Semantic Annotation of Mixed-unit Numeric Data

by

Amir Behrad Khorram Nazari

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

A significant portion of quantitative information about entities in open-sourced data and data lakes is presented in tabular format, yet these tables often lack consistent labeling and schema, complicating querying and integration tasks. This thesis addresses the challenge of identifying and annotating numerical columns that may contain data from multiple sources with inconsistent units. For instance, weight measurements might be expressed in kilograms or pounds without clear unit indications. We propose a robust method for annotating mixed-unit numeric data, develop a benchmark for this task, and introduce an algorithm that accurately detects semantic types (e.g., height) and links them to corresponding types in a knowledge graph. Our method outperforms state-of-the-art techniques, particularly in detecting mixed units and assigning appropriate semantic labels. Our evaluation of mixed-unit columns with varying levels of complexity confirms the effectiveness of our approach in improving annotation accuracy. Additionally, our evaluation provides new insights into the accuracy of annotating mixed-unit columns, a problem that has not been thoroughly explored in previous work.

To my family, whose unwavering support and encouragement have been my guiding light throughout this journey. And to my friends, who have stood by me and motivated me along the way.

Table of Contents

1 Introduction **1**

1.1 Background and Motivation 1
1.2 Use Cases 2
1.3 Problem Statement 4
1.4 Challenges in Column Semantic Mapping 4
1.5 Related Work and Limitations 6
1.6 Our Contribution 7
1.7 Thesis Structure 8

2 Related Work **10**

2.1 Column Type Annotation 10
2.1.1 Annotating Textual Columns 10
2.1.2 Annotating Numerical Columns 12
2.2 Tabular Data Search and Annotation 15
2.2.1 Finding Related Tables 15
2.2.2 Annotating Relational Data on the Web 17
2.3 Summary 18

3 Dataset Construction **19**

3.1 Reflectivity 19
3.2 MixedSAND Dataset 23

4 Methodology **25**

4.1 Background on Single Unit Column Annotation 26
4.2 Annotating Mixed-Unit Columns 27
4.2.1 Model Generation 27
4.2.2 Annotating Sub-Models 30
4.2.3 Aggregating Sub-Model Annotations 31
4.3 Relaxing the Assumption on Unit Count 31

5 Evaluation **33**
5.1 Experimental Setup 33
5.2 Detecting the Number of Units 34
5.3 Type Detection Performance 37
5.4 Stage 2 and 3 Evaluation 38
5.5 Evaluating the Impact of Relative Difference 41
5.6 Assessing Robustness to Changes in the Number of Units 42
5.7 Impact of Query Column Size on the Model 44

6 Conclusion **46**
6.1 Summary 46
6.2 Future Work 47

References **48**

List of Tables

1.1 The height and weight of NHL players integrated from two sources: https://www.espn.co.uk/ and https://records.nhl.com/. 3

4.1 Area of a few Canadian cities 28

5.1 Accuracy of MixedSAND compared to a KDE-based baseline in detecting multi-unit columns (%) 36
5.2 Performance of MixedSAND compared to KDE on determining the number of units in columns 36
5.3 Performance comparison of MixedSAND vs. SAND on semantic labeling of the Easy dataset 38
5.4 Performance comparison of MixedSAND vs. SAND on semantic labeling of the Medium dataset 38
5.5 Performance comparison of MixedSAND vs. SAND on semantic labeling of the Hard dataset 39
5.6 Performance comparison of MixedSAND vs. SAND and KS-test on semantic labeling of the WDC dataset 40
5.7 Performance of the Stages 2 and 3 of Model on the Wikitables Dataset 40
5.8 Impact of relative vs. absolute distance on MixedSAND's clustering accuracy 41
5.9 Accuracy of MixedSAND in distinguishing single-unit vs. mixed-unit columns using relative difference vs. absolute distance (%) 42
5.10 Column type annotation performance of MixedSAND on columns with varying numbers of units 43

List of Figures

3.1 Data with high reflectivity and their distributions, reflectivity = 0.875 21
3.2 Distribution of non-reflective data, reflectivity = 0 22

4.1 MixedSAND column annotation pipeline 26
4.2 Normalized area values (min-max scaling) for Canadian cities in Table 4.1. The visual gap between the third and fourth rows highlights the scaling issue leading to a misclassification. 29

5.1 Distribution of a mixed-unit column 35
5.2 Model Accuracy with Varying Query Column Size 45

Chapter 1

Introduction

1.1 Background and Motivation

In the digital age, an ever-growing number of tables reside in data lakes and open sources, containing potentially valuable data for various tasks such as question answering, fact verification, and decision-making processes [1, 2]. Data lakes, with their offering of a centralized repository to store both structured and unstructured data at any scale, are becoming increasingly prevalent in both the public and private sectors [3]. For instance, open government data can include tables of economic indicators, health statistics, or demographic information, while corporate data lakes might store customer information, sales data, or product inventories [4]. These datasets are rich in information and have the potential to drive significant insights and innovations across multiple domains.

However, effectively leveraging these tables for downstream tasks requires a clear understanding of the semantics of each column and the relationships between different columns. This semantic understanding is crucial for tasks such as query generation, data integration, and advanced analytics [5, 6]. For example, in the context of question answering systems, accurately understanding the meaning of columns can significantly improve the system's ability to provide correct and relevant answers. Similarly, in fact verification, the accuracy and the reliability of the verification process depend heavily on the correct interpretation of the data columns.

Tables within corporate databases also utilize semantic labels to enhance the performance of query generation tools. Semantic labels help bridge the gap between raw data and the meaningful insights that can be derived from it. They provide context, making it easier to interpret data correctly and use it effectively in analytical processes. For instance, a column labeled "Customer Age" in a sales database may provide information that can be used to segment customers for targeted marketing campaigns or to analyze age-related purchasing patterns.

1.2 Use Cases

Quantitative information about entities constitutes a significant portion of columns in these table repositories. However, the semantics of these quantitative columns are often inadequately represented [7]. For instance, consider a list of hockey players sourced from two different sporting websites and integrated into a single table without meticulous examination or mediation. As shown in Table 1.1, one data source represents the height (column C1) in meters and the weight (column C2) in kilograms, while the other data source expresses the height in feet and the weight in pounds. This amalgamation of data from disparate sources often overlooks unit consistency or conversion. This inconsistency in units poses a significant challenge for data integration and analysis. Without a standardized understanding of what each column represents, the data cannot be effectively used for downstream tasks such as information extraction and question answering. Therefore, it is essential to identify quantity columns with mixed units and assign accurate semantic labels to them. Accurate semantic labels ensure that the data can be correctly interpreted and utilized in various analytical tasks.

As another use case, imagine integrating medical records from different hospitals where patient weights are recorded in kilograms at one hospital and in pounds at another. Without converting these units to a common standard, any analysis of patient weight data would be flawed, leading to incorrect conclusions. Similarly, in financial

Entity	C1	C2	Source
Player	1.88	84	espn.co.uk
	1.85	87	
	1.85	93	
	1.93	104	
	6.25	192	nhl.com
	6.17	218	
	6.25	208	

Table 1.1: The height and weight of NHL players integrated from two sources: https://www.espn.co.uk/ and https://records.nhl.com/.

datasets, different sources might report monetary values in different currencies, and failing to account for these differences could lead to erroneous financial analyses.

The issue of mixed-unit columns becomes particularly prominent in scenarios involving table integration tasks, such as table union or stitching [8–10]. When tables from different sources are combined, columns may end up containing data in different units. For instance, during a table union operation, two columns representing the same attribute (e.g., height or weight) might be integrated despite being recorded in different units. Consequently, these mixed-unit columns require proper annotation to ensure their utility in subsequent data processing and analysis tasks. This is not only relevant for ensuring data quality but also for enabling accurate data analysis, which is critical for applications in data science and business intelligence. Moreover, several data integration tasks, such as data warehousing, ETL (Extract, Transform, Load) processes, and the creation of unified data views, frequently involve the merging of datasets from multiple sources. These tasks often result in columns that mix different measurement units, leading to inconsistencies that must be addressed. Accurately annotating these mixed-unit columns is essential for maintaining the integrity of integrated datasets and for performing reliable data analysis across various domains.

1.3 Problem Statement

The problem addressed in this thesis is the semantic annotation of numerical columns, with a particular focus on those with mixed units. Specifically, we aim to develop methods for identifying whether a column contains mixed units or not and annotating the semantic type of that column. By overcoming the challenges posed by unit inconsistencies, our goal is to enhance data integration and analysis capabilities.

Our study aims to address the problem by investigating whether quantity columns with mixed units can be identified and accurately labeled with semantic annotations. We leverage knowledge graphs as a source for semantic types. Knowledge graphs provide a structured representation of knowledge, capturing entities, their attributes, and the relationships between them [11, 12]. Using knowledge graphs for semantic labeling offers several benefits, including enhanced data interoperability, improved query performance, and more accurate data integration.

Knowledge graphs, such as DBpedia[1], Wikidata[2], and Google's Knowledge Graph[3], have been extensively used in various applications due to their ability to provide rich, contextual information about entities [13, 14]. By mapping table columns to entities and types in a knowledge graph, we can achieve a deeper understanding of the data's semantics. Detecting those mappings not only aids in data integration and query generation but also enhances the overall quality and usability of the data.

1.4 Challenges in Column Semantic Mapping

Mapping the columns of a table to a knowledge graph presents several challenges, particularly for numerical columns:

1. **Existence of Desired Semantic Types**: If the desired semantic types or column values do not exist in the knowledge graph, existing methods are unable

[1] http://dbpedia.org/
[2] https://www.wikidata.org/
[3] https://developers.google.com/knowledge-graph

to make accurate predictions. Knowledge graphs might lack specific types or values needed for precise mapping. For instance, a knowledge graph might have extensive information about people and locations but lack detailed types for certain scientific measurements or specialized industry data.

2. **Measurement Inaccuracies**: Quantitative measurements of identical entities from different sources seldom match precisely due to inherent inaccuracies in measurements and reporting. For example, the height of the Eiffel Tower is listed as 330 meters on Wikipedia, while it is stated as 324 meters on Wikidata. These discrepancies arise from measurement errors, rounding differences, or updates over time. Such variations are common in many domains, including scientific research, healthcare, and public records.

3. **Dynamic Nature of Quantitative Data**: Quantitative data frequently change over time due to the dynamic nature of measurements. Attributes such as the name and nationality of a player are less likely to change, whereas measurements such as height and weight can change regularly. This variability adds complexity to the task of semantic labeling. For example, consider a database tracking the population of cities over time; the population figures will change annually, requiring constant updates to maintain accurate semantic labels.

4. **Mixed Units**: Quantitative data may be reported using different units (e.g., kilograms and pounds for weight), complicating the mapping of columns with mixed units or those inconsistent with the knowledge graph. Ensuring unit consistency is critical for accurate data interpretation. For instance, in international trade data, weights might be recorded in metric tons in some countries and short tons in others, necessitating careful unit conversion to ensure accurate analysis.

Due to these challenges, reliably determining the semantic type of numerical data

and assigning appropriate annotations remains a formidable task. Accurate semantic labeling is essential for ensuring data quality and facilitating effective data integration and analysis.

1.5 Related Work and Limitations

Various methods exist for detecting the semantic types of columns, often involving mapping the columns to types in knowledge bases. However, most approaches are tailored for textual data [15–17]. Textual data, such as names and categories, are often easier to map to predefined types in a knowledge graph. For example, a column labeled "Country" can be easily mapped to country entities in a knowledge graph. Despite attempts to adapt these methods for numerical data, progress has been slow, and the adaptability of the methods remains questionable [17].

Many approaches targeting numerical data leverage the statistical properties of the columns, such as mean and standard deviation, along with statistical testing, to assign a type [18–21]. These methods assume that the values of each semantic type follow a known distribution and that the query column is a random sample from the same distribution. However, these assumptions are often violated. Statistical tests can reject the null hypothesis—that there is no significant difference between the means of the two distributions—when the means are significantly different, but they cannot confirm the null hypothesis when the means are similar.

Recent work attempts to relax some of these distributional assumptions, yet most methods assume that column values are uniformly expressed using the same unit [22]. This assumption may not hold when data are gathered from multiple sources, as discussed earlier. For example, data integration tasks often involve merging data from sources with different unit conventions, leading to mixed-unit columns.

1.6 Our Contribution

Our work aims to further relax assumptions about data distribution, specifically avoiding the assumption that column values are uniformly expressed using the same unit. To tackle the challenges associated with mixed-unit data, we propose a Three-staged Numeric Data Annotation Pipeline. This pipeline systematically addresses the problem by first generating plausible data models, then assigning semantic types, and finally aggregating the results. Our main contributions include:

1. **Three-Staged Numeric Data Annotation Pipeline**:

   (a) **Model Generation**: Plausible models of data subsets are generated, and data points are assigned to those sub-models. This stage involves clustering data points based on their statistical properties and unit conventions. For example, height measurements in meters and feet can be clustered separately, allowing the model to handle each subset appropriately.

   (b) **Type Annotation**: A semantic type is assigned to each sub-model following a cost optimization framework for numerical data. This stage leverages knowledge graphs to identify the most appropriate semantic type for each subset. For instance, height measurements might be mapped to a "Height" type in the knowledge graph, regardless of the unit used.

   (c) **Aggregation Phase**: Sub-model costs are aggregated to estimate the cost of each unifying model covering the entire column values and to select models with the least cost. This stage ensures that the final annotation is both accurate and consistent across the entire dataset. By considering the cost of different unifying models, the approach balances accuracy and complexity, selecting the most suitable semantic label for the entire column.

2. **Mixed-Unit Numeric Annotation Benchmark**: We develop a benchmark

for testing annotation methods on columns with mixed units. This benchmark provides a standardized dataset for evaluating the performance of different annotation approaches. The benchmark includes a variety of columns with mixed units, sourced from diverse domains such as sports, healthcare, and finance, ensuring comprehensive testing of annotation methods.

Our experimental evaluation on a diverse collection of data, including our benchmark and other datasets, demonstrates the superiority of our approach over strong baselines from the literature in both detecting and annotating mixed-unit numeric columns. Our approach effectively handles mixed-unit columns, providing accurate and consistent semantic labels that facilitate downstream analysis.

1.7 Thesis Structure

The remainder of this thesis is organized as follows:

- **Chapter 2** reviews related works, highlighting the limitations of existing methods and positioning our contributions within the broader context. This chapter provides an in-depth analysis of the current state-of-the-art methods, their strengths and weaknesses, and how our approach addresses existing gaps.

- **Chapter 3** details the construction of our datasets, including the mixed-unit numeric annotation benchmark. This chapter describes the process of collecting, cleaning, and preparing the datasets, ensuring their suitability for our experimental evaluation.

- **Chapter 4** outlines our methodology, explaining the steps involved in the proposed approach. This chapter provides a detailed explanation of the model generation, type annotation, and aggregation phases, highlighting the novel aspects of our approach.

- **Chapter 5** presents the experimental setup and results, comparing our approach with state-of-the-art methods. This chapter includes a comprehensive analysis of the performance of our approach on various datasets, demonstrating its effectiveness and robustness.

- **Chapter 6** concludes the thesis, summarizing our contributions and discussing the implications of our findings. This chapter also outlines potential future work, suggesting directions for further research in the field of semantic annotation of numerical data.

Chapter 2

Related Work

Related work can be categorized into two main areas: (1) column type annotation, and (2) table search and annotation.

2.1 Column Type Annotation

The literature on semantic labeling of table columns, or column type annotation methods, can be categorized into two main themes: textual columns and numerical columns. This structured overview highlights the specific advancements, benefits, and limitations of the relevant approaches, setting the stage for our work in the following chapters.

2.1.1 Annotating Textual Columns

Semantic labeling of tables has been extensively studied with a focus on textual columns. Some of these approaches may be applied to numerical data. One notable work in this area is by Efthymiou et al. [15], who explore the use of semantic embeddings to match web table rows with entities in knowledge bases. The approach leverages the semantic similarities between entities by embedding them in a continuous vector space, which captures the nuances of entity names and contexts. This method has shown significant improvement in annotation accuracy for textual data. However, it struggles with interpreting numerical data, especially those with mixed

units, as the embedding approach does not naturally extend to numerical values that require unit normalization and contextual understanding.

Zhang et al. [23] present a method for semantic table interpretation that improves upon the original TableMiner [24] system. Their approach, named TableMiner+, uses contextual information both within and outside the tables to enhance annotation accuracy. The incremental bootstrapping approach they adopt begins with preliminary annotations that are refined iteratively, reducing computational overhead and improving efficiency. Despite its strengths in dealing with textual data, TableMiner+ struggles with numerical columns, particularly when these columns contain mixed units. This is because TableMiner+ assumes that all values in a numerical column are in the same unit, simplifying the process of linking the column to a single property in a knowledge base. However, when units are mixed, the method lacks the ability to detect and convert between different units, leading to potential errors in semantic annotation.

Limaye et al. [16] introduce a comprehensive multi-step method to link table cells to knowledge bases through statistical and semantic analysis. The process effectively handles relationships between entities within tables, improving the robustness of the annotations. This method is mainly designed for textual data, but it can also handle numerical data. However, it struggles with mixed-unit data because it assumes each column is single-unit and tries to link all values to a specific property in the knowledge base.

Chen et al. [17] introduce ColNet, a neural network-based framework designed to annotate column types in web tables. ColNet integrates knowledge base (KB) reasoning with machine learning techniques, using convolutional neural networks (CNNs) to learn semantic representations of columns. It employs a multi-task learning approach that simultaneously performs column type prediction and entity linking, leveraging the relationships between these tasks for improved accuracy. While ColNet shows enhanced accuracy in column type prediction for textual data by combining KB reason-

ing with CNNs, its performance is dependent on the quality and comprehensiveness of the underlying KB. Moreover, the use of CNNs and multi-task learning increases the computational complexity, which may limit its application in resource-constrained environments.

Nishida et al. [25] present TabNet, a hybrid deep neural network architecture for table type classification that treats tables as matrices of text. TabNet uses a recurrent neural network to encode sequences of tokens for each cell, capturing both local and global semantic relationships within tables. While this architecture improves classification accuracy for textual data, it requires extensive computational resources, impacting its scalability. Additionally, TabNet primarily addresses textual data, leaving numerical data less effectively handled.

2.1.2 Annotating Numerical Columns

Several works have focused specifically on the semantic labeling of numerical columns, often highlighting the unique challenges posed by numerical data, such as unit inconsistencies and distributional assumptions.

Takeoka et al. [26] employ probabilistic models to annotate table columns, effectively handling ambiguous entity names and varying contexts. However, a key limitation is its reliance on the assumption that column values are uniformly expressed in the same unit, which is often violated in real-world datasets where numerical data from multiple sources are aggregated without unit standardization.

Alobaid et al. [18] apply fuzzy clustering techniques to annotate numerical data. Fuzzy clustering allows for the identification of potential semantic types by grouping similar data points. This method effectively handles datasets with varying degrees of uncertainty and imprecision. However, it heavily relies on statistical properties such as mean and standard deviation, making it sensitive to outliers and skewed data distributions. This sensitivity can limit its effectiveness in heterogeneous datasets where these statistical properties are not consistent across the data points.

Zhang et al. [27] employ a hybrid machine learning model to detect semantic types by integrating both local and global contextual information to enhance accuracy. The method uses a combination of recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to process table data. While effective in certain contexts, this approach requires substantial computational resources and high-quality training data, which can be challenging to obtain for numerical data. The need for extensive labeled datasets remains a significant limitation, particularly when dealing with numerical columns.

Nguyen et al. [28] introduce an embedding-based approach specifically for numerical data, converting numerical columns into a continuous vector space that captures their semantic meanings. By mapping these numerical vectors to a knowledge base, the approach can effectively determine the most relevant semantic labels. While promising, the approach's performance is heavily dependent on the quality and representativeness of the training data. If the training data does not adequately represent the diversity of the numerical columns in practice, the resulting embeddings may fail to capture the full range of semantic meanings.

Neumaier et al. [19], Pham et al. [20], and Ramnandan et al. [21] propose methods relying heavily on the statistical properties of numerical columns to assign semantic types. These methods use statistical tests to determine the relevance of data to specific types, assuming that values within each semantic type follow a known distribution. While these approaches can be effective in certain scenarios, they often struggle with real-world datasets where the assumptions about data distributions are violated. The inability to handle columns with mixed units further limits their applicability.

Kacprzak et al. [29] emphasize the importance of high-quality training data for effective semantic labeling. Their approach, while robust for consistent datasets, encounters significant challenges with heterogeneous data where statistical properties are not uniformly expressed. The reliance on statistical distributions makes these

methods less adaptable to datasets with diverse sources and inconsistent units.

Su et al. [22] introduce SAND, a novel method for annotating numeric columns in web tables by linking them to properties in knowledge graph. SAND leverages the semantic information available in knowledge graphs, bypassing the need for contextual information that may be missing or labeled data, which can be costly and time-consuming to obtain. The method involves identifying numeric columns in web tables and linking these numbers to corresponding properties in a knowledge graph. This process enhances the semantic understanding of the data, making it easier to query and analyze. One of the main benefits of SAND is its reliance solely on existing semantic information within knowledge graphs, which makes the approach more robust and less dependent on external data sources. However, the quality and comprehensiveness of the annotations are heavily dependent on the completeness and accuracy of the underlying knowledge graph. Additionally, the approach might struggle with highly heterogeneous or noisy web tables, where the numeric data does not clearly map to well-defined properties in the knowledge graph.

Ho et al. [30] present a method for extracting quantity facts from web tables through the normalization and contextualization of numerical values. This approach significantly improves the precision and relevance of search results involving quantitative data. The method involves recognizing quantities, normalizing their values and units, and aligning them with the appropriate entities in the tables. Contextual cues are then used to match sophisticated queries with modifiers. However, the process requires extensive contextual information and computational resources, limiting its scalability for large datasets.

Kruit et al. [31] introduce TAKCO, a large-scale platform designed to extract novel facts from web tables for inclusion in knowledge graphs. Takco focuses on identifying novel facts while maintaining high precision, addressing the issue of redundancy in traditional methods. The platform's novel interpretation algorithm enhances the accuracy of fact extraction by analyzing the structure and content of web tables.

However, the process requires sophisticated algorithms and significant computational resources, impacting scalability. The platform's effectiveness also depends on the quality and diversity of the web tables it processes.

Ibrahim et al. [32] address the quantity alignment problem by linking numerical data in tables with their textual context. This bidirectional linking enhances the interpretability and usability of web tables by connecting numerical facts with their explanations or references in the surrounding text. The method's reliance on high-quality contextual information can be a limitation, especially when the text is sparse or ambiguous. Additionally, the normalization and matching processes can be computationally intensive, particularly for large datasets.

2.2 Tabular Data Search and Annotation

A substantial body of research focuses on identifying tables that are related through joinability, unionability, or semantic similarity. Many of these studies either involve annotating tables or benefit from an annotation process.

2.2.1 Finding Related Tables

The work on finding related tables is aligned with ours, as it involves comparing tables and identifying whether two columns match in content or semantic type.

Nobari et al. [33] address the challenge of transforming tables to ensure joinability when data from different sources are formatted inconsistently. The approach focuses on normalizing data formats to enable seamless integration of tables. The relationship between this task and our work lies in the shared goal of addressing inconsistencies in data. While Nobari et al.'s work focuses on making columns joinable despite format discrepancies, our task involves handling inconsistencies where a column contains mixed-unit data, aiming to accurately annotate these columns by finding similar columns in a knowledge graph.

Khatiwada et al. [8] introduce Santos, a framework that improves union search

accuracy by considering the semantic relationships between pairs of columns within tables. Unlike traditional table union methods that rely solely on metadata or basic column metrics, Santos uses semantic models to better understand the relationships between columns, which can lead to more accurate unions. However, this relationship-based approach can also contribute to the unintentional merging of columns with different units, creating mixed-unit columns.

Nargesian et al. [9] address the challenge of identifying unionable tables in large datasets, particularly in open data repositories by proposing a probabilistic approach that models unionability based on domain-specific characteristics, allowing for flexible and scalable union searches. While this method enhances the ability to combine tables from diverse sources, it also increases the likelihood of merging columns with different units, thus creating mixed-unit columns that pose challenges for subsequent data analysis.

Narisawa et al. [34] explore the concept of numerical common sense by extracting and analyzing numerical expressions from web contexts. Their study introduces methods to infer whether a number is large, small, or normal based on its context, which can aid in understanding numerical annotations in data tables. This work provides an interesting perspective on the semantic interpretation of numerical data, especially when different units and ranges are involved. It is relevant to our research because it offers a method for determining whether a numerical value belongs to a specific type and unit, which is essential for ensuring the correct semantic labeling of mixed-unit columns in tabular data.

These works also highlight the importance of considering the effects of table union processes on column consistency, particularly in the context of numerical data. The union of tables from different sources can lead to mixed-unit columns, which require careful handling during semantic labeling and data integration tasks.

2.2.2 Annotating Relational Data on the Web

This group of works focuses on the task of annotating relational data on the web, which aligns with the broader objective of semantic annotation in our research.

Cannaviccio et al. [35] focus on annotating relational data on the web by using language models to identify and rank relationships between entities in tables. Their approach improves the accuracy of relational annotations, facilitating the extraction of meaningful insights from web-based tables. While their work centers on relational annotations, our research addresses the semantic annotation of numerical data, particularly columns with mixed units. Both approaches contribute to enhancing the semantic understanding of tabular data, albeit in different contexts.

Bollengala et al. [36] introduce the concept of relational duality in their work on the unsupervised extraction of semantic relations between entities on the web. This approach captures relationships either by listing all instances (extensional) or by defining paraphrases (intensional), which is essential for tasks like information extraction and relation detection. Although their focus is on web data, this work aligns with our research's goal of enhancing the semantic understanding of data—specifically, numerical data in tables—by accurately identifying and labeling relationships and semantic types.

Weston et al. [37] present an approach that connects language and knowledge bases through embedding models for relation extraction. Their method jointly learns embeddings for words, entities, and relationships from both text and knowledge bases, effectively integrating these sources of information. This approach enhances the extraction of semantic relations by leveraging both textual data and structured knowledge, which aligns with our research's goal of using knowledge graphs to semantically annotate mixed-unit columns in tabular data.

2.3 Summary

The review of existing literature on semantic labeling of tables reveals significant progress in both textual and numerical column type annotation. Approaches for textual columns, such as those leveraging semantic embeddings and neural network architectures, have achieved notable success in improving annotation accuracy by capturing semantic relationships within the data. However, these methods often struggle when extended to numerical columns, particularly when dealing with mixed units or requiring precise contextual understanding.

For numerical columns, methods that rely on statistical properties, probabilistic models, and knowledge graph integration have been developed to address the unique challenges posed by numerical data. While these approaches offer robust solutions for specific scenarios, they often face limitations related to assumptions about data distribution, the need for uniform data units, and the dependency on high-quality training data.

In the area of tabular data search and annotation, research has primarily focused on identifying related tables based on joinability, unionability, and semantic similarity. These methods typically involve annotating tables or utilizing existing annotations to enhance search capabilities and facilitate data integration. These approaches have significantly improved the ability to discover and link related tables, enabling more effective combination and analysis of data from diverse sources. However, the process of joining and unioning tables can lead to the creation of mixed-unit columns, which is a challenge that this thesis aims to address.

Overall, the literature highlights the need for approaches that can effectively handle the heterogeneity of numerical data without relying on strict assumptions about data uniformity. Our work builds on these insights, aiming to further advance the field by introducing a method that can more accurately and comprehensively annotate numerical data across a wide range of applications.

Chapter 3

Dataset Construction

There is a lack of datasets for evaluating semantic annotation of numeric columns. The SemTab challenge [38], held annually since 2019, evaluates tabular data mapping to a knowledge graph, but it primarily includes textual columns. In SemTab Challenge 2021 [39], a task was introduced in Round 2 to identify the semantic relationship between an entity and a numeric property (e.g., (Kielzugvogel,5.8) and (MT explosive motorboat,5.62)). However, this task differs from annotating a numerical column. Recent papers [18, 22] have introduced small manually annotated datasets, but all column values in these datasets have the same semantic type and unit. To our knowledge, there is no public multi-unit numeric column annotation dataset.

To fill this gap, we introduce a mixed-unit numeric column annotation dataset with varying levels of separability difficulty between units. To quantify this separability difficulty, we introduce the concept of *reflectivity* before discussing our dataset.

3.1 Reflectivity

Some mixed-unit columns are challenging to separate, even for human experts, due to significant overlap in data ranges from different units. In contrast, other mixed-unit scenarios are readily identifiable. To quantify this difficulty, we introduce the concept of *reflectivity*, which was used in a different context to quantify the interference between data dimensions [40].

Reflectivity, as defined by Agrawal and Srikant [40], measures the likelihood that a data point's reflection exists within the dataset. If $\vec{x_i}$ denotes a data point in a multidimensional space, the reflections of $\vec{x_i}$ include all points obtained by permuting the coordinates of $\vec{x_i}$, including the point $\vec{x_i}$ itself. For example, the reflections of (1,2) will be $\{(1, 2), (2, 1)\}$. The reflectivity of a 2-dimensional dataset D points is formally calculated as:

$$\text{Reflectivity}(D, r) = 1 - \frac{1}{|D|} \sum_{\vec{x_i} \in D} \frac{\theta(\vec{x_i})}{\rho(\vec{x_i})} \tag{3.1}$$

where $\theta(\vec{x_i})$ denotes the number of points within Euclidean distance r of $\vec{x_i}$, $\rho(\vec{x_i})$ is the number of points in D with at least one reflection within distance r of $\vec{x_i}$, $|D|$ is the dataset cardinality, and r is a distance threshold chosen experimentally. For higher dimensions, the reflectivity is computed as the average reflectivity across all 2-dimensional subspaces.

In our work, we are interested in the reflectivity relationship between different units of a property. Given two sets of quantities, U_1 and U_2, both measuring the same property but in different units, each value in U_1 can be considered similar to values in U_2 if their magnitudes are close. To capture this relationship between units, we create our dataset D as the Cartesian product $U_1 \times U_2$, which includes every pair (a, b) where $a \in U_1$ and $b \in U_2$. A high reflectivity in this dataset indicates a greater average number of reflections falling within distance r of existing data points. Intuitively, this scenario implies an increased degree of overlap between the data represented by the units U_1 and U_2. This overlap makes it more challenging to separate these units. To illustrate, consider data points (a_i, b_i), $(a_j, b_j) \in$ D that are not within distance r of each other. If the reflection (b_i, a_i) of (a_i, b_i) is within distance r (a_j, b_j), it suggests that the values b_i and a_j, as well as b_j and a_i, are similar. Since we know a_i and a_j belong to U_1, and b_i and b_j belong to U_2, this proximity of values from different units suggests a greater degree of overlap between the units. Conversely, a low reflectivity suggests that the values in different units are largely distinct, with minimal overlap.

In our dataset generation, we set the value of r such that the average number of points within distance r is 2.5. This decision was made to capture reflections that fall within top 2-3 neighbours of a data point. This choice was based on experimental observations, aiming to balance the detection of overlapping values while controlling the reflectivity of the dataset. Setting the number of neighbors to 2-3 ensures that the reflectivity values remain meaningful; if set to 0, the reflectivity for all datasets would be 0, while a larger number would yield excessively high reflectivity values for most columns.

Entity	Size (meters)	Size (yards)
Soccer Fields	100	
	84	91.86
	110	
		100.58
	103	112.64
		82.30

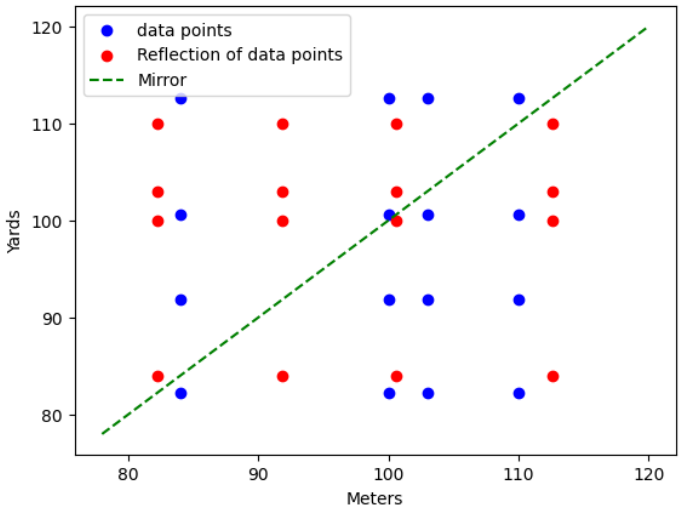

Figure 3.1: Data with high reflectivity and their distributions, reflectivity = 0.875

Examples of high and low reflectivity can be seen in Figures 3.1 and 3.2. In each of the figures, we present a table that includes two numerical columns, displaying data

in one or two different units. Figure 3.1 illustrates high reflectivity, while Figure 3.2 shows low reflectivity. In Figure 3.1, a dataset is constructed using all possible pairs where the first element is a size in meters and the second value is a size in yards. The figure shows these pairs and their reflections, which are close to the original points, indicating high overlap (as evident in the table). Conversely, Figure 3.2 shows player weights in pounds and kilograms, with distant reflections demonstrating minimal overlap.

Entity	Weight (lb)	Weight (kg)
Players	210	95.25
	194	
	150	
	176	
		67
	178.56	81
		73
		92

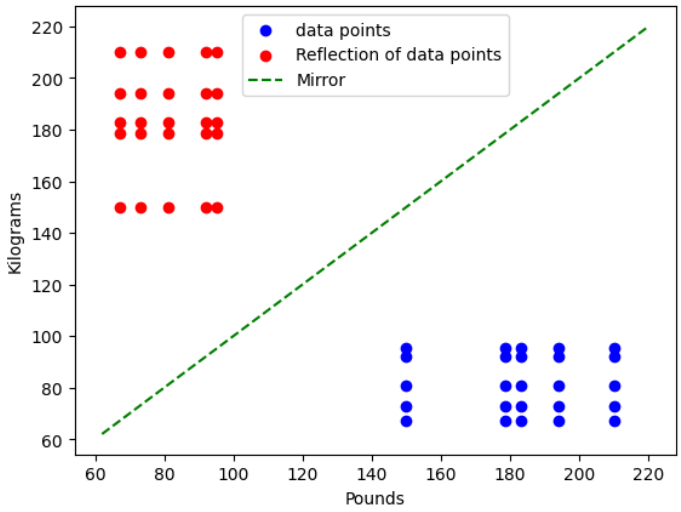

Figure 3.2: Distribution of non-reflective data, reflectivity = 0

3.2 MixedSAND Dataset

We introduce the MixedSAND dataset for evaluating multi-unit semantic annotation models. Leveraging real-world numeric data from Wikidata, we have transformed it into a robust dataset designed to test the model's ability to handle mixed units. The dataset is generated systematically, allowing control over key parameters such as unit counts, percentages of each unit within a column, and levels of data reflectivity within each column.

We generate three variations of the MixedSAND dataset, each consisting of 200 columns with 30 rows per column. Each column consists of data in two different units, with an equal distribution of values from each unit. The values of different units in each column are present in varying proportions and degrees of overlap.

- Easy Dataset: This dataset is designed to present the least challenge in identifying mixed-unit columns. The majority of columns (60%) exhibit low reflectivity (< 0.3), indicating minimal overlap between the two units present within each of these columns. This characteristic makes the identification of mixed-unit columns relatively straightforward in this dataset. The remaining columns are divided equally between those with moderate reflectivity (0.3 - 0.6) and those with higher reflectivity (> 0.6).

- Medium Dataset: This dataset introduces a moderate level of difficulty in identifying mixed-unit columns. The distribution of reflectivity levels differs from the Easy dataset. The majority of columns (60%) exhibit moderate reflectivity (0.3 - 0.6), indicating a noticeable overlap between data points with different units. The remaining columns are divided equally between those with low reflectivity (< 0.3) and those with higher reflectivity (> 0.6).

- Hard Dataset: This dataset presents the most challenging scenario for mixed-unit column identification. The majority of columns (60%) exhibit high re-

flectivity (> 0.8). This indicates that a substantial portion of the data points appear with both units, making it significantly more difficult to distinguish between the two units and identify mixed-unit columns. The remaining columns are equally divided between those with low reflectivity (< 0.3) and those with moderate reflectivity (0.3 - 0.6).

Evaluating models across these three diverse datasets with varying levels of difficulty in mixed-unit column annotation provides a comprehensive assessment of their performance and robustness across a wide range of potential challenges.

Chapter 4

Methodology

Consider a table with a set of columns $c_1, \ldots, c_n$, and let val(c_i) denote the set of values in column c_i. We focus on annotating columns containing only numeric values. Each table often describes a set of entities (e.g., person, organization, location) or relationships between entities, and this limits the set of types a column can take.

Problem 1 (Numeric Column Type Annotation) *Let T be a set of entity types, P be a set of properties and U be a set of units. A semantic type can be denoted as a triple <t, p, u> where $t \in T$, $p \in P$ and $u \in U$. The problem of column type annotation for a numeric column c_q is the task of assigning a semantic type to c_q.*

Our approach involves using the proximity of val(c_q) to the samples val(c) of candidate semantic types c.

In this work, we propose a solution that involves using the proximity of val(c_q) to the samples val(c) of candidate semantic types c. We utilize a knowledge graph (KG) to construct our candidate semantic types. For each candidate semantic type c, we require a representative set of data points, val(c). We operate under *the closed world assumption*, meaning that the candidate set is considered complete. This assumption is commonly made in similar approaches on annotating tabular data [15, 17, 22, 23].

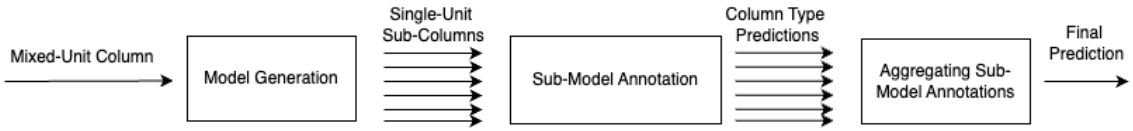

Figure 4.1: MixedSAND column annotation pipeline

4.1 Background on Single Unit Column Annotation

A few methods have been developed under the assumption that the input column consists of a single unit [18, 22]. Our methodology is built around SAND [22], a method that is shown to achieve the state-of-the-art performance for annotating single-unit numeric columns, and it is reviewed in this section. SAND[22] compares a given numeric column, known as the query column, with all candidate columns in the knowledge graph. This comparative analysis is executed by constructing a complete bipartite graph between the query column and the candidate column. On one side, nodes represent the numbers in the query column, while on the other side, they correspond to the numbers in one of the candidate columns. Edges in this graph signify the numerical disparity between the nodes they connect. Each mapping of the query column to a candidate column is represented with a subgraph and is associated with a cost. This cost is defined as the cumulative cost of the edges in the mapping.

SAND utilizes a minimum cost flow algorithm[41, 42] to identify the minimum-weight subgraph encompassing all nodes from the query column. The weight of this subgraph represents the cost associated with mapping the query column to the candidate column.

On the other hand, the model operates under the assumption that all query columns are single-unit columns, and each query column is compared with candidate columns that solely feature a singular unit. When units are mixed within the query column, the model maps values across different units from a mixed-unit query column to single-unit candidate columns. Consequently, the model fails to produce

satisfactory results when the query column consists of mixed units, as shown in our evaluation section.

Furthermore, this approach presumes that all values in the query column align with units available in the knowledge graph. If the data exists in a different unit within the knowledge base, SAND's predictive accuracy is compromised.

4.2 Annotating Mixed-Unit Columns

We initiate the process of annotating by considering the column to be mixed-unit [1]. Our approach consists of a three-staged pipeline, as illustrated in Figure 4.1: (1) *model generation*, where plausible models of data subsets are generated, and data points are assigned to those sub-models, (2) *sub-model annotation*, where a semantic type is assigned to each sub-model, following a cost optimization framework for numerical data similar to SAND [22], and (3)*aggregation phase*, where sub-model costs are aggregated to estimate the cost of each unifying model covering the entire column values, and to select models with the least cost.

4.2.1 Model Generation

Given a column with mixed units, each grouping of the column values can be associated with a specific unit. The total number of possible groupings or models of data is determined by the powerset of the column values. Our hypothesis is that *quantities or measurements with the same unit are more likely to be closer to each other than those with different units*. Based on this hypothesis, a clustering of the column values should place values with the same unit in the same cluster. As the simplest and most commonly used clustering method, k-means is an option. However, there are two key problems that need to be addressed:

1. *Choosing an appropriate distance function*: The scale of the numbers can vary significantly between units (e.g., millimeters and kilometer), making the abso-

[1]This assumption is relaxed in Section 4.3.

Entity	Property	Unit
City	684.4	km^2
	115	
	87,430,000	m^2
	630,200,000	

Table 4.1: Area of a few Canadian cities

lute difference between two quantities less meaningful when they are of different units.

2. *Determining the number of units (k)*: Often the number of units in a mixed column is unknown, which complicates the determination of the appropriate number of clusters k. We initially assume the number of units in a column is known. This condition is relaxed in Section 4.3.

Choosing an appropriate distance function The default distance function in k-means is the Euclidean distance, which treats all differences between quantities the same. This poses a problem when clustering quantities with different measurement scales and units. For example, consider the "Area" column in Table 4.1, which lists the areas of Canadian cities in both square meters and square kilometers. Without knowing the units, the default k-means distance function (Euclidean distance) would incorrectly cluster the first three cities with areas 684.4, 115, and 87,430,000 together, while separating the fourth city with an area of 630,200,000. This error occurs because the default metric fails to account for the scale variations inherent in mixed-unit data, as depicted in Figure 4.2 which uses min-max normalization for clearer illustration. In contrast, humans typically have no problem recognizing these scale differences and will likely cluster the first two cities with measurements in square kilometers together and the last two cities with measurements in square meters together. Our proposed solution to this problem is the Bray-Curtis distance, a normalized relative difference

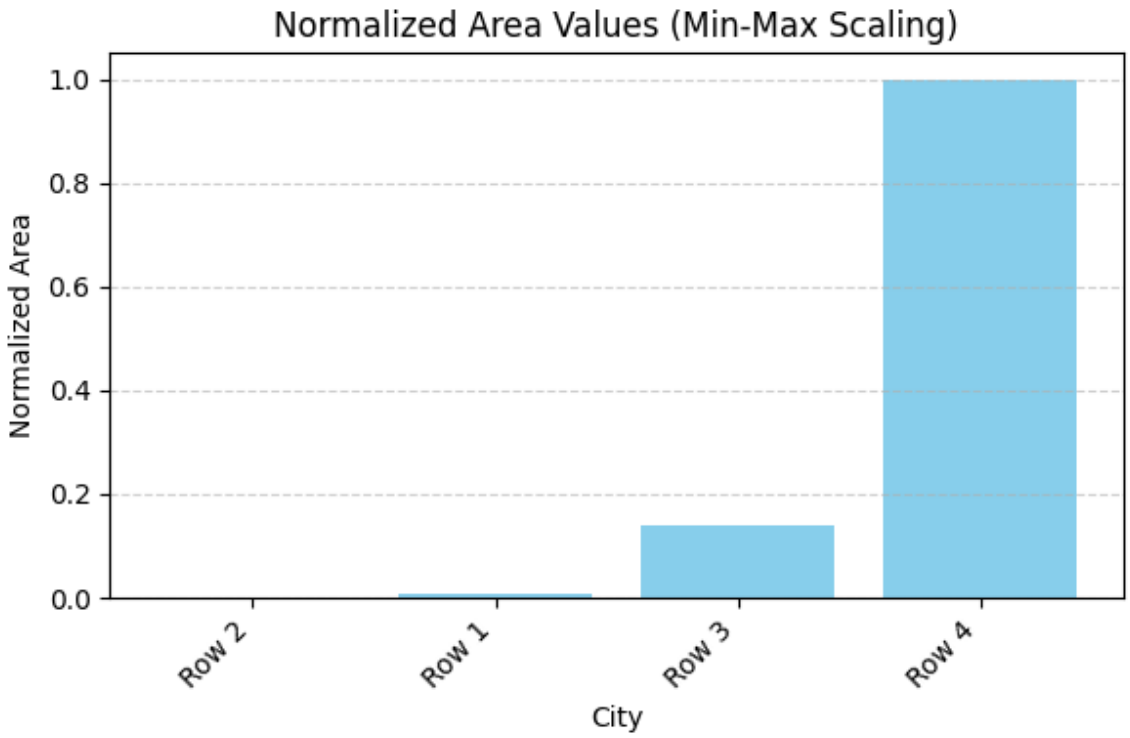

Figure 4.2: Normalized area values (min-max scaling) for Canadian cities in Table 4.1. The visual gap between the third and fourth rows highlights the scaling issue leading to a misclassification.

function defined in Equation 4.1. This distance function is commonly used in ecology for comparing quantities with different scales [43]. The Bray-Curtis distance incorporates the magnitudes of quantities into the comparison, effectively replicating this intuitive clustering of city areas in our example and mitigating scaling errors. In our city area example, the Bray-Curtis distance between the areas in Rows 3 and 4 is 0.7563, indicating that Row 3 is considered closer to Row 4 than Row 1, as the distance between Rows 1 and 3 is 0.9999. Section 5.5 confirms that this adaptation results in more accurate unit-based clustering with the k-means algorithm. Additionally, the Bray-Curtis distance is a metric, ensuring smooth integration within clustering algorithms.

$$\text{distance}(x, y) = \frac{|x - y|}{x + y} \quad \text{for } x, y \geq 0 \text{ and } x + y > 0. \tag{4.1}$$

Clustering We employ a clustering approach using the k-means algorithm, but with the Bray-Curtis distance function, with k set to the number of units, to partition a mixed-unit column into sub-columns. Initially, we assume the number of units is known, a presumption we relax in Section 4.3. Assuming the clustering process is entirely accurate (for clarity in the following sections, though errors may occur), each resulting sub-column should predominantly contain values associated with the same semantic type and unit. Next, we discuss the annotation process for each sub-column.

4.2.2 Annotating Sub-Models

In our model generation, a query column is partitioned into sub-columns, and each sub-column is expected to map to an atomic semantic type and unit. This mapping is performed using a single-unit annotation method; in our case, this is done using SAND, as discussed in Section 4.1. However, there are two issues in using a single-unit annotation process. First, while the knowledge graph may be complete in terms of the semantic types covered, it is less likely to include all possible units for each property or sufficient samples for each unit. Thus, the knowledge graph may not have the exact unit that matches a query sub-column. To address this, we compile a set of possible units and conversion rates [2] between all convertible units, though it's not exhaustive. When comparing a candidate column from the knowledge graph to a query column, we expand the knowledge graph column with all other applicable units for which we have conversion rates. This type expansion is also applied to data samples using our conversion tables, meaning each new unit will have data samples. For example, if the knowledge graph has human height in centimeters, the data samples in centimeters are mapped to meters, feet, etc.

Another issue is the cost function in SAND, which is defined as the sum of the edge weights in a mapping, with the edge weight determined by the absolute difference between matching quantities. We replace this function with one that provides a more

[2]https://exchangeratesapi.io https://www.unitconverters.net

accurate cost estimate when dealing with multiple units, as discussed in the next subsection.

4.2.3 Aggregating Sub-Model Annotations

In the previous two steps, a mixed-unit column is separated into sub-columns, and each sub-column is mapped to a semantic type. When a column consists of multiple sub-columns, those sub-columns may be assigned different semantic types and units, each associated with a cost. The cost of a sub-column prediction is defined as the sum of the edge weights in the subgraph that maps the query sub-column to a candidate knowledge graph type.

Our MixedSAND model treats sub-column type predictions as possible candidates for the entire column. The cost of a prediction for the entire column is defined as the cumulative costs of sub-column predictions. However, this cost model is only meaningful if the costs across different sub-columns are comparable. Using the default edge weight in SAND, defined as the absolute difference between the quantities connected by the edge, these costs are not comparable across different units.

We modify the edge weights so that the weight of an edge connecting two quantities u and v is defined using Bray-Curtis distance, as given in Equation 4.1. As discussed earlier, this new edge weight ensures the cost is normalized based on the scale of the quantities matched, making the mappings costs across different sub-columns and candidate types comparable.

4.3 Relaxing the Assumption on Unit Count

In this section, we relax the assumption from the previous section that the number of units in a query column is known. Detecting the number of units based solely on the distribution of values is not straightforward, especially when the ranges of values for different units overlap. Each choice of k yields a model of data with an associated cost. Our hypothesis posits that *a correct model should yield a better mapping of the*

query column and result in the least cost.

Based on this hypothesis, we vary the number of units k from one to a maximum and estimate the cost of the mapping under each value of k. For example, for $k = 2$, our query column is divided into two sub-columns using k-means clustering. With each sub-column assumed to contain only one unit, we proceed with the annotation process individually for each. This results in a final annotation with an associated cost value for each sub-column. The cumulative cost of the two sub-column mappings gives the cost for having two units. We expect the range of possible values for k to be small, and that the optimal number of partitions, where the cost value is minimized, signifies the ideal partitioning of the column into its constituent units.

Chapter 5

Evaluation

In this section, we evaluate our model's performance on diverse datasets with varying parameters, such as data reflectivity and unit counts. These datasets were generated specifically for testing purposes to ensure a comprehensive assessment. We begin by detailing our experimental setup, followed by an analysis of the model's accuracy in determining if a column contains mixed units. Next, we assess its ability to determine the number of units within mixed-unit columns. We then compare our model's performance against SAND, the state-of-the-art baseline, and the Kolmogorov Smirnov (KS) test which is used in a few related work[19, 20, 29]. Following this, we evaluate the model's effectiveness in annotating columns after they have been correctly clustered. We also examine the impact of using relative difference versus absolute distance on the model's performance and assess its robustness to changes in the number of units within a column. Finally, we investigate the impact of query column size on the model's accuracy.

5.1 Experimental Setup

To evaluate our model's performance, we utilize the following datasets:

- MixedSAND datasets: As detailed in Chapter 3, these are mixed-unit datasets derived from the Wikidata knowledge graph. We generated three variations (Easy, Medium, and Hard) to assess the model's robustness across different

levels of unit overlap and reflectivity.

- WDC dataset: This dataset is the same real-world, single-unit dataset employed in the evaluation of SAND [22]. Specifically, we utilize the same subset of the WDC table corpus that was used in the SAND evaluation, which contains 69 columns. This subset is particularly relevant for our purpose as it allows for direct comparison with the state-of-the-art method on a standardized benchmark.

By evaluating our model on both the mixed-unit MixedSAND datasets and the single-unit WDC subset, we can gain a comprehensive understanding of its capabilities across diverse scenarios.

5.2 Detecting the Number of Units

Single-unit vs. multi-unit columns. We first evaluate our model's ability to detect whether a column contains mixed units or not. To assess this, we use three datasets: the easy and hard datasets introduced in the previous section, which contain mixed-unit columns, and the subset of the WDC dataset utilized for evaluation in [22] which contain single-unit columns. The results of running the model on each dataset are presented in Table 5.1.

As shown in the results, the accuracy of identifying single-unit columns in the WDC dataset (which is a single-unit dataset) is higher than the accuracy of identifying mixed-unit columns in the hard dataset. This outcome aligns with our expectations, given that we utilized the hard dataset characterized by columns with high reflectivities, making it inherently more challenging to discern whether they contain mixed units. The complexity arises due to the substantial overlap of numbers in different units within these columns, contributing to a lower accuracy in recognizing them as mixed units compared to single units.

Given that a mixed-unit column contains data from different units (e.g., weight

in pounds and kilograms), and because of the differences in range and scale between units, the overall distribution of the column is likely to appear as if composed of two or more distinct distributions. As one can see in Figure 5.1, this phenomenon manifests as multimodality, where the distribution exhibits multiple peaks or modes. To provide context for our model's performance, we employed a common method for

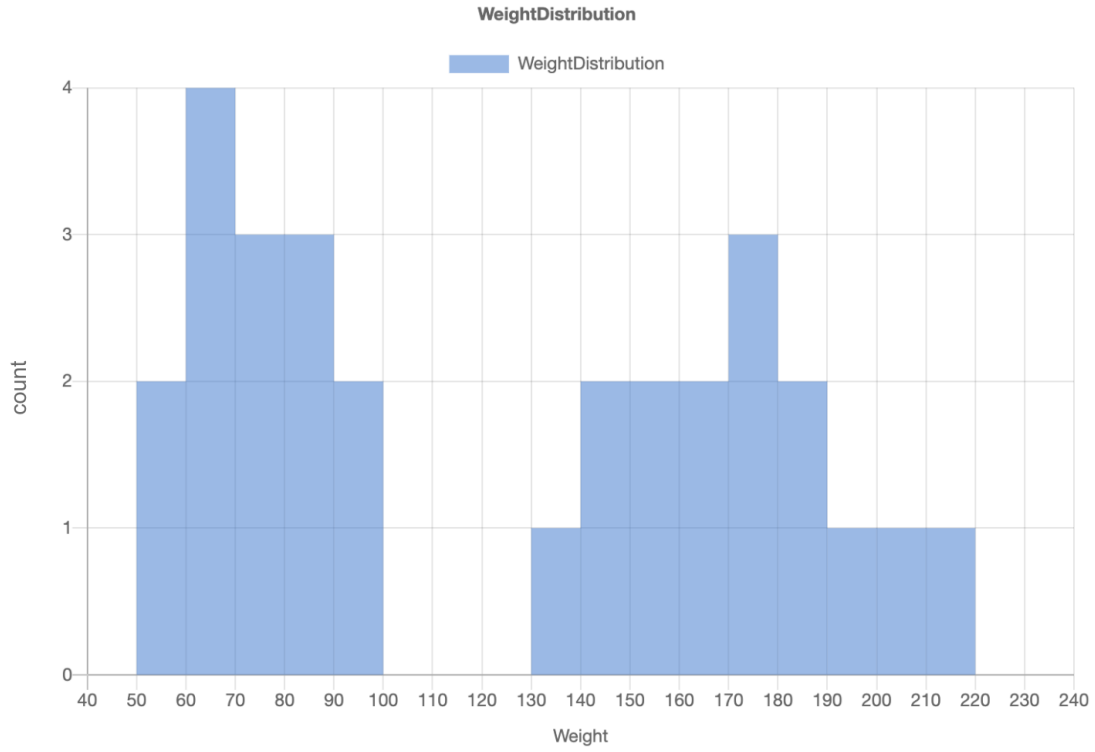

Figure 5.1: Distribution of a mixed-unit column

identifying multimodal distributions: Kernel Density Estimation (KDE)[44]. KDE is a non-parametric statistical technique that estimates the probability density function. We can use KDE to assess the potential multimodality of a column's distribution [45, 46], which is manifested by the number of "peaks" when plotting the data distribution. The presence of multiple distinct peaks in the KDE plot indicates a multimodal distribution, which suggests a mixed-unit column. Conversely, a smooth, unimodal distribution is more characteristic of a single-unit column. We anticipate that the KDE-based baseline will perform reasonably well on datasets with clear unit separa-

tion but might face challenges in scenarios with high reflectivity. In those cases, the overlap between different units could mask multimodality in the KDE plot. In fact, as can be seen in the Table 5.1, our model demonstrated superior performance to the KDE-based baseline, achieving a 27% higher accuracy on the hard dataset. This result highlights our method's ability to discern subtle unit patterns.

	Hard dataset	**Easy dataset**	**WDC**
Number of Columns	200	200	69
MixedSAND Accuracy (%)	0.58	0.765	0.696
KDE Accuracy (%)	0.50	0.55	0.652

Table 5.1: Accuracy of MixedSAND compared to a KDE-based baseline in detecting multi-unit columns (%)

Determining the number of units Next, we evaluate our model's ability to accurately count the number of distinct units within mixed-unit columns. To facilitate this evaluation, we constructed a dataset, which we refer to as the Varying Unit Count Dataset (VUCD), consisting of 300 distinct, randomly generated columns derived from the real-world dataset Wikidata. The dataset was designed to include an equal number (75) of columns containing 2, 3, 4, or 5 different units. The results are presented in Table 5.2.

	MixedSAND	**KDE**
Number of columns	300	300
Accuracy	**0.516**	**0.193**

Table 5.2: Performance of MixedSAND compared to KDE on determining the number of units in columns

Clearly, our method significantly outperforms the KDE approach. This is likely due to KDE's reliance on density estimation, which can become inaccurate when

distributions are not well-separated. Consequently, when data from different units overlap, the performance of KDE is expected to deteriorate.

5.3 Type Detection Performance

In this section, we present an end-to-end evaluation. We take a column without prior knowledge of whether it contains mixed units or not, input it into our model, and then evaluate the performance based on the predicted semantic types output by our model.

We have also compared MixedSAND with SAND [22] (the state-of-the-art baseline). Tables 5.3, 5.4 and 5.5 present the performance in terms of top-1, top-3, and top-5 accuracy, which measures the fraction of annotated columns for which the correct semantic type was returned in the top-n predictions.

The results indicate that our model outperforms SAND across the Easy, Medium, and Hard datasets. The performance gap between our model and the SAND model is more significant in the Easy dataset compared to Medium and Hard. This is because the Easy dataset primarily consists of low reflectivity data, meaning that data in different units have minimal overlap and may have large gaps. The SAND model aims to annotate all of the data at once, but it struggles to do so in the Easy dataset. However, in the Hard dataset, where there is significant overlap between data in different units, the SAND model performs well.

Our model demonstrates strong performance across all datasets, with accuracy slightly higher in the Easy and Medium datasets compared to the Hard dataset. This is because our model can more effectively recognize and separate distinct units when there is minimal overlap between values, as is the case in the Easy dataset. In contrast, the higher degree of overlap in the Hard and Medium datasets makes unit separation more challenging, leading to a slight decrease in accuracy compared to the Easy dataset.

We also evaluated our model against the SAND and KS-test on the WDC dataset,

	Method	Top-n Accuracy
n = 1	SAND	0.195
	MixedSAND	0.485
n = 3	SAND	0.36
	MixedSAND	0.75
n = 5	SAND	0.53
	MixedSAND	0.825

Table 5.3: Performance comparison of MixedSAND vs. SAND on semantic labeling of the Easy dataset

	Method	Top-n Accuracy
n = 1	SAND	0.22
	MixedSAND	0.335
n = 3	SAND	0.455
	MixedSAND	0.64
n = 5	SAND	0.65
	MixedSAND	0.80

Table 5.4: Performance comparison of MixedSAND vs. SAND on semantic labeling of the Medium dataset

which comprises single-unit columns. The results, presented in Table 5.6, demonstrate that our model performs comparably to SAND, the current state-of-the-art baseline, and outperforms KS-test even when tested on a single-unit dataset.

5.4 Stage 2 and 3 Evaluation

In this section, we evaluate our model's performance on a dataset where the columns have already been separated and clustered. Specifically, we focus on scenarios where we have multiple numeric columns with the same type and property but potentially different units. Given that these columns share the same type and property, our goal

	Method	Top-n Accuracy
n = 1	SAND	0.275
	MixedSAND	0.33
n = 3	SAND	0.455
	MixedSAND	0.60
n = 5	SAND	0.665
	MixedSAND	0.775

Table 5.5: Performance comparison of MixedSAND vs. SAND on semantic labeling of the Hard dataset

is to annotate them correctly, considering only the possibility of differing units. This evaluation allows us to assess the annotation capabilities of our model independently of the clustering step.

This evaluation focuses on the Annotating Sub-Models (Section 4.2.2) and Aggregating Sub-Model Annotations (Section 4.2.3). Annotating columns when the clustered are separated is highly relevant in real-world scenarios, particularly in table integration tasks such as table union or stitching. In these situations, it is common to encounter columns with the same type and property but with uncertain unit consistency. This section assesses how effectively our model handles these scenarios.

To perform this evaluation, we constructed a dataset consisting of 100 tables, each containing two columns. Both columns in each table correspond to the same type and property but have different units. This dataset was derived from real data within Wikitables, which includes 1.9 million tables extracted from Wikipedia. We systematically identified numeric columns representing the same property and manually annotated a subset to determine their type and unit. From this, we selected columns with the same type and property but different units to create the dataset used in this evaluation.

We applied our model to this dataset, and the results are summarized in Table

	Method	Top-n Accuracy
n = 1	KS-test	0.069
	SAND	0.116
	MixedSAND	0.101
n = 3	KS-test	0.129
	SAND	0.232
	MixedSAND	0.26
n = 5	KS-test	0.277
	SAND	0.42
	MixedSAND	0.405

Table 5.6: Performance comparison of MixedSAND vs. SAND and KS-test on semantic labeling of the WDC dataset

5.7. As expected, the accuracy of our model in this specific task is higher than its accuracy for mixed-unit and single-unit columns. This is predictable because, in the case of mixed-unit and single-unit columns, the model must first determine whether a column is mixed-unit or not before proceeding to the annotation step, introducing potential errors at each stage. Therefore, it is anticipated that the accuracy of this specific aspect of our model is superior to the overall model accuracy.

	Top-n Accuracy
n = 1	0.62
n = 3	0.76
n = 5	0.89

Table 5.7: Performance of the Stages 2 and 3 of Model on the Wikitables Dataset

5.5 Evaluating the Impact of Relative Difference

In this section, we examine the advantages of using relative difference (or Bray-Curtis distance) instead of absolute distance for two key tasks: determining the number of units within mixed-unit columns and distinguishing between single-unit and mixed-unit columns.

Determining the Number of Units. To illustrate the effectiveness of the relative difference function, we compared our model's performance using relative difference versus absolute distance on the VUCD dataset, which contains mixed-unit columns with varying numbers of units.

As shown in Table 5.8, our model consistently achieves superior results when employing relative difference. As argued earlier (Section 4.2.1), mixed-unit data involves values measured on different scales (e.g., kilograms vs. pounds), absolute differences can be misleading. In contrast, relative difference accounts for these scale variations, allowing for more accurate clustering and unit count determination.

	VUCD
MixedSAND using relative difference (%)	51.6
MixedSAND using absolute distance (%)	25
Number of Columns	300

Table 5.8: Impact of relative vs. absolute distance on MixedSAND's clustering accuracy

Differentiating Between Single-Unit and Mixed-Unit Columns To assess the impact of relative difference on the identification of mixed-unit columns, we compared our model's performance using relative difference versus absolute distance on the easy (mixed-unit) and WDC (single-unit) datasets.

Table 5.9 reveals a clear improvement in performance when using relative difference. This is attributed to the inherent limitations of absolute distance when dealing with mixed-unit data. Large absolute differences can occur between values belonging to the same unit if their magnitudes are high, leading to over-clustering. Conversely, small absolute differences between values from different units with low magnitudes can result in under-clustering. Relative difference mitigates these issues by considering the scale of the values, thus enhancing the model's ability to correctly distinguish between single-unit and mixed-unit columns.

	Easy dataset	**WDC**
Number of Columns	200	69
MixedSAND using relative difference	0.765	0.696
MixedSAND using absolute distance	0.665	0.493

Table 5.9: Accuracy of MixedSAND in distinguishing single-unit vs. mixed-unit columns using relative difference vs. absolute distance (%)

5.6 Assessing Robustness to Changes in the Number of Units

Lastly, we investigate the robustness of our method with respect to the number of units within a mixed-unit column. To achieve this, we needed datasets containing mixed-unit columns with varying numbers of units, while keeping other influential metrics, such as reflectivity, constant. To this end, we constructed four datasets, each containing columns with a specific number of units (ranging from two to five types, with a total of seven possible types). We assume there are no more than seven units in a single column, which justifies the selection of this upper limit. Each dataset has the following characteristics:

- It contains 100 columns, each with 40 rows.

- The reflectivity distribution of the columns matches that of the easy dataset described in the evaluation setup section, and the data is derived from the real-world dataset Wikidata.

The results of evaluating our model's performance across these four datasets are summarized in Table 5.10. It's important to note that with seven possible types, a random selection would yield an accuracy of approximately 0.14.

	Dataset with	Top-n Accuracy
n = 1	2-Units	0.49
	3-Units	0.41
	4-Units	0.41
	5-Units	0.42
n = 3	2-Units	0.74
	3-Units	0.69
	4-Units	0.70
	5-Units	0.70
n = 5	2-Units	0.82
	3-Units	0.80
	4-Units	0.80
	5-Units	0.81

Table 5.10: Column type annotation performance of MixedSAND on columns with varying numbers of units

As evidenced by Table 5.10, our model demonstrates robustness to changes in the number of units within a column. Notably, the performance remains consistent across datasets containing columns with 3, 4, or 5 units.

5.7 Impact of Query Column Size on the Model

In this section, we analyze the impact of query column size on the model's accuracy. For this evaluation, we required datasets where only the query column size varies while other parameters remain constant. To achieve this, we constructed a dataset of single-unit columns derived from real-world numeric data in Wikidata, with each column containing 50 entries. To assess our model's performance across different query column sizes, we used random samples from the query columns rather than the entire column.

Figure 5.2 presents the top-n accuracy, which initially shows that as the length of the query column increases, the model's accuracy improves. However, at a certain point, further increasing the query column length leads to a decrease in accuracy. While it might be expected that larger query columns would enhance model accuracy, this is not always the case. When the query column size exceeds the size of the candidate column, injection mapping becomes impossible. To address this, we sample the query column to match the candidate column's length whenever it surpasses the candidate column size. For the injection mapping algorithm to function correctly, the data in the query column must map entirely to the data in the candidate column. However, if the candidate column contains outlier data, this can lead to inaccurate predictions by the model.

In the single-unit dataset used in this section, the average size of the candidate columns is 43. As illustrated in Figure 5.2, the model's accuracy improves as the query column size increases up to 30. Beyond this point, as the query column size increases further, the accuracy begins to decrease.

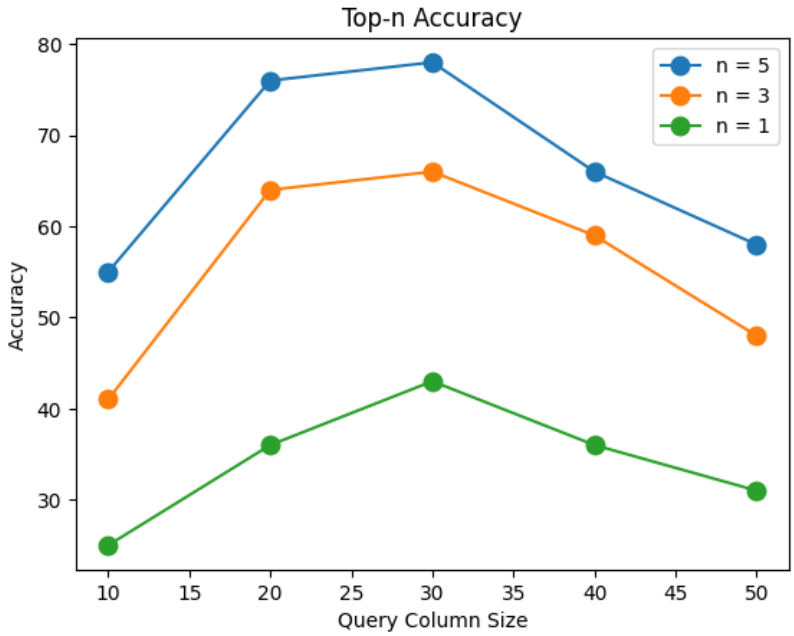

Figure 5.2: Model Accuracy with Varying Query Column Size

Chapter 6

Conclusion

6.1 Summary

In this thesis, we tackled the complex problem of annotating numerical columns with mixed units in tabular data—a challenge often encountered in data integration tasks such as query generation, data analysis, and decision-making processes. When tables from diverse sources are combined, inconsistent unit representations can significantly hinder the effective use of data.

To address this issue, we introduced a novel three-staged annotation pipeline that does not assume uniform unit representation across columns. Our approach includes: (1) generating plausible models for subsets of data, (2) assigning semantic types through a cost optimization framework, and (3) aggregating sub-model costs to determine the most accurate overall annotation. This method advances the state-of-the-art by incorporating a new benchmark for mixed-unit column annotation and leveraging the SAND model's capabilities in a novel way.

Our extensive experiments, including evaluations on our newly introduced mixed-unit benchmark and other datasets, demonstrate that our method outperforms existing approaches in accuracy. By effectively detecting and annotating mixed-unit numeric columns, our approach addresses a critical challenge in data integration and validation.

6.2 Future Work

While our approach has demonstrated significant effectiveness in annotating mixed-unit numerical columns, several promising avenues for future research could further advance its capabilities. One key direction involves improving the efficiency of our model by pre-determining whether a column is mixed-unit before starting the annotation process. This advance could reduce computational overhead and enhance the overall running time.

Another important area for exploration is the enhancement of annotation accuracy by considering multiple columns within the same table. By leveraging the interdependencies between columns, our method could better utilize the context provided by related columns to refine annotations. For example, if columns within a table are known to be related, incorporating this information could improve the precision of unit detection and semantic labeling.

Exploring integration with other data sources and knowledge graphs could also provide richer semantic context and improve annotation accuracy. Future research could investigate how combining multiple knowledge sources impacts the effectiveness of unit detection and type assignment.

In summary, our research establishes a strong foundation for the annotation of mixed-unit numerical data, offering insights that pave the way for further developments in data integration and analysis. By pursuing these future directions, we can enhance the applicability and robustness of our approach, ultimately contributing to more accurate and efficient data processing in diverse real-world scenarios.

References

[1] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International journal of information management*, vol. 35, no. 2, pp. 137–144, 2015.

[2] N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, *et al.*, "Linked open government data: Lessons from data. gov. uk," *IEEE Intelligent Systems*, vol. 27, no. 3, pp. 16–24, 2012.

[3] C. Madera and A. Laurent, "The next information architecture evolution: The data lake wave," in *Proceedings of the 8th international conference on management of digital ecosystems*, 2016, pp. 174–180.

[4] B. Ubaldi, "Open government data: Towards empirical analysis of open government data initiatives," 2013.

[5] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2002, pp. 233–246.

[6] A. Doan and A. Y. Halevy, "Semantic integration research in the database community: A brief survey," *AI magazine*, vol. 26, no. 1, pp. 83–83, 2005.

[7] X. L. Dong and D. Srivastava, "Big data integration," in *2013 IEEE 29th international conference on data engineering (ICDE)*, IEEE, 2013, pp. 1245–1248.

[8] A. Khatiwada, G Fan, R Shraga, and Z Chen, "Santos: Relationship-based semantic table union search," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2023, pp. 123–135.

[9] F. Nargesian, E. Zhu, K. Q. Pu, and R. J. Miller, "Table union search on open data," in *Proceedings of the VLDB Endowment*, vol. 11, 2018, pp. 813–825.

[10] O. Lehmberg and C. Bizer, "Stitching web tables for improving matching quality," *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1502–1513, 2017.

[11] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, *et al.*, "Knowledge graphs," *ACM Computing Surveys (Csur)*, vol. 54, no. 4, pp. 1–37, 2021.

[12] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs.," *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, no. 1-4, p. 2, 2016.

[13] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, *et al.*, "Dbpedia: A nucleus for a web of open data," in *international semantic web conference*, Springer, 2007, pp. 722–735.

[14] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[15] V. Efthymiou, O. Hassanzadeh, M. Rodriguez-Muro, and V. Christophides, "Matching web tables with knowledge base entities: From entity lookups to entity embeddings," in *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16*, Springer, 2017, pp. 260–277.

[16] G. Limaye, S. Sarawagi, and S. Chakrabarti, "Annotating and searching web tables using entities, types and relationships," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1338–1347, 2010.

[17] J. Chen, E. Jiménez-Ruiz, I. Horrocks, and C. Sutton, "Colnet: Embedding the semantics of web tables for column type prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 29–36.

[18] A. Alobaid and O. Corcho, "Fuzzy semantic labeling of semi-structured numerical datasets," in *Knowledge Engineering and Knowledge Management: 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings 21*, Springer, 2018, pp. 19–33.

[19] S. Neumaier, J. Umbrich, J. X. Parreira, and A. Polleres, "Multi-level semantic labelling of numerical values," in *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15*, Springer, 2016, pp. 428–445.

[20] M. Pham, S. Alse, C. A. Knoblock, and P. Szekely, "Semantic labeling: A domain-independent approach," in *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15*, Springer, 2016, pp. 446–462.

[21] S. K. Ramnandan, A. Mittal, C. A. Knoblock, and P. Szekely, "Assigning semantic labels to data sources," in *European semantic web conference*, Springer, 2015, pp. 403–417.

[22] Y. Su, D. Rafiei, and B. Khorram Nazari, "Sand: Semantic annotation of numeric data in web tables," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 2342–2351.

[23] Z. Zhang, "Effective and efficient semantic table interpretation using tableminer+," *Semantic Web*, vol. 8, no. 6, pp. 921–957, 2017.

[24] Z. Zhang, "Towards efficient and effective semantic table interpretation," in *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13*, Springer, 2014, pp. 487–502.

[25] K. Nishida, K. Sadamitsu, R. Higashinaka, and Y. Matsuo, "Understanding the semantic structures of tables with a hybrid deep neural network architecture," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[26] K. Takeoka, M. Oyamada, S. Nakadai, and T. Okadome, "Meimei: An efficient probabilistic approach for semantically annotating tables," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 281–288.

[27] D. Zhang, Y. Suhara, J. Li, M. Hulsebos, *et al.*, "Sato: Contextual semantic type detection in tables," *arXiv preprint arXiv:1911.06311*, 2019.

[28] P. Nguyen, K. Nguyen, R. Ichise, and H. Takeda, "Embnum+: Effective, efficient, and robust semantic labeling for numerical values," *New Generation Computing*, vol. 37, pp. 393–427, 2019.

[29] E. Kacprzak, J. M. Giménez-García, A. Piscopo, L. Koesten, *et al.*, "Making sense of numerical data-semantic labelling of web tables," in *Knowledge Engineering and Knowledge Management: 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings 21*, Springer, 2018, pp. 163–178.

[30] V. T. Ho, K. Pal, S. Razniewski, K. Berberich, *et al.*, "Extracting contextualized quantity facts from web tables," in *Proceedings of the Web Conference 2021*, 2021, pp. 4033–4042.

[31] B. Kruit, P. Boncz, and J. Urbani, "Takco: A platform for extracting novel facts from tables," in *Companion Proceedings of the Web Conference 2021*, 2021, pp. 705–707.

[32] Y. Ibrahim, M. Riedewald, G. Weikum, and D. Zeinalipour-Yazti, "Bridging quantities in tables and text," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, 2019, pp. 1010–1021.

[33] A. D. Nobari and D. Rafiei, "Efficiently transforming tables for joinability," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, IEEE, 2022, pp. 1649–1661.

[34] K. Narisawa, Y. Watanabe, J. Mizuno, N. Okazaki, *et al.*, "Is a 204 cm man tall or small? acquisition of numerical common sense from the web," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 382–391.

[35] M. Cannaviccio, D. Barbosa, and P. Merialdo, "Towards annotating relational data on the web with language models," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1307–1316.

[36] D. T. Bollegala, Y. Matsuo, and M. Ishizuka, "Relational duality: Unsupervised extraction of semantic relations between entities on the web," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 151–160.

[37] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier, "Connecting language and knowledge bases with embedding models for relation extraction," *arXiv preprint arXiv:1307.7973*, 2013.

[38] ., *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching*, https://www.cs.ox.ac.uk/isg/challenges/sem-tab/, [Online; accessed 20-May-2024], 2019-2024.

[39] V. Cutrona, J. Chen, V. Efthymiou, O. Hassanzadeh, *et al.*, "Results of semtab 2021," *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching*, vol. 3103, pp. 1–12, 2022.

[40] R. Agrawal and R. Srikant, "Searching with numbers," in *Proceedings of the 11th international conference on World Wide Web*, 2002, pp. 420–431.

[41] J. B. Orlin, "A polynomial time primal network simplex algorithm for minimum cost flows," *Mathematical Programming*, vol. 78, pp. 109–129, 1997.

[42] A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by successive approximation," *Mathematics of Operations Research*, vol. 15, no. 3, pp. 430–466, 1990.

[43] C. Ricotta and J. Podani, "On some properties of the bray-curtis dissimilarity and their ecological meaning," *Ecological Complexity*, vol. 31, pp. 201–205, 2017.

[44] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall, 1986.

[45] J. Ameijeiras-Alonso, R. M. Crujeiras, and A. Rodriguez-Casal, "Multimode: An r package for mode assessment," *arXiv preprint arXiv:1803.00472*, 2018.

[46] B. W. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 43, no. 1, pp. 97–99, 1981.