning improved F1 to 0.974 (+0.192 vs make-constrained transfer).

Manual audit of top-1 transfer links (RQ3). Table 15 summarizes a manual review of 100 sampled top-1 predicted links per in-domain dataset. For each dataset, the audit treated the selected top-1 linkage as correct if the predicted CICTT aircraft type was judged to refer to the same aircraft type as the source record. Because full ground truth labels were not available for AIDS, NTSB, and NWSD, these results are reported as an audit of apparent match correctness rather than as formal precision and recall estimates over the full candidate set. For BTS, the audit provides a qualitative check that is consistent with the quantitative evaluation reported in Table 14.

Table 15

Manual review of sampled candidate matches for in-domain evaluation-only datasets

Evaluation Dataset	Manual Label
BTS exhaustive eval-only	76/100
BTS make-constrained eval-only	93/100
AIDS make-constrained eval-only	77/100
NTSB make-constrained eval-only	90/100
NWSD make-constrained eval-only	81/100

Note. Counts reflect a manual review of 100 randomly sampled top-1 predicted links per dataset. "Correct" indicates the predicted CICTT alignment was judged to refer to the same aircraft type as the source record. Results are descriptive and do not constitute full-dataset performance metrics.

Answer to RQ3. In-domain transfer using a Ditto model trained on other aviation sources can produce useful aircraft type alignments to CICTT for new in-domain sources, but performance depends on candidate-generation constraints. For BTS, transfer-style top-1 evaluation achieved F1 = 0.782 under exhaustive candidate generation and improved to F1 =

0.818 under make-constrained candidate generation, while a BTS-specific fine-tuned model achieved substantially higher performance (F1 = 0.974). For additional unlabeled in-domain sources (AIDS, NTSB, and NWSD), manual audits of sampled top-1 links indicate that transfer produces a plausible normalization starting point, but results should be interpreted as descriptive and conditional on the executed make-constrained candidate-generation strategy.

Hypothesis Testing (Baseline vs Deep Learning). $H1_0$ stated that there is no difference in performance between the traditional $RecordLinkage$ approach (baseline matcher) and the Ditto deep learning approach while $H1_a$ stated that there is a significant difference in performance. To formally evaluate these hypotheses, a McNemar's Test was performed on the paired predictions of the baseline $RecordLinkage$ using exhaustive pairwise comparison. A new Ditto dataset was constructed evaluate the trained model only on the test set, but using exhaustive negative pairs. This ensured that the model was being evaluated on data that it had never seen before, but the model was also using the same negative candidate pair pool as the traditional feature-based approach. So the top-1 best-performing match based on the Ditto test set generated with an exhaustive negative candidate pair generation strategy and the traditional baseline model results were filtered to the test rows from the Ditto test data. The McNemar test was selected to compare the predictive accuracy of two models on the same dataset, focusing on the discordant pairs where the models disagree. The test was conducted on a total of 364 unique CICTT source records under a top-1 exhaustive evaluation strategy for both models. Table 16 shows the contingency table of correct and incorrect model responses for the two models.

Table 16

Contingency Table and McNemar's Test Results for Top-1 Linkage (N = 364)

	Ditto Correct	Ditto Incorrect
Baseline Matcher Correct	199	25
Baseline Matcher Incorrect	81	59

Note. Correctness is defined as the model's top-1 predicted link matching the manually curated ground truth. The upper right and lower left cells represent the discordant pairs used to calculate the Chi-squared statistic. $\chi^2(1, N = 364) = 28.54$, $p < 0.001$.

The statistical analysis produced a chi-squared statistic $\chi^2(1, N = 364)$ of 28.54 with a p-value of less than 0.001, leading to the rejection of the null hypothesis $H1_0$ in favor of the alternate hypothesis $H1_a$. The high chi-squared statistic is supported by the contingency table showing that Ditto correctly matched 81 pairs not matched by the baseline matcher and 25 records were correctly matched by the baseline matcher that were not resolved with the Ditto top-1 result on the exhaustive set of comparisons using left records from the test set. Both models missed 59 rows from the matches in the test set.

Deployment (As Executed)

This section describes the Deployment phase of the CRISP-DM process as applied in this study. The Deployment phase was implemented through three components: exploring the aviation safety use case, creating GitHub artifacts, and documenting results.

Explore Aviation Safety Use Case. The aviation safety use case explored in this study is the normalization of aircraft type representations across multiple aviation data sources to enable cross-source risk analysis. This use case requires two capabilities. First, aircraft type entries from heterogeneous sources must be aligned to a consistent reference representation. Second, the

resulting links must be expressed in a form that can be joined to downstream accident, incident, utilization, or registry tables for analysis.

In the as-executed workflow, the trained Ditto models were applied to candidate aircraft type pairs drawn from multiple aviation sources. Predicted links were then reviewed to assess whether the model produced plausible cross-source alignments across both structured and free-text aircraft type representations. For in-domain sources without full ground truth labels, this use case was explored through a structured qualitative review process, supported by a top-1 linkage strategy that selected a single proposed alignment per left-hand record based on the model confidence score. This approach produced a practical candidate mapping that can be inspected and used as a starting point for normalization in an applied safety analysis setting. For the in-domain sources, candidate pairs were generated using a make-constrained comparison strategy, comparing each left-hand record only to CICTT candidates within the same standardized make.

The make-model union model learned aircraft type signals under heterogeneous input schemas, including variation in which fields were available and how aircraft type was expressed. Transfer results indicate that the union model can produce plausible candidate alignments for new in-domain sources, but that performance improves when a source-specific model can be trained using ground truth labels. The BTS case illustrates this difference, where a BTS-trained model achieved substantially stronger performance than transfer-style evaluation.

Because all sources were aligned to the CICTT taxonomy in this study, the resulting CICTT identifiers can serve as a shared reference key to support joins across datasets. When a use case requires higher confidence, predicted links can be manually reviewed and corrected, allowing analysts to focus effort on ambiguous or inconsistent records rather than reviewing every row.

Once aircraft types are normalized to a shared CICTT reference, multiple safety event sources can be aggregated and compared to utilization sources. This supports analyses such as normalized event rates by make-model aircraft type, using denominators such as aircraft counts or operational activity where available. The trained models can be applied directly to prepared Ditto inference inputs such as all_pairs.txt files. If a safety application requires a different level of granularity, an additional union model would need to be trained using datasets prepared at the corresponding hierarchy level. The make-model aggregation process derived from labeled make-model-series data provides a blueprint for preparing these additional datasets and retraining the corresponding models.

Create GitHub Artifacts. Project artifacts were organized into two GitHub repositories. This supports reproducibility and allows independent inspection of the study workflow. It also makes the labeled aircraft type data available for use by other researchers. The trained Ditto model is provided as a resource for aviation safety analysts who would like to apply the approach to their own data.

The first repository contains the code used to preprocess the data and run the *RecordLinkage* baseline model. This repository is available at https://github.com/karnabryan/aircraft_er (Bryan, 2026a) and is described further in Annex A. Annex A includes selected code excerpts and a more detailed description of the repository contents.

The second repository is a fork of the Ditto framework, originally available at https://github.com/megagonlabs/ditto (Bryan, 2026b), and is available as https://github.com/karnabryan/ditto_aircraft_er. This repository is described further in Annex B. Because this study used Ditto's existing structure, the fork includes additional aircraft-specific

configuration files and run scripts that are not intended as general-purpose enhancements for Ditto users. No changes were made to the underlying Ditto model code that would generalize beyond this study context.

Document Results. The preparation of this manuscript was the primary method of documenting results for this study. The study workflow is documented in the main chapters, and code traceability and implementation details are described in Annex A and Annex B.

Evaluation of the Findings

This section interprets the study results reported in light of the study's conceptual framing and the related literature summarized in Chapters 1 and 2. The study is organized using the CRISP-DM framework (Shearer, 2000) and evaluates an entity resolution artifact for linking heterogeneous aviation data sources by aircraft type in the absence of shared identifiers. Interpretations are organized by the study's three research questions.

RQ1: Traditional Entity Resolution Performance

The $RecordLinkage$ baseline results indicate that feature-based entity resolution can recover a substantial portion of FAA-to-CICTT links when the sources are schema-aligned and comparable fields exist, but performance degrades under exhaustive candidate generation due to extreme class imbalance shown in Table 7 (best pairwise F1 = 0.415). This behavior aligns with established ER pipeline theory in which candidate generation and blocking are necessary to manage the combinatorial growth of comparisons and reduce false positives in large candidate universes (Christen, 2012; Elmagarmid et al., 2007; Herzog et al., 2007). The primary interpretation therefore focuses on reframing the evaluation as top-1 linkage selection which better matches the downstream linkage objective. As seen in Table 8 this improves performance, but the highest F1 achieved was 0.682.

RQ2: Deep Learning Entity Resolution Performance

Across labeled MMS tasks (RQ2a), Ditto demonstrates strong held-out performance, supporting the conclusion that modern transformer-based ER approaches can match aircraft type between data systems under the executed supervised settings. Negative sampling and evaluation framings were also assessed (Table 9). Evaluation settings were shown to substantially influence conclusions about model quality. In this study, held-out performance under a constructed negative sampling distribution differs from performance under stress-test and exhaustive-style candidate pools, where precision drops substantially while recall remains high. This pattern aligns with Papadakis et al.'s (2023) critique that commonly used benchmarks and protocols may be "too easy" relative to real-world variety and candidate imbalance, and it supports treating candidate generation and negative sampling as experimental variables rather than secondary preprocessing choices.

The MMS schema variation and union MMS model runs (RQ2b) focused on schema heterogeneity (Table 12). Ditto's COL/VAL serialization is designed to preserve attribute context while relaxing strict schema alignment requirements (Li et al., 2021). In this study, Ditto remained effective under schema variation and union training. The flexibility of the serialization to automatically adapt to heterogeneous and mixed schemas is well matched to the variety observed in aviation aircraft type representations across sources.

In the Make-Model (MM) level of hierarchy (RQ2c), results for schema variation and union training showed that Ditto performance on held-out test sets remained strong for this level of hierarchy and schema adaptation. Further, the union model trained on pooled candidate pairs had comparable performance to source-specific MM models when evaluated on the same source-specific held-out test partitions (Table 13). These results are consistent with the broader direction

of recent frameworks and toolkits that aim to reduce schema-specific feature engineering and improve scalability across heterogeneous sources (Efthymiou et al., 2023). Although this study did not implement explicit domain adaptation architectures, the union-training results are directionally consistent with multi-source and domain adaptation research that targets generalization across sources with differing distributions and limited labels (Jin et al., 2021; Tu et al., 2022; Trabelsi et al., 2022). Similarly, the MMS and MM experiments on the same data are aligned with current directions on multi-granularity and intent-dependent entity resolution (Fu et al., 2020; Yao et al., 2022; Genossar et al., 2023), although they do not implement a hierarchical model framework.

RQ3: In-Domain Transfer

The in-domain transfer results indicate that transfer-style inference can produce correct alignments for heterogeneous aviation sources under the executed candidate-generation strategy, with improved performance under make-constrained candidate generation and substantially improved performance when source-specific labels are available for fine-tuning (Table 14). The BTS results mirror findings from recent domain adaptation and multi-source ER literature that transfer performance is sensitive to domain shift and improves when even modest target-domain supervision is available (Tu et al., 2022; Trabelsi et al., 2022). The improvement observed under make-constrained candidate generation is also consistent with the practical reality emphasized across recent ER systems work that transfer and inference quality depends on upstream candidate restriction choices that shape the model's effective decision environment (Efthymiou et al., 2023).

For AIDS, NTSB, and NWSD, results are summarized via manual audits of sampled top-1 links rather than full-dataset metrics (Table 15). These audits provide descriptive evidence that

transfer-style alignments can be plausible under the executed constraints, while also reflecting a limitation highlighted in recent applied AI reviews in aviation: multi-source integration and annotation scarcity remain persistent barriers to scaling advanced analytics across heterogeneous aviation datasets (Yang & Huang, 2023; Nanyonga et al., 2025). Within the bounds of the executed evaluation strategy and available labels, the RQ3 findings support a conservative conclusion that in-domain transfer is feasible as a starting inference strategy, and that source-specific labeling and fine-tuning materially improve measurable performance when feasible.

Hypothesis Testing (H1)

To evaluate the significance of the performance gap between the traditional $RecordLinkage$ baseline and the Ditto deep learning framework, a formal McNemar's Test was performed. The test, reported in Table 16, indicates that the null hypothesis $H1_0$ stating that both models would exhibit identical error rates could be rejected in favor of the alternative hypothesis $H1_a$ that the models had a significant difference in predictive accuracy. Table 16's contingency table shows that 81 records could be successfully resolved by Ditto and not the baseline, while 25 records were resolved by the baseline and not Ditto. There were 59 record pairs that were not identified as matches by either model.

Evaluation Summary

Across the executed evaluation framings, deep learning models achieved higher linkage performance than the feature-based baseline on the schema-aligned taxonomy task, and McNemar's test on paired top-1 outcomes rejected $H1_0$ in favor of $H1_a$ (Table 16). Results also show that union-trained checkpoints preserved held-out performance across multiple source schemas (Tables 12-13), and transfer-style evaluation and audit summaries provide descriptive

evidence of alignment quality for additional in-domain sources under make-constrained candidate generation (Tables 14-15).

Limitations

This section describes limitations that may affect interpretation of the findings and the generalization of results beyond the evaluated datasets. Limitations arise from data quality and manual labeling, feasibility-driven candidate generation that relied on manual make-based blocking rather than an automated blocking strategy, and reproducibility constraints related to evaluation framing and stochastic model training.

The study relied on a finite set of public aviation sources with heterogeneous schema completeness and uneven field standardization. Aircraft type entries often contained duplicates or near-duplicates, inconsistent manufacturer naming, and series information that was missing, inconsistently encoded, or embedded within other fields. These properties increase ambiguity during labeling and matching and may reduce comparability across sources that represent aircraft type at different granularities.

The primary labeled gold dataset required manual mapping in the absence of common identifiers. Although this enabled supervised evaluation, manual labeling may include errors or subjectivity, particularly for aircraft families with overlapping naming conventions and cases where sources differ in how variants and series are represented. For several in-domain datasets, full ground truth was not feasible, so transfer findings rely partly on manual audits and qualitative review rather than complete quantitative evaluation. Indeed, this limitation was formally observed in the analysis where a qualitative audit of mutual failures revealed that the model's performance was limited by the quality of the data itself. Two "non-matches" were

identified as correct rejections of manual labeling errors in the ground truth, while a third was attributed to a corrupted entry in the master taxonomy ("OW-5M-5M").

Exhaustive candidate generation was not feasible for larger in-domain sources. In-domain transfer evaluation therefore used make-constrained pairing based on standardized make assignment applied through a manual process. This strategy improves feasibility and reflects a practical blocking approach, but it may exclude true matches when make is missing, mis-standardized, or represented under alternate naming conventions. In-domain transfer performance should therefore be interpreted as conditional on the executed candidate generation strategy. For operational use, an automated blocking and standardization step would need to be developed and evaluated to reduce manual effort and improve coverage.

Performance and reproducibility also depend on evaluation framing and stochastic processes. Held-out test sets reflect performance under constructed sampling distributions, while exhaustive and stress-test evaluations reflect highly imbalanced linkage conditions where false positives dominate. Top-1 evaluation better reflects linkage selection but measures the quality of selecting one proposed link per left entity rather than pairwise classification across the full candidate universe. Ditto training is stochastic, and this study reports single executed runs per configuration rather than repeated runs to quantify variance. Results should therefore be interpreted as representative of the executed procedure rather than as stable point estimates across random seeds and environments.

Summary

This chapter reported the as-executed workflow to build and evaluate aircraft type entity resolution across heterogeneous aviation sources that lack common identifiers. It documented

data and schema heterogeneity across sources and described the construction of a manually curated gold dataset that enabled supervised evaluation and baseline comparisons.

For RQ1, the traditional $RecordLinkage$ baseline showed limited performance under exhaustive pairwise evaluation because extreme class imbalance drives false positives, while a top-1 framing improved practical linkage selection but still left substantial residual error and required schema-aligned feature engineering per source pair. For RQ2, Ditto-based deep learning models achieved substantially stronger performance on the curated taxonomy task and remained effective under schema variation and union training, supporting multi-source inference without meaningful degradation in source-specific performance. For RQ3, in-domain transfer was evaluated on noisier datasets using make-constrained candidate generation, top-1 linkage selection, and manual audits, with BTS providing a labeled in-domain case that demonstrated the benefit of source-specific fine-tuning compared to the in-domain transfer. A formal McNemar's test on the model developed in support of RQ1 and RQ2 yielded a significant result, calling for a rejection of the null hypothesis in favor of a significant difference in performance between the two models.

The findings indicate that deep learning entity resolution can outperform a transparent feature-matching baseline for aircraft type normalization. A deep learning approach can accommodate schema heterogeneity and support a multi-source approach through a generic serialization approach. In-domain transfer shows promise under practical labeling and candidate-generation constraints, particularly when the target sources share similar schema and naming conventions with the sources used to train the multi-source model.

Chapter 5: Implications, Recommendations, and Conclusions

The problem addressed in this study is the challenge of merging aviation data from diverse sources that lack common keys, hindering safety analysts' ability to assess risk using multiple information sources. The purpose of this constructive research study is to develop an entity resolution model that can connect disparate aviation databases by matching across data sources using aircraft make-model-series specifications.

This study used the CRISP-DM workflow to build and evaluate an aircraft type entity resolution artifact for linking heterogeneous aviation data sources without shared identifiers. Using a manually curated FAA to CICTT gold dataset, a transparent feature-based baseline was compared with a deep learning model for the schema-aligned taxonomy-to-taxonomy case. Follow-on experimentation with the Ditto model explored schema-aligned, schema-varied, union-trained, hierarchical variation, and in-domain transfer settings. This showed that the Ditto model remained effective under schema heterogeneity and multi-source inference. Transfer evaluation produced useful top-1 linkages on unlabeled, heterogeneous sources. Limitations include a lack of source standardization, partial ground truth, make-constrained candidate generation relying on manual blocking, and the use of a stochastic model for training and evaluation that may provide challenges for generalization.

This chapter discusses implications framed in the context of continued operational safety, where aircraft type is a critical fleet aggregator for estimating hazard frequency and exposure and for targeting safety interventions consistent with the FAA's Monitor Safety/Analyze Data (MSAD) continued operational safety workflow (Federal Aviation Administration, 2023; National Academies of Sciences, Engineering, and Medicine, 2022). The implications are

organized by research questions and the hypothesis test, followed by recommendations for both practice and research.

Implications

This section addresses the implications of the study organized by the three research questions. The first focused on a deterministic feature-based baseline entity resolution model. The second focused on a deep learning implementation which included experimentation on negative candidate generation, multi-source learning, and models at multiple levels of hierarchy. The final research question focused on in-domain transfer learning for heterogeneous, unlabeled data.

Traditional Entity Resolution (RQ1)

The $recordlinkage$ library's feature-based baseline shows that deterministic entity resolution can recover a portion of true links for aircraft type when the two sources are schema-aligned and comparable fields exist. Performance degrades under exhaustive candidate generation because false positives dominate. Under a top-1 linkage-selection framing, the baseline provides a reasonable "best guess" for many records, but it does not support reliable, automated normalization without additional review. This is partially due to the scope focus on the matching phase of the pipeline which treated blocking as an out-of-scope future step. Integrated blocking could improve precision.

A key implication is that this baseline is not portable across the broader aviation ecosystem without repeated schema engineering. The nine data sources used in this analysis all encode aircraft type in different ways, many with fewer fields, and information on make, model, and series embedded in free text or inconsistent fields. The feature-based paradigm requires dedicated schema matching for each source pair combination. This reinforces that deterministic

matching is best positioned as a transparent benchmark or a solution in cases where schema alignment already exists, rather than as a scalable integration strategy across heterogeneous safety and utilization sources. This matters for the continued operational safety setting described in Chapter 2 because these methods advocated for the aggregation of several data sources, including those which have been revealed in this study to be unsuitable for the clean, schema-aligned case (Federal Aviation Administration, 2023; Federal Aviation Administration, 2017).

Deep Learning Entity Resolution (RQ2)

Across the labeled MMS and MM tasks, Ditto's most consequential practical implication is its consistently high recall across the labeled evaluation settings. The model reliably recovers true matches even when aircraft type strings differ in formatting and when sources vary in which aircraft type attributes are available. Precision remains the main operational risk under exhaustive candidate universes, but an integrated blocking strategy could improve precision. Even without a blocking step, the top-1 linkage-selection based maximizing match scores for each left-hand record already provides relatively good performance (e.g., an F1 of 0.884). Qualitative error analysis showed that most incorrect best matches were very close to a correct match and almost always a similar variant. The top-1 use case provides an initial capability for an analyst to apply the model as a normalization tool to get the most likely mapping per left record and review exceptions. These exceptions can be reviewed by examining other high-scoring candidate matches, not just the top-ranked prediction.

The most significant implication of the Ditto results is the demonstrated schema flexibility. The generic serialization enables matching without strict column-to-column alignment, which fits observed variation across aviation data sources. Union training strengthens this implication by producing a single reusable model that preserves source-specific performance

while supporting multi-source inference. This reduces the need to build and maintain separate matchers per source pair.

In-Domain Transfer (RQ3)

The transfer results show promise toward the goal of a fully automated solution across heterogeneous aviation sources that can normalize records by aircraft type and use this as a common identifier enabling the calculation of refined risk-based metrics. Current capability was demonstrated on the union-trained MM model with some degree of success. Transfer of the union model to in-domain heterogeneous and "dirty" datasets showed that without source-specific training, the top-1 best-guess produced a starting point for further inspection. Further labeling of the BTS data source for evaluation demonstrated that a blocking strategy improved results (from F1 = 0.782 under exhaustive candidate generation to F1 = 0.818 under make-constrained candidate generation).

A dedicated model trained on the BTS dataset achieved an F1 of 0.974. This shows that source-specific labeling and fine-tuning can improve performance. Adding new sources to the union model as new labeled data becomes available could further improve performance, and additional in-domain transfer could benefit from additional training sources in the union model. Transfer learning provides a path to the normalization of data across a broad range of heterogeneous sources. The ability to produce analytics over a broad range of sources enables higher fidelity risk calculations and better predictive capability for aviation safety analyses. (National Academies of Sciences, Engineering, and Medicine, 2022).

Hypothesis Test

The statistically significant advantage of the deep learning framework was confirmed through a formal hypothesis test using the McNemar's Test on a strictly held-out test set (N =

364). The results yielded a chi-squared statistic of $\chi^2(1, N = 364) = 28.54$ with a p-value < 0.001, providing empirical evidence to reject the null hypothesis $H1_0$ in favor of a significant difference in predictive accuracy $H1_a$. Under an exhaustive evaluation framing, the deep learning model successfully resolved 81 unique records that were not identified by the deterministic baseline. While the baseline correctly identified 25 records that Ditto missed involving exact numerical shorthand or perfectly aligned strings, its performance was bounded by reliance on schema-aligned feature engineering and limited ability to capture cross-field semantic equivalences. Based on this statistical validation, the Ditto framework provides a measurable shift in data integration capability, establishing a credible foundation for automated aviation safety analysis.

The 59 mutual failures identified in the test set underscore the inferential boundaries of the model. These cases represent instances where the underlying data does not demonstrate patterns that can be generalized from the training set, sometimes due to taxonomy corruption or legacy labeling errors. Other times these missed identifications were due to inherent lack of standardization in definitions of model and series between manufacturers or the fact that aircraft designs (e.g., a fixed model and series) can be produced by multiple manufacturers or be sold from one manufacturer to another. While the model demonstrates very high performance on the labeled source populations evaluated in this study suggesting strong operational utility for known data, it cannot infer a match when the requisite lexical or semantic patterns are absent or the model lacks relevant context.

Recommendations for Practice

Based on the findings, a deep learning entity resolution approach is sufficiently mature to be used for aircraft type normalization on cross-source aviation safety analysis. Not only does

the approach provide a measurable improvement over a feature-based baseline on schema-aligned sources, a Ditto-based approach using COL/VAL formatted strings can support heterogeneous schemas and aircraft type descriptors without the need for schema-matching making it ideal for multi-source scenarios.

For deployment, results support using a union-trained make-model model to produce a single checkpoint for inference on heterogeneous schemas. Without further research, a simple blocking based on make-standardization and make-based blocking combined with a top-1 approach using the highest scoring match already provides a high degree of accuracy. This can be operationalized to a workflow that is scanned by a human analyst for validation in what would otherwise be an overwhelming manual task. Where a new in-domain source is operationally important, the BTS results indicate that even modest source-specific labeling and fine-tuning can yield large performance gains relative to transfer-only inference, enabling a practical pathway from source-agnostic normalization to higher-confidence source-specific models.

Further research is required to develop a robust, integrated blocking capability to fully operationalize the entity resolution pipeline. The findings of this study indicate that model performance is highly sensitive to the negative candidate generation strategy, which is directly influenced by the initial blocking step. Pairwise matching can result in an extreme case of class imbalance. This study demonstrated that model training needed both “hard” negative samples that are difficult to distinguish and a broad variety in negative sampling to ensure generalization. Future work should involve a comprehensive re-evaluation of the end-to-end pipeline, with a refined blocking strategy to analyze the interplay between refined blocking strategies, negative candidate generation, and final predictive accuracy.

The CICTT taxonomy can be used as a reference key to integrate heterogeneous aviation data sources by aircraft type and enable safety rate calculations across these disparate sources. By linking aircraft type information from heterogeneous sources to a CICTT identifier, this identifier can function as a shared join key across incident, accident, utilization, and registry tables. Practically, this enables analyses described in the aviation safety literature that require denominators and stratification, such as normalized event rates by make-model and comparisons across fleets or time periods (Federal Aviation Administration, 2023; National Academies of Sciences, Engineering, and Medicine, 2022).

Recommendations for Future Research

A primary takeaway of this study for the broader Entity Resolution community is that negative candidate selection strategies can fundamentally dominate conclusions regarding model quality. In this study, a notable degradation in performance was demonstrated between the negative candidate sets and the exhaustive set of negative candidates. The initial taxonomy-to-taxonomy case with 1:10 negative candidate generation showed an F1 score of 0.989 when evaluated against the 1:10 negative candidate pool and an F1 score of 0.218 when evaluated against the exhaustive set of negative candidates. This difference in F1 was entirely driven by precision as recall remained persistently high. Part of the Papadakis et al. (2023) critique of current benchmarks findings is that the datasets are often “too easy.” The authors claimed that datasets that are too small, are “too clean,” or too domain specific, calling for benchmarks with more records and greater variety. Matching baselines use the negative candidate sampling to evaluate precision, creating one of these “too easy” cases where high F1 scores are achieved which cannot be repeated against an exhaustive candidate pool.

Furthermore, the "learn-harder" (LH) experiments conducted in this study suggest that increasing "confusable" negatives does not automatically improve generalization. In fact, moderate LH settings underperformed the random-negative baseline. Only the most extreme LH ratio (1:90) improved exhaustive-style F1, implying that hard negatives are only beneficial when precisely calibrated to the specific deployment distribution and volume. Consequently, a future research agenda for the ER community should treat negative sampling and candidate generation as first-class experimental variables rather than secondary preprocessing details.

A significant contribution of this research is the curation and release of the FAA-to-CICTT gold-standard dataset, which effectively labels linkages in aircraft type definitions between five distinct aviation data sources. This dataset serves as a specialized "stress test" for multi-source frameworks like DADER, DAME, and TransER (Kirielle et al., 2022; Trabelsi et al., 2022; Tu et al., 2022) as it requires models to reconcile not only lexical variations but also the complex structural "shuffling" of attributes. Juxtaposition of the attributes in the fields (e.g., make, model, and series appearing in the wrong field) was observed in many of the taxonomy-to-taxonomy records that were correctly matched by Ditto and not by the feature-based $recordlinkage$ model.

This study further contributes to the field of hierarchical entity resolution by demonstrating how structured domain taxonomies can be operationalized within a deep learning framework. While earlier foundational work by Leitão et al. (2013) emphasized the necessity of exploiting structural relationships in hierarchical data, this research provides an empirical application of these concepts to the aircraft type hierarchy. By evaluating model performance at distinct levels of granularity, in this study make-model (MM) and make-model-series (MMS), this study confirms the observations of Fu et al. (2020) regarding the need for multi-level

matching networks to handle heterogeneous entities and provides an additional test case for the FlexER framework proposed by Genossar et al. (2023) to expand on the Ditto model.

The dataset provides a labeled multi-source, hierarchical dataset to further develop more sophisticated ER models and provides a blueprint for how domain-specific taxonomies can be operationalized as universal join keys. This contribution enables future researchers to move beyond pairwise classification and explore domain adaptation strategies necessary for truly scalable, global data integration in safety-critical environments.

Conclusions

This study demonstrates that deep learning can provide a practical and measurable improvement over a transparent feature-based entity resolution baseline for aircraft type normalization across heterogeneous aviation data sources. Using a constructive research process around a manually curated FAA-to-CICTT gold dataset, the study showed that Ditto not only outperformed the traditional matcher on the core taxonomy-to-taxonomy task, but also remained effective under schema variation, union-trained multi-source inference, and in-domain transfer settings. The formal McNemar's test on a strictly held-out top-1 evaluation set confirmed a statistically significant difference in predictive accuracy between the approaches, supporting rejection of the null hypothesis.

The results showed that operational performance depends heavily on candidate generation, blocking, and data quality, especially in highly imbalanced exhaustive-linkage settings. These findings indicate that a deep learning entity resolution framework paired with improved blocking, targeted labeling, and continued dataset curation offers a credible foundation for scalable aircraft type normalization and, by extension, more robust cross-source aviation safety analysis. More broadly, the study provides a practical model for using entity resolution

with a standardized taxonomy as a surrogate join key across heterogeneous data sources, enabling a multi-source analytic approach that could be adapted to similar problems.

References

Agarwal, A., Gite, R., Laddha, S., Bhattacharyya, P., Kar, S., Ekbal, A., Thind, P., Zele, R., & Shankar, R. (2022). Knowledge Graph—Deep Learning: A Case Study in Question Answering in Aviation Safety Domain. Language Resources and Evaluation Conference. https://doi.org/10.48550/arxiv.2205.15952

Aguiar, M., Stolzer, A., & Boyd, D. D. (2017). Rates and causes of accidents for general aviation aircraft operating in a mountainous and high elevation terrain environment. Accident Analysis and Prevention, 107, 195–201. https://doi.org/10.1016/j.aap.2017.03.017

Ahadh, A., Binish, G. V., & Srinivasan, R. (2021). Text mining of accident reports using semi-supervised keyword extraction and topic modeling. Process Safety and Environmental Protection, 155, 455–465. https://doi.org/10.1016/j.psep.2021.09.022

AirGuide Business. (2023). Boeing 737 MAX officially returns within China with China Southern flight. Gale Academic OneFile, NA.

Akritidis, L., Fevgas, A., Bozanis, P., & Makris, C. (2020). A self-verifying clustering approach to unsupervised matching of product titles. The Artificial Intelligence Review, 53(7), 4777–4820. https://doi.org/10.1007/s10462-020-09807-8

Amin, N., Yother, T., Johnson, M. E., & Rayz, J. (2022). Exploration of Natural Language Processing (NLP) Applications in Aviation. Collegiate Aviation Review International, 40(1), 204–2016.

Ancel, E., Shih, A. T., Jones, S. M., Reveley, M. S., Luxhøj, J. T., & Evans, J. K. (2015). Predictive safety analytics: Inferring aviation accident shaping factors and causation. Journal of Risk Research, 18(4), 428–451. https://doi.org/10.1080/13669877.2014.896402

Backes, T., Hienert, D., & Dietze, S. (2022). Towards hierarchical affiliation resolution: Framework, baselines, dataset. International Journal on Digital Libraries, 23(3), 267–288. https://doi.org/10.1007/s00799-022-00326-1

Badanik, B., Janossy, M., & Dijkstra, A. (2021). The use of expert judgement methods for deriving accident probabilities in aviation. Promet, 33(2), 205–216. https://doi.org/10.7307/ptt.v33i2.3634

Barlaug, N., & Gulla, J. A. (2021). Neural Networks for Entity Matching: A Survey. ACM Transactions on Knowledge Discovery from Data, 15(3), 1–37. https://doi.org/10.1145/3442200

Belin, T. R., & Rubin, D. B. (1995). A Method for Calibrating False-Match Rates in Record Linkage. Journal of the American Statistical Association, 90(430), 694–707. https://doi.org/10.2307/2291082

Bhattacharya, I., & Getoor, L. (2007). Collective entity resolution in relational data. ACM Transactions on Knowledge Discovery from Data, 1(1), 5. https://doi.org/10.1145/1217299.1217304

Binette, O., & Steorts, R. C. (2022). (Almost) all of entity resolution. Science Advances, 8(12), eabi8021. https://doi.org/10.1126/sciadv.abi8021

Blom, H. A. P., & Bloem, E. A. (2010). Modeling and estimation of accident rate and trend in air transport. 2010 13th International Conference on Information Fusion, 1–8. https://doi.org/10.1109/ICIF.2010.5711871

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135–146. https://doi.org/10.1162/tacl_a_00051

Bombardier. (2023, December 10). Business Aircraft: Challenger 350. https://businessaircraft.bombardier.com/en/aircraft/challenger-350

Boyd, D. D. (2017). A review of general aviation safety (1984-2017). Aerospace Medicine and Human Performance, 88(7), 657–664. https://doi.org/10.3357/AMHP.4862.2017

Brendel, A. B., Lembcke, T.-B., & Kolbe, L. M. (2022). Towards an Integrative View on Design Science Research Genres, Strategies, and Pivotal Concepts in Information Systems Research. ACM SIGMIS Database: The DATABASE for Advances in Information Systems, 53(4), 9–23. https://doi.org/10.1145/3571823.3571826

Brunner, U., & Stockinger, K. (2020). Entity matching with transformer architectures—A step forward in data integration. OpenProceedings.Org. https://search.datacite.org/works/10.5441/002/edbt.2020.58

Bryan, K. (2026). *aircraft_er* (Version 1.0) [Computer software]. GitHub. https://github.com/karnabryan/aircraft_er

Bryan, K. (2026). *ditto_aircraft_er* (Version 1.0) [Computer software]. GitHub. https://github.com/karnabryan/ditto_aircraft_er

Bureau of Transportation Statistics. (2025, September 22). Transtats: Air Carriers: T-100 Domestic Segment (All Carriers). https://www.transtats.bts.gov/

Buselli, I., Oneto, L., Dambra, C., Verdonk Gallego, C., García Martínez, M., Smoker, A., Ike, N., Pejovic, T., & Ruiz Martino, P. (2021). Natural language processing for aviation safety: Extracting knowledge from publicly-available loss of separation reports. Open Research Europe, 1, 110. https://doi.org/10.12688/openreseurope.14040.2

Cheng, Y., Jiao, Y., Wei, W., & Wu, Z. (2019). Research on construction method of knowledge graph in the civil aviation security field. 2019 IEEE 1st International Conference on Civil

Aviation Safety and Information Technology (ICCASIT), 556–559. https://doi.org/10.1109/ICCASIT48058.2019.8973190

Christen, P. (2008). Febrl: A freely available record linkage system with a graphical user interface. Proceedings of the Second Australasian Workshop on Health Data and Knowledge Management - Volume 80, 80, 17–25.

Christen, P. (2012). Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection (1. Aufl. ed.). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31164-2

Christen, P. (2019). Data linkage: The big picture. Harvard Data Science Review. https://doi.org/10.1162/99608f92.84deb5c4

Churchwell, J. S., Zhang, K. S., & Saleh, J. H. (2018). Epidemiology of helicopter accidents: Trends, rates, and covariates. Reliability Engineering & System Safety, 180, 373–384. https://doi.org/10.1016/j.ress.2018.08.007

Creswell, J. W., & Creswell, J. D. (2018). *Research design* (Fifth edition ed.). SAGE.

De Bruin, J. (2019). Python Record Linkage Toolkit: A toolkit for record linkage and duplicate detection in Python (Version v0.14) [Python]. Zenodo. https://doi.org/10.5281/zenodo.3559043

De Florio, F. (2016). Airworthiness: An introduction to aircraft certification and operations (Third Ed.). Elsevier. https://www.sciencedirect.com/book/9780080968025/airworthiness

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 7, 1–30. https://doi.org/10.5555/1248547.1248548

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In J. Burstein, C. Doran, & T.

Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, *10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

Dong, X. L., & Srivastava, D. (2015). Big Data Integration. Springer International Publishing. https://doi.org/10.1007/978-3-031-01853-4

Dunn, H. L. (1946). Record Linkage. American Journal of Public Health and the Nations Health, 36(12), 1412–1416. https://doi.org/10.2105/AJPH.36.12.1412

Durak, U., Becker, J., Hartmann, S., & Voros, N. S. (2018). Ontologies in aeronautics. In Advances in Aeronautical Informatics (pp. 67–85). Springer International Publishing AG. https://doi.org/10.1007/978-3-319-75058-3_6

Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., & Tang, N. (2018). Distributed representations of tuples for entity resolution. Proceedings of the VLDB Endowment, 11(11), 1454–1467. https://doi.org/10.14778/3236187.3236198

Edwards, A. L. (1948). Note on the “Correction for Continuity” in Testing the Significance of the Difference between Correlated Proportions. *Psychometrika*, *13*(3), 185–187. https://doi.org/10.1007/BF02289261

Efthymiou, V., Ioannou, E., Karvounis, M., Koubarakis, M., MacIejewski, J., Nikoletos, K., Papadakis, G., Skoutas, D., Velegrakis, Y., & Zeakis, A. (2023). Self-configured Entity

Resolution with pyJedAI (Vol. 1–Book, Section). https://dspace.library.uu.nl/handle/1874/436691

Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering, 19(1), 1–16. https://doi.org/10.1109/TKDE.2007.250581

Englehardt, E., Werhane, P. H., & Newton, L. H. (2021). Leadership, engineering, and ethical clashes at Boeing. Science and Engineering Ethics, 27(1), 12. https://doi.org/10.1007/s11948-021-00285-x

Etikan, I., Musa, S. A., & Alkassim, R. S. (2015). Comparison of Convenience Sampling and Purposive Sampling. American Journal of Theoretical and Applied Statistics, 5(1), Article 1. https://doi.org/10.11648/j.ajtas.20160501.11

Fan, M., Han, X., Fan, J., Chai, C., Tang, N., Li, G., & Du, X. (2023). Cost-Effective In-Context Learning for Entity Resolution: A Design Space Exploration (No. arXiv:2312.03987). arXiv. https://doi.org/10.48550/arXiv.2312.03987

Federal Aviation Administration. (2017). Order 8000.71—Aircraft Make, Model, and Series Taxonomy. Federal Aviation Administration. https://www.faa.gov/regulations_policies/orders_notices/index.cfm/go/document.information/documentID/1031146

Federal Aviation Administration. (2026, February 15). Aircraft Type Designators (ORDER JO 7360.1K). https://www.faa.gov/documentLibrary/media/Order/FAA_Order_JO_7360.1K_Aircraft_Type_Designators.pdf

Federal Aviation Administration. (2025a). Aircraft registration releasable aircraft database download [Dataset]. Author. https://www.faa.gov/licenses_certificates/aircraft_certification/aircraft_registry/releasable_aircraft_download/

Federal Aviation Administration. (2025b). FAA Wildlife Strike database [Dataset]. Author. https://wildlife.faa.gov/home

Federal Aviation Administration. (2023, October 13). Monitor safety/analyze data (ORDER 8110.107B). https://www.faa.gov/documentLibrary/media/Order/Order_8110.107B_Final.pdf

Federal Aviation Administration. (2024a, January 28). Data downloads. https://av-info.faa.gov/

Federal Aviation Administration. (2024b, January 28). FAA Aviation Safety Information Analysis and Sharing (ASIAS) System. https://www.asias.faa.gov/apex/f?p=100:1::::::

Fellegi, I. P., & and Sunter, A. B. (1969). A Theory for Record Linkage. Journal of the American Statistical Association, 64(328), 1183–1210. https://doi.org/10.1080/01621459.1969.10501049

Genossar, B., Shraga, R., & Gal, A. (2023). FlexER: Flexible Entity Resolution for Multiple Intents. Proceedings of the ACM on Management of Data, 1(1), 1–27. https://doi.org/10.1145/3588722

Getoor, L., & Machanavajjhala, A. (2012). Entity resolution: Theory, practice & open challenges. Proc. VLDB Endow., 5(12), 2018–2019. https://doi.org/10.14778/2367502.2367564

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learing. The MIT Press.

Halevy, A., Rajaraman, A., & Ordille, J. (2006). Data integration: The teenage years. Proceedings of the 32nd International Conference on Very Large Data Bases, 9–16.

Herkert, J., Borenstein, J., & Miller, K. (2020). The Boeing 737 MAX: Lessons for engineering ethics. Science and Engineering Ethics, 26(6), 2957–2974. https://doi.org/10.1007/s11948-020-00252-y

Herzog, T. N., Scheuren, F. J., & Winkler, W. E. (2007). Data Quality and Record Linkage Techniques. Springer Science & Business Media.

Huang, J., Hu, W., Bao, Z., & Qu, Y. (2020). Crowdsourced Collective Entity Resolution with Relational Match Propagation. 2020 IEEE 36th International Conference on Data Engineering (ICDE), 37–48. https://doi.org/10.1109/ICDE48307.2020.00011

Ilyas, I. F., & Chu, X. (2019). Data Cleaning. Association for Computing Machinery.

International Civil Aviation Organization. (2025a). DOC 8643—Aircraft Type Designators [Dataset]. Retrieved June 15, 2025, from https://www.icao.int/publications/DOC8643/Pages/default.aspx

International Civil Aviation Organization. (2006). Commercial Aviation Safety Team (CAST)/ International Civil Aviation Organization (ICAO) Common Taxonomy Team (CICTT): Adopting CICTT Taxonomies and Standards.

International Civil Aviation Organization. (2019). International standards for aircraft make, model, and series groupings: Business Rules. https://www.intlaviationstandards.org/Documents/AircraftMakeModelSeriesBusinessRules1.pdf

International Civil Aviation Organization (ICAO). (2025b). Commercial Aviation Safety Team (CAST)/ICAO Common Taxonomy Team (CICTT) [Dataset]. Author. https://www.intlaviationstandards.org/apex/f?p=240:2::::::

International Civil Aviation Organization (ICAO). (2026). Commercial Aviation Safety Team (CAST)/ICAO Common Taxonomy Team (CICTT). https://www.intlaviationstandards.org/apex/f?p=240:1

Javdani, D., Rahmani, H., Allahgholi, M., & Karimkhani, F. (2019). Deepblock: A novel blocking approach for entity resolution using deep learning. 2019 5th International Conference on Web Research (ICWR), 41–44. https://doi.org/10.1109/ICWR.2019.8765267

Jin, D., Sisman, B., Wei, H., Dong, X. L., & Koutra, D. (2021). Deep transfer learning for multi-source entity linkage via domain adaptation. *Proceedings of the VLDB Endowment*, *15*(3), 465–477. https://doi.org/10.14778/3494124.3494131

Kaufman, A. R., & Klevs, A. (2022). Adaptive Fuzzy String Matching: How to Merge Datasets with Only One (Messy) Identifying Field. Political Analysis, 30(4), 590–596. https://doi.org/10.1017/pan.2021.38

Khodizadeh-Nahari, M., Ghadiri, N., Baraani-Dastjerdi, A., & Sack, J.-R. (2021). A novel similarity measure for spatial entity resolution based on data granularity model: Managing inconsistencies in place descriptions. Applied Intelligence (Dordrecht, Netherlands), 51(8), 6104–6123. https://doi.org/10.1007/s10489-020-01959-y

Kierszbau, S., Klein, T., & Lapasset, L. (2022). ASRS-CMFS vs. RoBERTa: Comparing two pre-trained language models to predict anomalies in aviation occurrence reports with a

low volume of in-domain data available. Aerospace, 9(10), 591. https://doi.org/10.3390/aerospace9100591

Kierszbaum, S., & Lapasset, L. (2020). Applying distilled BERT for question answering on ASRS reports. 2020 New Trends in Civil Aviation (NTCA), 33–38. https://doi.org/10.23919/NTCA50409.2020.9291241

Konda, P., Das, S., Suganthan, P., Doan, A., Ardalan, A., Ballard, J. R., Li, H., Krishnan, G., & Raghavendra, V. (2016). Magellan. Proceedings of the VLDB Endowment, 9(12), 1197–1208. https://doi.org/10.14778/2994509.2994535

Kooli, N., Allesiardo, R., & Pigneul, E. (2018). Deep Learning Based Approach for Entity Resolution in Databases. Lecture Notes in Computer Science, Journal Article, 3–12. https://doi.org/10.1007/978-3-319-75420-8_1

Köpcke, H., Thor, A., & Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. Proc. VLDB Endow., 3(1–2), 484–493. https://doi.org/10.14778/1920841.1920904

Kouki, P., Pujara, J., Marcum, C., Koehly, L., & Getoor, L. (2019). Collective entity resolution in multi-relational familial networks. Knowledge and Information Systems, 61(3), 1547–1581. https://doi.org/10.1007/s10115-018-1246-2

Ledvinka, M., Lališ, A., & Křemen, P. (2019). Toward data-driven safety: An ontology-based information system. Journal of Aerospace Information Systems, 16(1), 22–36. https://doi.org/10.2514/1.I010622

Li, H., Li, S., Hao, F., Zhang, C. J., Song, Y., & Chen, L. (2024). BoostER: Leveraging Large Language Models for Enhancing Entity Resolution. Companion Proceedings of the ACM Web Conference 2024, 1043–1046. https://doi.org/10.1145/3589335.3651245

Li, Y., Li, J., Suhara, Y., Doan, A., & Tan, W.-C. (2023). Effective entity matching with transformers. The VLDB Journal. https://doi.org/10.1007/s00778-023-00779-z

Li, Y., Li, J., Suhara, Y., Wang, J., Hirota, W., & Tan, W.-C. (2021). Deep Entity Matching. ACM Journal of Data and Information Quality, 13(1), 1–17. https://doi.org/10.1145/3431816

Logeswaran, L., Chang, M.-W., Lee, K., Toutanova, K., Devlin, J., & Lee, H. (2019). Zero-Shot Entity Linking by Reading Entity Descriptions. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3449–3460). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1335

Luo, P., Li, M., & Li, Z. S. (2020). An Internet of Things (lot) perspective of understanding the Boeing 737 MAX crash. 2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai), 1–8. https://doi.org/10.1109/PHM-Shanghai49105.2020.9280967

Lv, M., Cao, X., Wu, T., & Li, Y. (2023). A Civil Aviation Customer Service Ontology and Its Applications. Data Intelligence, 5(4), 1063–1081. https://doi.org/10.1162/dint_a_00237

Marais, K. B., & Robichaud, M. R. (2012). Analysis of trends in aviation maintenance risk: An empirical approach. Reliability Engineering & System Safety, 106, 104–118. https://doi.org/10.1016/j.ress.2012.06.003

McNemar, Q. (1947). Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika*, *12*(2), 153–157. https://doi.org/10.1007/BF02295996

Megagon Labs. (2026). *Megagonlabs/ditto* [Python]. https://github.com/megagonlabs/ditto (Original work published 2020)

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In Y. Bengio & Y. LeCun (Eds.), 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings. http://arxiv.org/abs/1301.3781

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., & Raghavendra, V. (2018). Deep Learning for Entity Matching. Proceedings of the 2018 International Conference on Management of Data, 19–34. https://doi.org/10.1145/3183713.3196926

Nanyonga, A., Joiner, K., Turhan, U., & Wild, G. (2025). Applications of Natural Language Processing in Aviation Safety: A Review and Qualitative Analysis. AIAA SCITECH 2025 Forum. AIAA SCITECH 2025 Forum. https://doi.org/10.2514/6.2025-2153

National Academies of Sciences, E., and Medicine. (2022). Evaluation of the transport airplane risk assessment methodology (1 ed.). National Academies Press. https://doi.org/10.17226/26519

National Aeronautics and Space Administration. (2026, March 1). Aviation Safety Reporting System. https://asrs.arc.nasa.gov/search/database.html

National Research Council. (1998). Improving the continued airworthiness of civil aircraft. http://doi.org/10.17226/6265

National Transportation Safety Board. (2025). Census of US Civil Aviation Accidents [Dataset]. http://data.ntsb.gov/avdata

National Transportation Safety Board. (2024). In-flight structural failure, Alaska Airlines flight 1282. National Transportation Safety Board. https://www.ntsb.gov/investigations/Pages/DCA24MA063.aspx

Newcombe, H. B., Kennedy, J. M., Axford, S. J., & James, A. P. (1959). Automatic Linkage of Vital Records. Science, 130(3381), 954–959. https://doi.org/10.1126/science.130.3381.954

Paganelli, M., Buono, F. D., Baraldi, A., & Guerra, F. (2022). Analyzing how BERT performs entity matching. Proc. VLDB Endow., 15(8), 1726–1738. https://doi.org/10.14778/3529337.3529356

Papadakis, G., Efthymiou, V., Thanos, E., Hassanzadeh, O., & Christen, P. (2023). An analysis of one-to-one matching algorithms for entity resolution. The VLDB Journal, 32(6), 1369–1400. https://doi.org/10.1007/s00778-023-00791-3

Papadakis, G., Kirielle, N., Christen, P., & Palpanas, T. (2023). A Critical Re-evaluation of Benchmark Datasets for (Deep) Learning-Based Matching Algorithms (No. arXiv:2307.01231). arXiv. https://doi.org/10.48550/arXiv.2307.01231

Peeters, R., Steiner, A., & Bizer, C. (2025). Entity Matching using Large Language Models (Version 1) [Dataset]. OpenProceedings.org. https://doi.org/10.48786/EDBT.2025.42

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Primpeli, A., & Bizer, C. (2020). Profiling Entity Matching Benchmark Tasks. Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 3101–3108. https://doi.org/10.1145/3340531.3412781

Rep. DeFazio, P. A. [D-O.-4. (2020, November 18). H.R.8408 - 116th Congress (2019-2020): Aircraft Certification Reform and Accountability Act (2020-09-29) [Legislation]. https://www.congress.gov/bill/116th-congress/house-bill/8408

Rose, R. L., Puranik, T. G., & Mavris, D. N. (2020). Natural Language Processing Based Method for Clustering and Analysis of Aviation Safety Narratives. Aerospace, 7(143), 143. https://doi.org/10.3390/aerospace7100143

Sachs, A., Perez-Moreno, H., & Compton, N. B. (2024, January 11). Some anxious travelers are avoiding Boeing Max flights. The Washington Post. https://www.washingtonpost.com/travel/2024/01/11/flight-anxiety-boeing-737-max-door-plug/

Shearer, C. (2000). The CRISP-DM model: The new blueprint for data mining. Journal of Data Warehousing, 5(4), 13–22.

Smith, J. B. (2023). Avoiding system failures with event interval probability—737 MAX case study. https://ieeexplore.ieee.org/document/10088220

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Sterkenburg, T. F., & Grünwald, P. D. (2021). The no-free-lunch theorems of supervised learning. Synthese (Dordrecht), 199(3–4), 9979–10015. https://doi.org/10.1007/s11229-021-03233-1

Sun, C., & Shen, D. (2022). Towards deep entity resolution via soft schema matching. Neurocomputing (Amsterdam), 471, 107–117. https://doi.org/10.1016/j.neucom.2021.10.106

Tu, J., Fan, J., Tang, N., Wang, P., Chai, C., Li, G., Fan, R., & Du, X. (2022). Domain Adaptation for Deep Entity Resolution. Proceedings of the 2022 International Conference on Management of Data, 443–457. https://doi.org/10.1145/3514221.3517870

United States Department of Transportation. (2026, January 28). Dynamic Regulatory System. https://drs.faa.gov/browse

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. ukasz, & Polosukhin, I. (2017). Attention is All you Need. Advances in Neural Information Processing Systems, 30. https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Walker, M., & Chokshi, N. (2024, January 6). F.A.A. Orders airlines to ground some Boeing 737 Max 9 jets after midair emergency. The New York Times. https://www.nytimes.com/2024/01/06/business/alaska-airlines-flight-boeing-grounding.html

Wang, T., Chen, X., Lin, H., Chen, X., Han, X., Sun, L., Wang, H., & Zeng, Z. (2025). Match, Compare, or Select? An Investigation of Large Language Models for Entity Matching. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, & S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 96–109). Association for Computational Linguistics. https://aclanthology.org/2025.coling-main.8/

Wang, X., Yang, X., Fu, J., & Qiu, X. (2020). Research on the Construction Technology of Knowledge Graph in Aviation. IOP Conference Series. Materials Science and Engineering, 751(1), 12040. https://doi.org/10.1088/1757-899X/751/1/012040

Wild, G. (2023). Airbus a32x versus Boeing 737 safety occurrences. IEEE Aerospace and Electronic Systems Magazine, 38(8), 4–12. https://doi.org/10.1109/MAES.2023.3276347

Winkler, W. E. (1993). Improved decision rules in the Fellegi–Sunter model of record linkage, *Proceedings of the Section on Survey Research Methods* (pp. 274–279). American Statistical Association.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. https://eric.ed.gov/?id=ED325505

Yang, C., & Huang, C. (2023). Natural language processing (NLP) in aviation safety: Systematic review of research and outlook into the future. Aerospace, 10(7), 600. https://doi.org/10.3390/aerospace10070600

Yu-ping, Z., Jun, S., Yong, C., & Yue, S. (2020). Research on the knowledge ontology model of attribute-oriented airworthiness review compliance activities. 2020 7th International Conference on Dependable Systems and Their Applications (DSA), 453–458. https://doi.org/10.1109/DSA51864.2020.00077

Zeakis, A., Papadakis, G., Skoutas, D., & Koubarakis, M. (2023). Pre-Trained Embeddings for Entity Resolution: An Experimental Analysis. Proceedings of the VLDB Endowment, 16(9), 2225–2238. https://doi.org/10.14778/3598581.3598594

Zhang, D., Li, Z., Wang, X., Tan, K.-L., & Chen, G. (2022). Towards one-size-fits-many: Multi-context attention network for diversity of entity resolution tasks. IEEE Transactions on

Knowledge and Data Engineering, 34(12), 1. https://doi.org/10.1109/TKDE.2021.3060790

Zhao, Q., Li, Q., & Wen, J. (2018). Construction and application research of knowledge graph in aviation risk field. MATEC Web of Conferences, 151, 5003. https://doi.org/10.1051/matecconf/201815105003

Ziv, R., Gronau, I., & Fire, M. (2022). CompanyName2Vec: Company Entity Matching Based on Job Ads. 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA). https://doi.org/10.48550/arxiv.2201.04687

Appendix A As-Executed Workflow Details and Deviations

This Appendix provides the implementation details necessary to duplicate this work, together with the two project GitHub repos. The two GitHub repositories are https://github.com/karnabryan/aircraft_er which contains all data preparation and the baseline feature-based data matcher and https://github.com/karnabryan/ditto_aircraft_er which is forked from megagonlabs/ditto and contains the Ditto deep learning entity resolution model. The forked repository contains no modifications to the Ditto framework, but all aircraft type specific runs and metric calculations.

Mapping Chapter 3 Plan to the As-Executed Workflow and Deviations

Chapter 3 proposed a CRISP-DM-based workflow that emphasized (a) constructing a gold standard dataset from multiple aviation sources, (b) training and evaluating an entity resolution model across multiple hierarchy levels, and (c) demonstrating maturity of the approach through an aviation safety normalization use case. The as-executed workflow depicted in Figure 5 followed this same overall structure, but several elements were modified during implementation to address taxonomy granularity constraints, class construction requirements for supervised learning, and practical limitations encountered when extending the model to additional "dirty" in-domain datasets. The deviations below document where the as-executed workflow differed from the original plan and why those changes were made.

Hierarchy levels and aggregation strategy. The original design proposed evaluating aircraft type matching at four levels of aggregation, including models for master-model and master-series levels. During implementation, the two primary taxonomies did not align at either of these levels of aggregation. The master-model and master-series levels are defined in the

ICAO taxonomy, but not consistently in the FAA taxonomy. While the FAA taxonomy has a "master id" field which initially seemed to represent one of these levels, upon inspection the levels of granularity in this column was inconsistently defined and not well curated. As a result, hierarchy was reduced to two operational levels that could be constructed deterministically and evaluated consistently across sources: make-model-series (MMS) and make-model (MM). This constrained the modeling and evaluation to use only these two levels of hierarchy where consistent ground-truth labels could be established.

Preprocessing expanded to include supervised-learning inputs. While Chapter 3 described data preparation primarily in terms of building curated links across sources, the as-executed workflow required additional preprocessing steps specific to constructing the training files needed for deep learning entity resolution. Each data source and features set combination needed to be represented in Ditto's required COL/VAL string format including the column names and row values as a single string. This string is concatenated for each pair and given a binary match/non-match label, so each model configuration needs its own train/validation/test set. This expanded preprocessing scope reflected the practical requirement that the Ditto deep learning ER model encodes a single text string to represent all features for each record pair rather than using integrated tables with many distinct features.

Negative sampling became an experimental design variable. The original plan anticipated negative example generation, but the as-executed process treated negative sampling as a core experimental factor. Multiple negative sampling strategies were evaluated to understand how "harder" negative examples (e.g., pairs that share make-model or series) affect false positive behavior and overall model performance. This led to systematic experimentation

across parameterizations that varied the number and composition of negative pairs, and to the adoption of larger negative pools to reduce false positives in practice.

Addition of multi-source training using a union model. Chapter 3 focused on developing a model for taxonomy-to-taxonomy matching and then transferring to additional sources. During implementation, the work expanded to include training on a union of multiple source-specific training sets to evaluate the performance of a multi-source model. While this union of multiple source-specific strategy was evaluated at both levels of hierarchy (MMS and MM), the MM level of aggregation included a larger set of training datasets. This step was added to better support the study objective of developing a general-purpose aircraft type matcher.

Evaluation expanded to compare source-specific vs union-trained generalization. The original design focused on the in-domain transfer of additional aviation sources using inference and fine-tuning but did not mention multi-source models. During the execution phase, an additional evaluation pattern was introduced to compare the performance of a source-specific model trained on that specific source pair to the performance of the same source evaluated under the union-trained model. This provided a direct empirical check of whether union training preserved accuracy on individual sources while improving generalization.

Fine-tuning from checkpoints removed due to implementation constraints. The original design emphasized evaluating transfer to new sources using inference as well as model fine-tuning. While Ditto itself is based on fine-tuning to entity resolution on a specific dataset, the envisioned fine-tuning step would have used the Ditto aircraft type model trained on the baseline case as input and fine-tuned for a new in-domain dataset. This fine-tuning was not possible without modifying the Ditto code as there is no fine-tuning included in the existing GitHub Ditto repository. Additionally, training from scratch was feasible and each of the datasets

had enough training data to produce good results independently. For these reasons, the multi-source union model for in-domain transfer was pursued over fine-tuning for in-domain transfer.

Creation of a fully labeled in-domain dataset for one transfer case. To enable at least one quantitative in-domain evaluation beyond spot checking, a fully labeled dataset was constructed for a single "dirty" in-domain source using the BTS dataset. This enabled a controlled comparison between a BTS-trained model and BTS evaluation under the union-trained model, while acknowledging that labeling all in-domain datasets was not feasible within scope. The BTS dataset was selected because there were far fewer entries than some of the other in-domain evaluation datasets. The BTS dataset required additional labeling of 252 unique aircraft type records compared to 2,345 unique aircraft type for AIDS and 12,827 for NTSB.

Manual make normalization and blocking introduced for transfer evaluation feasibility. Although blocking was explicitly out of scope in Chapter 3, transfer evaluation to large in-domain datasets required a feasibility adjustment to avoid exhaustive Cartesian pairing. A simplified make-based blocking strategy was introduced to constrain candidate generation for evaluation. The make was manually mapped to the ICAO formatting for that make and each row in the evaluation dataset was compared only with all candidate pairs sharing the same make. This blocking strategy was used for evaluation purposes only and was treated as a pragmatic implementation constraint rather than a blocking contribution.

Use case framing adjusted to match scope. The original language of "Demonstrate Aviation Safety Use Case" was revised to "Explore Aviation Safety Use Case" to reflect the scope of the executed work. The as-executed study supported the use case through model-based alignment and qualitative inspection of match outputs but did not implement a full end-to-end operational normalization pipeline across all sources or attempt to calculate risk based on the

aligned tables. A full end-to-end use case demonstration would require an additional automated step to provide blocking.

Clarification of evaluation and hypothesis testing scope. Chapter 3 proposed a parametric comparison of means using a *t*-test to evaluate model performance between the baseline $RecordLinkage$ model and the Ditto model results. However, in Chapter 4, the evaluation framework is refined to better align with the data properties of the FAA-to-CICTT taxonomy-to-taxonomy task. While precision, recall, and F1 are reported across all configurations to provide descriptive performance, the formal hypothesis test is now centered on the errors of the top-1 best match per CICTT row using McNemar's test (McNemar, 1947).

This methodological shift is motivated by the fact that both models are evaluated on paired observations rather than independent observations and the fact that the Ditto model involves stochastic optimization through random initialization and batch shuffling. This means that the F1 score is a representative point estimate from a single training run rather than the mean of a set of runs from the stochastic model. To test the overall pairwise performance of the two top-1 models, McNemar's test utilizes a two-by-two contingency table to analyze where the models agree and disagree as proposed by Dietterich (1998). Python's statsmodels library was used incorporating the Edwards' standard continuity correction (Edwards, 1948).

Data Preprocessing

This section describes the preprocessing steps executed to prepare the study datasets for baseline and deep-learning entity resolution. It documents the business and data understanding analyses that shaped feature and hierarchy decisions and details the construction of the FAA-to-CICTT gold dataset, the $RecordLinkage$ baseline matcher inputs, and the Ditto training and evaluation files. These steps produce a consistent set of MMS- and MM-level datasets, including

source-specific and union configurations, as well as evaluation-only in-domain datasets used to assess transfer learning on the union configuration.

Business Understanding (As Executed)

Aviation safety analysis often requires combining records from multiple public datasets (e.g., operational activity, aircraft registry information, incident/accident narratives, and taxonomy references). In practice, these sources rarely share a universal aircraft type identifier, and aircraft type is represented using different conventions depending on the system's purpose. The central business problem in this study is therefore not simply "matching strings," but enabling cross-source analysis by identifying which sources can be linked deterministically and which require entity resolution to reconcile aircraft type specifications.

The business understanding in this study focused on (a) identifying which aircraft type fields are available across public aviation datasets and (b) determining whether any shared keys exist that allow deterministic linkage. Table 4 summarizes how aircraft type is represented across the study's data sources and whether each source provides standardized linkage keys. While all sources contain some representation of aircraft type, they vary substantially in structure and granularity. Some sources provide discrete make-model-series (MMS) fields that support direct, field-level comparison (e.g., CICTT, FAA, NTSB). Other sources collapse MMS information into fewer fields (e.g., ICAO, AIDS, OPER, REG), often including series within the model string. Finally, several sources represent aircraft type primarily as free text (e.g., NWSD, BTS) where all aircraft type information is described in a single textual string.

Two attributes emerged as deterministic linkage keys across the data sources, the ICAO type designator and the registry code. The ICAO type designator provides a direct join key between the CICTT and ICAO reference taxonomies. The registry code supports deterministic

linkage among FAA, OPER, and REG. Together, these keys create isolated linkage clusters. Within these clusters, data can be directly joined via direct joins. Beyond these silos, the primary cross-source challenge remains: there is no shared identifier that connects ICAO-based representations (CICTT/ICAO) to the registry-linked FAA representations (FAA/OPER/REG) in a way that extends to additional accident and utilization datasets.

The study's gold dataset design is validated by the Table 4 structure. Because an effective method for linking FAA taxonomy records to CICTT taxonomy records would bridge the gap between the taxonomy island (CICTT/ICAO) and the registry island (FAA/OPER/REG), the study prioritizes manual alignment between FAA and CICTT where no shared key exists. This manually curated mapping provides the ground truth required to train and evaluate entity resolution approaches under controlled conditions. The remaining datasets (AIDS, NTSB, NWSD, BTS) then serve as in-domain transfer targets, supporting later evaluation of whether a learned matcher generalizes to "dirty" in-domain inputs.

Table 4 frames the practical constraints that shaped the modeling decisions reported in this chapter. Deterministic linkage enables limited integration within subsets of the aviation data ecosystem, but broader cross-source normalization requires entity resolution methods capable of matching heterogeneous aircraft type representations across structured, partially structured, and free-text formats where not all sources use the same fields or granularity.

Data Understanding (As Executed)

Following the business understanding phase, the study conducted a deeper dive into actual content of the columns represented in each of the study data sets to gain an understanding of the level of standardization in these fields. This provides insight into (a) which specific fields are useful to encode aircraft type (e.g., make, model, series, type certificate, ICAO designator,

and descriptive names), (b) the degree to which those fields are standardized versus free-form, and (c) whether sources align at a common level of granularity. These observations guided which fields were feasible as model inputs for deep entity resolution and whether matching should be evaluated at the make-model-series (MMS) level or reduced to make-model (MM) for the various data sources.

Table 5 shows variations in aircraft type fields for a single make model across nine aviation data sources. These highlight variations between Bombardier and Airbus Canada for the make field, along with abbreviations, misspellings, and other variations seen in the data. The model field is most often coded as BD500 but is sometimes also encoded as A220 or CS100 or CS300. The two series values 1A10 and 1A11 are contained in some of the datasets and not others representing a difference in the reported level of granularity. Information from the model and series fields are sometimes repeated across the columns. Finally, the four-letter ICAO designators BCS1 and BCS3 show little resemblance to the other fields, where some designator types are easier to guess (e.g. B38M represents a Boeing 737 Max 8). Those sources with a single free-text entry field for aircraft type show the greatest inconsistency. Matching records across sources requires context beyond variations in strings, including an understanding that an BD500, A200, and CS100 can all mean the same thing.

Table 5 presents a representative snapshot of aircraft type strings for the BD500 make-model family across nine aviation data sources. Values are shown as recorded in each system and illustrate that a single aircraft family may be represented as a structured MMS fields (make, model, series), partially structured strings in which series is embedded within a combined model value (e.g., “BD-500-1A11”), or a single free-text aircraft type description. The table also demonstrates cross-source naming variation driven by both organizational conventions and

historical changes. For example, type certificate ownership transitioned from Bombardier to Airbus Canada, so manufacturer fields may reflect either entity depending on source and time period. In addition, the aircraft family may be referenced by certification-oriented naming (e.g., “BD-500”) or by marketing/common naming (e.g., “A220”), which may appear without the corresponding certification identifier in some datasets.

At a high level, the BD500 family can be summarized into two dominant variants: BD-500-1A10 (popular name A220-100; ICAO designator BCS1) and BD-500-1A11 (popular name A220-300; ICAO designator BCS3). Table 5 shows that the level to which MMS entries are not expressed consistently across sources and cannot be identified through simple deterministic joins or string similarity alone, particularly when records contain only a marketing name (e.g., “A220”) or only a coded type designation string (e.g. BSC1). This motivates the need to evaluate ER approaches that can learn associations across heterogeneous aircraft type representations.

Within the two reference taxonomies, aircraft type is represented using different conventions. CICTT records BD500 as structured MMS values (e.g., model “BD500” with series “1A10” or “1A11”) and may associate multiple popular names with the same make-model family (e.g., CS100, A220-100, ACJ TWO TWENTY). In contrast, ICAO expresses the same family primarily through the standardized ICAO type designator (e.g., BCS1, BCS3) alongside a descriptive model string (e.g., “BD-500 CSeries CS100”), which collapses model and descriptive naming into a single field. These differences indicate that even “clean” reference sources do not follow a single shared schema for aircraft type, which affects what can be matched deterministically versus what requires ER.

The FAA taxonomy provides relatively consistent abbreviations and structured fields, but also uses MMS-style codes (e.g., “BD-500-1A10”) that embed series information into a single,

combined string which must be parsed for field-level matching. The registry and operator sources align well with the FAA taxonomy, although the same information is stored in the "model" field in the registry and the "series" field in the operator tables. While the CICTT taxonomy does not tend to repeat information between the make-model-series columns, these sources do repeat the information from "model" in the "series" field.

The remaining datasets demonstrate even greater variety. While the AIDS series column aligns well with the other FAA sources, the make column demonstrates duplicate rows due to inconsistent make specifications and spelling errors (the series "BD-500-1A10" is repeated three times with makes AIRBUS CANADA LP, BOMBARDIER, BOMBARIER). Additionally, the AIDS source does have a series column which is sometimes populated and sometimes null. This means that the series information is sometimes collected in the series column and sometimes rolled up into the model column. NTSB provides structured make and model that is well-defined for this example but has many repeated rows due to inconsistent representations in other examples. Aircraft type in NWSD and BTS is expressed as a single unconstrained textual description, namely "A-220" and "Bombardier C Series Cs100." The goal for data matching is to recognize "A-220" and "Bombardier C Series Cs100" as the same aircraft type, which clearly requires the algorithm to have significant context about the heterogeneous representations of aircraft type.

This data understanding analysis step informed subsequent modeling choices in several ways. First, it showed that MMS-level matching was only feasible for sources that provide make, model, and series as discrete fields or as consistently structured MMS-like strings (e.g., "BD-500-1A11"). Otherwise, comparable MMS features cannot be constructed reliably across sources. For heterogeneous sources with collapsed model-series fields or free-text descriptions

MM-level matching was more appropriate because not all datasets provide a series attribute in a consistent or extractable form. Second, the BD500 example illustrates recurring sources of variation that create challenges for data matching, including use of abbreviations, alternate manufacturer naming, embedded series conventions, and multiple valid descriptive names for the same aircraft family. Finally, several sources contain duplicate or near-duplicate aircraft type rows. Standard ER practice treats duplicate detection as a distinct preprocessing step so that downstream matching could operate on de-duplicated representations. Deep ER approaches may not require this constraint.

Data Preparation: Create Ground Truth Datasets (As Executed)

The primary objective of the ground-truth construction step was to create a labeled dataset aligning an FAA aircraft type reference representation to the CICTT taxonomy in the absence of shared keys. This "gold" dataset provides supervised labels for model development and evaluation and serves as the bridge between the two deterministic linkage groups identified in the business understanding phase (ICAO-designator-linked taxonomies and registry-code-linked operational tables).

FAA reference source substitution. Chapter 3 reported 8,027 aircraft type rows in the FAA taxonomy reference. During execution, that specific version of the FAA aircraft reference file was no longer available for download from its original location. To preserve the planned FAA taxonomy structure, the study substituted an available equivalent file, the FAA aircraft reference distributed with the AIDS dataset. This substituted reference file retained the same schema and primary key structure as the originally planned FAA reference but contained fewer rows (5,118). All FAA-taxonomy preprocessing and labeling steps were therefore implemented using the 5,118-row FAA reference file available at the time of analysis.

Consolidating FAA reference, operations, and registry into FAA-aligned table.

Before creating a ground-truth mapping between the "FAA taxonomy" and the CICTT taxonomy, preprocessing was carried out to create a joined version of all FAA-adjacent data sources (FAA, OPER, and REG in Table 4). This produced a single FAA-aligned dataset for labeling and focused the analysis on the FAA aircraft types that could be validated against operator and registry information. The consolidated file used for gold dataset construction was created as follows.

First, the FAA aircraft type reference table containing 5,118 rows joined with an aggregated operator-aircraft table (OPER in Table 4). The OPER summary table was created by joining an operator file with an operator-aircraft file and aggregating by the aircraft make-model-series code (acft_mms_code). Aggregations included the count of distinct operators of that aircraft type and the count of all aircraft of that aircraft type This summary was then joined to the aircraft type reference using acft_mms_code, providing additional numeric columns for the number of operators with aircraft of that type and the number of aircraft currently assigned for commercial operations of that type. These utilization measures provided additional context during labeling and disambiguation and linked to one of the utilization datasets to the gold dataset.

Next, this was joined with the FAA Registry data (REG in Table 4). The registry data is stored in flat files and includes a file for aircraft type (acft_ref), a file for actively registered aircraft (master), and a file for deregistered aircraft (dereg). The reg_code field was used as a join key to connect registry data to the FAA aircraft type reference table. In doing so, 475 records had a null registry code and, upon inspection, were dropped because they appeared to be older, less prominent aircraft types. The remaining aircraft types were then joined with both the

master and dereg files to identify aircraft types that did not appear in either file. This removed an additional 884 aircraft types. The resulting dataset contained 3,759 aircraft types and included counts of registered and deregistered aircraft. This filtering reduced labeling effort and ensured the gold dataset reflected aircraft types that occur in the FAA registry population rather than unused reference entries.

Manual FAA-to-CICTT mapping process. The consolidated FAA-aligned dataset was then manually mapped to the CICTT taxonomy in Excel using a dedicated manual_mapping column. The target label stored in this column is the CICTT aircraft identifier, recorded as an integer value.

Mapping was performed primarily using the FAA aircraft reference fields (make, model, and series values), with additional context used when needed to disambiguate aircraft families. In practice, the most useful secondary fields were the type certificate number and the popular name. During labeling, the CAST/ICAO Taxonomy Team Aircraft Analysis Tool was used to validate candidate matches and accelerate lookups (ICAO, 2024). After registry-based filtering, 3,678 of the 3,759 FAA-aligned aircraft type rows were assigned a CICTT identifier with a non-null manual_mapping entry. The remaining 81 rows were excluded from MMS-level ground truth because the available description was ambiguous or represented only make-model information rather than a consistent make-model-series specification.

Since the FAA taxonomy was used as a basis for the labeling, there were no duplicate acft_mms_code values, while there are several duplicate FAA rows mapping to the same CICTT identifier. For example, “AS-350-B2,” “AS-350B2-350B2”, and "AS-350D-350B2” were all mapped to the same CITT identifier, reflecting alternative encodings of the same aircraft family and inconsistent placement of series information across model and series fields. Although

acft_mms_code is unique in the FAA reference table as a textual field, the rows are not strictly unique at the entity level. In total, 451 FAA rows participated in many-to-one mappings of this kind.

The final output of this step is a manually labeled FAA-to-CICTT gold dataset with 3,678 FAA aircraft type rows assigned to CICTT identifiers (positive labels), derived from a filtered FAA-aligned input table of 3,759 aircraft type rows. Ground truth served three purposes. First, it provided the labeled positive pairs required to train and evaluate both the baseline rules-based matcher and the deep learning matcher (Ditto) for the primary taxonomy-to-taxonomy task. Second, it supported comparisons across hierarchy levels (make-model-series and make-model) by anchoring evaluation to consistent labels. Third, it provided a foundation for constructing additional source-pair datasets (e.g., taxonomy-to-registry) to test whether the learned matcher could generalize across heterogeneous input formats.

Data Preparation: Create Baseline Data Matcher (As Executed)

To establish a transparent comparison point for the deep learning matcher, a deterministic baseline data matching model was implemented using the $RecordLinkage$ library. The baseline matcher used a traditional entity resolution approach based on equivalent column comparisons between the FAA and CICTT taxonomies. For each candidate pair, the model calculated a series of column match scores and summed these into a total score. This single additive score was then used with a fixed threshold *t* to produce a record match decision which is a common approach for rule-based ER systems. Gold labels were then used to calculate pairwise precision, recall, and F1 scores.

Input datasets and reference tables. The baseline matcher utilized the manually labeled taxonomy-to-taxonomy gold dataset. From this file, two reference tables were constructed, a

CICTT table and an FAA table, containing only make, model, series, type certificate, and popular name columns for each dataset. Because the CICTT identifier may appear in multiple FAA rows (there is a many-to-one mapping), the CICTT table was de-duplicated by grouping on the CICTT identifier and its descriptive fields before matching. The FAA dataset contained a total of 3,678 rows while the CICTT table contained 3,456 unique rows.

Candidate generation (no blocking). Candidate generation was performed with full indexing, comparing every CICTT reference record to every FAA aircraft type record. This ensured that baseline performance reflected only the scoring rules (not a blocking strategy), at the cost of creating a very large candidate-pair universe. In the executed run, the full candidate set contained 12,711,168 candidate pairs.

Standardization and cleaning of matching fields. Prior to scoring, all matching fields were standardized using a consistent cleaning function. The cleaning procedure lowercased text, removed non-alphanumeric characters (excluding spaces and digits), normalized separators (hyphens/underscores to whitespace), and removed bracketed content. This step followed standard entity-resolution cleaning practice to reduce superficial string variation in punctuation, formatting, and capitalization while preserving the underlying aircraft type information.

Feature construction and scoring rules. For each candidate pair, binary similarity indicators were computed across five attribute families: make, model, series, type certificate, and popular name. Each of these attributes used two binary indicators. The first was computed using the Jaro-Winkler similarity metric with a threshold of 0.85 and the second captured exact string equality between the two strings. Thus, an exact match on a field contributed two points (Jaro-Winkler and exact), whereas near matches contributed only a single point (Jaro-Winkler only).

The ten binary similarity features were then summed into a single total score with a maximum value of 10, where higher scores indicate greater agreement across the five fields.

Pairwise linkage classification (thresholding). After calculating the ten binary similarity indicators for each candidate FAA to CICTT pair, the baseline matcher produced a single additive total_score ranging from 0 to 10. Pairwise linkage classification was then performed by applying a fixed threshold *t* to this score: all candidate pairs with a total score greater than or equal to *t* were classified as matches, and all remaining pairs were classified as non-matches. For each threshold value, the predicted matches were evaluated against the manually labeled gold links (3,678 labeled positive pairs) using pairwise precision, recall, and F1 over the full exhaustive candidate universe of 12,711,168 candidate pairs.

Top-1 assignment (best-match per record). In addition to threshold-based pairwise classification, a top-1 assignment strategy was evaluated to reflect downstream settings where each FAA aircraft type record must link to at most one CICTT entity. This approach more accurately represents the goal of assigning a matching aircraft type from one data source to that used by another data source. For each FAA MMS row, all FAA-CICTT candidate pairs were scored, and the single highest-scoring candidate was selected as the predicted link. No link was produced when the maximum available score was zero. When multiple candidates tied for the maximum score, ties were broken deterministically using exact type certificate agreement, exact model agreement, exact series agreement, exact make agreement, and finally the smallest CICTT identifier to ensure stable selection.

Data Preparation: Create Ditto Input Files (As Executed)

To train a deep learning entity resolution model using Ditto, the manually labeled FAA to CICTT gold dataset was transformed into Ditto's required pairwise text format. In Ditto, each

training example is a pair of records represented as two serialized strings (a left record and right record) plus a binary label, rather than a structured feature table. The executed preprocessing step therefore focused on serializing each CICTT and FAA record into Ditto's COL/VAL representation, generating negative (non-match) examples at controlled ratios, and producing train/validation/test text files in the exact format required by the Ditto training pipeline. All datasets contained a 80/10/10 split and several output files: train.txt, valid.txt, test.txt, all_pairs.txt, train_with_id.txt, valid_with_id.txt, test_with_id.txt, and all_pairs_with_id.txt. The id-preserving files simplified downstream error analysis. The train.txt, valid.txt, test.txt, and all_pairs.txt files have no header row and are tab delimited in the format that Ditto requires. The files ending in "_with_id.txt" are comma delimited and have header row to support downstream analysis.

Baseline Taxonomy-to-Taxonomy. The Ditto input files were created from the cleaned matched gold file (baseline_cleaned_matched_data.csv). This file contains one row per FAA record with its mapped CICTT identifier. The CICTT columns were selected as the left records and the FAA as the right records because CICTT was always selected as the left dataset in all Ditto model runs. For both the CICTT and FAA records, the model inputs were limited to five aircraft type attribute families: make, model, series, popular name, and type certificate. For simplicity, these columns were renamed to "make," "model," "series," "name," and "cert" for both datasets. These columns were selected because they collectively encode make-model-series specifications while also providing additional disambiguating context for aircraft families (e.g., popular names and certification identifiers).

Each record was serialized into a single text string using Ditto's COL/VAL convention: each non-empty field becomes a token sequence of the form COL <field> VAL <value>,

concatenated into one string per record. For example, the BD500-1A10 CICTT record in Table 5 is serialized as “COL make VAL BOMBARDIER COL model VAL BD500 COL series VAL 1A10 COL name VAL CS100 COL cert VAL A-236.” The two CICTT and FAA serializations are concatenated into a single string followed by a binary match (1) or non-match (0) label.

Positive training pairs were created directly from the gold dataset by treating each row as a labeled match (label = 1). In the baseline configuration, negative pairs were generated using a 10:1 ratio by sampling 10 random negative pairs for each positive pair. The combined set of positive and negative pairs was then shuffled and split into 80% training, 10% validation, and 10% test partitions using the scikit-learn Python library. Each row is a tab-separated triplet consisting of the left entity string, the right entity string, and the binary label.

For each record, the negatively sampled pair is drawn from a pool of 3,677 non-matching records, i.e., rows where the manual_mapping CICTT id was not equal to the record CICTT id. These form pool for non-matched pairs, with many of the negatively sampled record pairs sharing little overlapping information. To address this, a strategy referred to as learning harder in this manuscript was explored. The learning harder strategy attempts to create negative pairs that share some common fields and are therefore harder to differentiate, providing the model with the opportunity to learn subtle distinctions rather than only easy cases. For these learn-harder negative training examples, in addition to randomly selected negatives, a subset of negatives was selected from the same make and model (MM), thus differing only in series. A third subset of negative examples used pairs with a matching series. This means that the series field matched exactly, but the make and model did not match. These were created because an inspection of the $RecordLinkage$ model rows with low scores showed several records matching on series only. These three sampling strategies were implemented by varying the size of the negative pool for

random pairs, same MM, and same series, denoted, e.g., (1, 8, 1). The learn harder strategy applied to the entire set of pairs (train, validation, and test files). The baselines generated were:

- Baseline LH 1:3 (baseline_lh_0), ratio random, same MM, same series: (0,3,0)
- Baseline LH 1:10 (baseline_lh_1): ratio (1,8,1)
- Baseline LH 1:50 (baseline_lh_2): ratio (20,20,10)
- Baseline LH 1:90 (baseline_lh_3): ratio (50,20,10)

In this way, for the CICTT-to-FAA taxonomy-to-taxonomy runs, five different train/validate/test sets were produced: baseline, baseline_lh_0, baseline_lh_1, baseline_lh_2, and baseline_lh_3. The results section includes analysis of the five different baseline options. This set of baseline files was only created for the CICTT-to-FAA taxonomy-to-taxonomy case. All other model runs used a standard 1:10 ratio of randomly selected negative pairs.

Baseline Evaluation Datasets. The $RecordLinkage$ baseline model used exhaustive comparison over all 12,711,168 candidate pairs. In contrast, evaluating the Ditto model on the full exhaustive candidate set is computationally expensive. Three evaluation-only datasets were therefore created to support comparisons between the two models and to understand how baseline negative-pair generation impacts the rate of false positives. These three sets are exhaustive over a small set of positive candidate pairs, meaning that each positive match in the set is compared against all remaining other rows in the match dataset (the non-matching rows). For each selected positive CICTT-FAA match, the left record was fixed and paired with every FAA record whose manual_mapping did not equal that CICTT id. The first evaluation set is based on the nine records with make “Canadair.” The second is based on 10 randomly selected matches, and the third is based on 100 randomly selected matches. These evaluation-only datasets are referred to as the Eval Canadair, Eval Random (10), and Eval Random (100)

datasets. These eval sets are not separated into train, validation, and test sets, but are provided as Ditto-formatted labeled pairs that can be used directly with the Ditto matcher.

Additional Make-Model-Series (MMS) Datasets. The baseline model is evaluated with matches at the MMS level so that each record must match all three make, model, and series to be a match. At the MMS level, three additional Ditto input datasets were created, CICTT-to-Registry, FAA-to-Registry, and a union configuration. Because the Registry code was stored as a key in the FAA base table, the Registry manufacturer and model fields are part of the labeled gold dataset mapping because of the common registry code join key described in Table 4. The Registry data only has these two fields, manufacturer and model, rather than the five selected fields in the taxonomy-to-taxonomy baseline model. An advantage of the Ditto framework is that the generic COL/VAL format does not constrain the records to a pre-defined set of fields. This means that the serialization for Registry entries can include only these two columns. A dataset was created for both CICTT-to-Registry and FAA-to-Registry record pairs.

The union dataset is a dataset at the MMS level of granularity that merges all records from the baseline dataset with those of the CICTT-to-Registry dataset. It is noteworthy that this union dataset leaves the CICTT columns as the left dataset but that the right columns vary both the dataset and the columns utilized in the model. To allow for the evaluation of the union model against each individual source dataset, the train/validate/test files were merged separately and before randomizing the row order. In this way, a record in the train file in the CICTT-to-Registry dataset will also be in the train file of the union dataset. This split-preserving construction prevents train/test leakage. The CICTT-to-Registry and FAA-to-Registry, and union datasets all use a standard 1:10 ratio of randomly selected negative pairs.

Make-Model Datasets. In addition to the MMS baseline task, the process flow for this study included modeling also at the make-model (MM) level. The MM level matches if two records share the same make and model but does not require a series-level match. This level of aggregation is useful for evaluating matching when series information is unavailable, inconsistent, or not reliably extractable across sources. From an analytic point of view, aircraft data may also be aggregated at the MM level for trend analysis. Under the MM aggregation strategy, the left-hand entity represents a CITT AGG, the CICTT aggregated make-model entity defined as the combination of CICTT make and CICTT model. These were created by collapsing the MMS gold mappings to the MM-level. Because a single make-model may correspond to multiple variants at the row level (e.g., multiple series values), MM-level labeling yields a one-to-many relationship between a CICTT AGG entity and row-level entries in other sources.

Across all MM datasets, records were serialized using Ditto's COL/VAL format. The CICTT AGG left record used four of the fields from the MMS baseline datasets—make, model, popular name, and type certificate—serialized as "make," "model," "name," and "cert." The right-hand record schema varied by source depending on available fields, demonstrating Ditto's flexibility to ingest heterogeneous feature sets without requiring a fixed schema. For all MM datasets, the final selected runs used no learn-harder configurations and instead applied a standard 1:10 ratio of randomly sampled non-match pairs.

The first MM dataset matched CICTT make-model entities to row-level CICTT make-model-series entries. This dataset is expected to achieve very high matching performance, as the matcher only needs to ignore series-level information, even when series is included in the right-hand string as extraneous content. The right-hand columns consist of the five standard fields, make, model, series, name, and cert. Multiple CICTT row-level entries may exist for a single

make-model, so this dataset may contain multiple labeled match rows for the same CICTT AGG entity.

The second dataset matched the CICTT AGG entity to FAA taxonomy rows. These FAA rows also include make, model, series, name, and cert, and there may be more than one match per CICTT AGG because multiple series rows can map to the same aggregated make-model entity. The MM Registry dataset contains only two columns, manufacturer and model. Again, a single CICTT AGG entity may have more than one Registry match.

The remaining MM datasets were based on the ICAO Aircraft Type Designator DOC8643 reference taxonomy. DOC8643 is an ICAO reference source organized around a four-letter code for an aggregated aircraft type and a single description field (see Table 5). There may be more than one description for a single four-letter code which appears as multiple rows for a single code. Using this source, several datasets were prepared. The DOC8643 dataset includes both the code and the description as two COL/VAL entries. The DOC8643_description dataset contains only the description field as a single COL/VAL entry. The DOC8643_code dataset contains only the four-letter code field. A final dataset, DOC8643_with_drops, contains a random mixture of rows with both description and code fields, description-only rows, and code-only rows.

A final union dataset merged records from the CICTT, FAA, Registry, DOC8643, DOC8643_description, and DOC8643_code datasets. Similar to the MMS union model, the model trained on the union data can be evaluated against the individual source datasets to assess whether training across multiple sources reduces performance relative to a source-specific model.

Evaluation-only Datasets (In-Domain Transfer). The in-domain aviation datasets used for transfer evaluation are summarized in Table 4 and include three event/occurrence sources, AIDS, NTSB, and NWSD, and one utilization source, BTS. These datasets do not share ground truth labels, except for BTS, which was manually labeled to support quantitative evaluation. Because the aircraft type in these sources may not include fully granular data (e.g. include series information), all evaluation-only runs were prepared for the MM-level. The make model union model is comprised of many different sources that specify aircraft type in a variety of ways, making it compelling for in-domain transfer use.

While blocking is beyond the scope of this study, exhaustive pairwise comparison on the larger in-domain datasets was not computationally feasible. To reduce the candidate space, a standardized CICTT make was manually assigned to each record so that each row was compared only against other MM types within the same make. This process was not automated but performed manually. For dirty datasets with multiple ways of expressing the same aircraft type, "tail" records with non-standard make specifications may not have been assigned a standardized make. As a result, some records in these tails were not included in the pairwise comparisons.

The AIDS data source contained two fields, make and model, which were encoded in the COL/VAL format as two entries. There were 2,345 unique MM values, which resulted in 47,188 total evaluation pairs when candidate pairs were constrained to share the same make. The NTSB source contained three fields, make, model, and series, and included 12,837 distinct MMS representations. Many of these were repeated values with slight variations, and the series field was often unpopulated, or sometimes embedded within the model field. Constraining candidate generation by make resulted in 229,998 comparison pairs. The NWSD dataset contained a single field, "aircraft," which is a non-standardized descriptor. There were 569 distinct values, and applying the make constraint produced 16,755 comparison pairs. Because these three datasets are unlabeled, they were prepared only as evaluation sets, and each was written as a single all_pairs.txt file rather than separate train/validation/test partitions.

The BTS utilization dataset contained a single field, description, with 328 unique values. Because this dataset had a manageable number of rows, it was manually labeled so that train/validation/test datasets could also be created for this source.

The preprocessing steps described above produced a set of Ditto-formatted datasets that support modeling at two levels of aggregation and across multiple right-hand source schemas, including structured taxonomies, partially structured registry tables, and code/description reference representations. In addition, evaluation-only in-domain datasets were prepared to support transfer assessment under practical candidate-generation constraints. The next section reports the modeling runs executed using these prepared inputs, including the baseline taxonomy-to-taxonomy Ditto configurations, the MM source-pair and union models, and transfer evaluation on in-domain aviation datasets.

Modeling

This section provides additional details for the as-executed modeling for both deep learning modeling results and in-domain transfer learning.

Modeling: Deep Learning Model Results (As Executed)

The MMS level included experimental runs for baseline taxonomy-to-taxonomy matching and inference-only tasks to examine the impact of negative pair selection strategy. It also looked at schema variation and multi-source models. The MM level further experimented with schema variation and multi-source models.

[RQ2] Baseline taxonomy-to-taxonomy. Baseline taxonomy-to-taxonomy runs were trained using Ditto input files created from the labeled CICTT-to-FAA gold dataset. Five versions of the CICTT-to-FAA training data were used to study the effect of negative sampling on false positives. The baseline configuration used a 1:10 ratio of positive pairs to randomly generated negative pairs. Four additional train, validation, and test sets implemented a learn-harder strategy in which a portion of negative examples were sampled to be more confusable with positives. These variations changed both the overall positive-to-negative ratio and the

composition of negatives. Each dataset configuration produced a separate fine-tuned model checkpoint. A batch process was used to train each model and then run inference on the held-out test partition for that dataset configuration.

In addition to evaluation on the held-out test partition, evaluation-only datasets were used to study the role of negative pairs on both precision and recall. These evaluation-only datasets contain a small set of true matches with the exhaustive set of negative candidate pairs. These datasets are referred to as Eval Canadair, Eval Random (10), and Eval Random (100). For each of these datasets, a small set of known matched rows was paired against the full set of non-matching candidates in the evaluation pool to create a highly imbalanced evaluation setting. All five CICTT-to-FAA baseline models were applied to these evaluation-only candidate sets to examine the behavior of each negative sampling configuration under conditions that provide an exhaustive linkage scenario for a small subset of positive pairs. A full exhaustive candidate set was also evaluated for direct comparability with the traditional $RecordLinkage$ baseline under exhaustive candidate generation.

[RQ2] Schema variation (MMS-level). Beyond the CICTT-to-FAA task, additional MMS source-pair models were trained to evaluate performance when the right-hand record schema differs from the full taxonomy feature set. Two models were trained for CICTT-to-Registry and FAA-to-Registry matching. In these configurations, the right-hand record serialization reflected the Registry schema and contained only manufacturer and model fields. These runs were executed to assess whether Ditto could learn robust matching behavior when the right-hand source provides fewer structured attributes and a different field layout than the baseline taxonomy-to-taxonomy task.

[RQ2] Merged data set Union model (MMS-level). A union MMS configuration was trained by combining the CICTT-to-FAA dataset with the CICTT-to-Registry dataset. The union construction preserved split boundaries by merging training files with training files, validation files with validation files, and test files with test files prior to shuffling. This design supported multi-source training without introducing split leakage across the datasets. The union model was then evaluated on the union test partition. Inference was also performed on the original source test partitions to assess whether exposure to multiple right-hand schemas preserved performance relative to the source-specific models.

[RQ2] Make Model Hierarchy. To evaluate matching at a coarser level of aggregation, a set of Make-model datasets was constructed using aggregated CICTT make-model entities as the left-hand representation and multiple right-hand sources. The make-model level of aggregation was used with five of the datasets in the 9 total study datasets referenced in Table 4. A CICTT MM to CICTT MMS dataset was included as a reference configuration in which multiple MMS rows could map to a single MM entity. Models were also created at the MM level for the FAA taxonomy reflecting the fields available in those sources at this aggregation level.

The ICAO DOC8643 reference taxonomy was used for several MM runs that varied which DOC8643 attributes were provided to the model. Separate datasets were constructed using both the ICAO type designator code and description, the code only, and the description only. A fourth dataset included a randomized mixture of both columns, code only, and description only to test the model performance using partially observed schemas.

[RQ2] Make Model Union Model. As with the MMS-level, these train, validation, and test files were merged within split and then shuffled to train a Make Model union model. The Make Model Union model was intended to support inference across multiple right-hand formats

using a single model checkpoint. To assess whether union training preserved source-specific behavior, the union model was used for inference on each original single-source MM dataset.

Modeling: In-Domain Transfer Learning (As Executed) (RQ3)

To explore in-domain transfer beyond curated taxonomy and registry sources, Ditto evaluation datasets were constructed for four additional aviation sources: AIDS, NTSB, NWSD, and BTS. These sources represent more heterogeneous aircraft type specifications, including partially structured fields and unconstrained free-text descriptions. Because full ground truth labeling was not feasible for most in-domain sources within scope, transfer evaluation relied on inference with structured review of a small set of predicted links and simple qualitative review.

As explained in the Data Preparation section, candidate generation for in-domain sources was not exhaustive. To make evaluation feasible, records in each in-domain dataset were manually assigned a standardized make, and candidate pairs were generated only within the same make. This make-constrained candidate generation significantly reduced the candidate space from an exhaustive scenario so that inference on all records was feasible.

For these in-domain inference runs, outputs were summarized using a top-1 linkage selection strategy. For each left-hand record, candidate pairs were ranked by model confidence, and the single highest-confidence candidate was selected as the proposed link. If the top-ranked candidate was predicted as a non-match, no link was created for that left-hand record. This top-1 linkage strategy was used as a practical way to produce one proposed alignment per left record for downstream inspection.

For the unlabeled in-domain AIDS, NTSB, and NWSD datasets, model outputs were first assessed through qualitative inspection of the selected top-1 links. In addition, a random sample of 100 links for each dataset was manually reviewed and labeled as match or non-match to

provide a basic quality check on the proportion of correctly matched inputs. This manual audit was used to summarize the apparent correctness of the top-1 links in the absence of full ground truth labels.

The main quantitative results for in-domain transfer were based on the BTS source because the source contained a manageable number of unique aircraft type descriptions and could be feasibly manually labeled. Two complementary BTS evaluations were conducted. First, inference was performed under transfer conditions using models trained on other sources and evaluated under both exhaustive and make-constrained candidate generation settings. Second, a BTS-specific fine-tuned model was trained using the labeled BTS train, validation, and test partitions and evaluated on the BTS test split. This enabled direct comparison between a dedicated in-domain model and a transfer setting where the model had not been trained on the BTS representation.

Appendix B Institutional Research Board Approval Letter

IRB-FY24-25-12 - Initial: Exempt from Further Review Inbox ×

do-not-reply@cayuse.com Mon, Jul 15, 2024, 2:17 PM

to k.bryan6020, lfulton

National UNIVERSITY

9388 Lightwave Ave.
San Diego, CA 92123
irb@nu.edu

Notice of Exemption

July 15, 2024

To: Karna Bryan

Project Title: Enhancing Aircraft Type Matching Across Data Sources Using Entity Resolution
NU IRB Number: IRB-FY24-25-12
Determination: Exempt from further review 45 CFR 46.101

Status: Active - Research activities may begin as of July 15, 2024

Dear Karna Bryan:

The study referenced above has been reviewed by the National University IRB. The IRB has determined your research is exempt from further review under 45 CFR 46.104, which means you will not need to renew your study and may begin your study effective immediately. However, if you find the need to change your study in any way, you will need to submit a modification to the IRB prior to implementing the changes. This will allow the IRB to determine whether or not the study still meets exemption criteria.

Please review your Post Approval Responsibilities here: Approved Documents Guidelines

For any questions regarding your protocol, please reach out to the IRB at irb@nu.edu.

Sincerely,

Dr. Joseph Marron, IRB Chair

Dr. Brianne Mongeon, Director, HRPP & IRB

Jenessa Eberhardt, Associate Director, HRPP & IRB