Scientific Research Publishing

Journal of Computer and Communications, 2026, 14(7), 221-241
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Study of Error Detection Methods for Tabular Data with a Small Number of Labelled Instances

Wenjia Tian[1,2], Jinwen Shao[1]

[1]Beijing Academy of Science and Technology, Beijing, China
[2]Institute of Urban Safety and Environmental Science, Beijing Academy of Science and Technology, Beijing, China
Email: 297282966@qq.com

How to cite this paper: Tian, W.J. and Shao, J.W. (2026) A Study of Error Detection Methods for Tabular Data with a Small Number of Labelled Instances. *Journal of Computer and Communications*, **14**, 221-241.
https://doi.org/10.4236/jcc.2026.147011

Received: June 30, 2026
Accepted: July 27, 2026
Published: July 30, 2026

Abstract

The massive data of the information age raises stricter standards for data quality. Conventional data error detection approaches depend on handcrafted rules or abundant expert labels, limiting their performance under scarce annotation conditions. To tackle this issue, we present JCED (Joint Contextual Error Detector), a low-label error detection model for tabular data optimized end-to-end via four core components: 1) a distance correlation-based logical grouping module to capture nonlinear column correlations; 2) a joint representation module integrating context-agnostic word2vec and context-aware Sentence Transformer for better semantic context modeling; 3) a two-stage hard sample pair sampling module combining intra-cluster compactness and BCE-Rerank to select informative samples; 4) an inverse propensity score weighting classifier to alleviate sampling bias and boost generalization. Evaluations on five real-world datasets reveal that with merely 5 - 25 labeled samples, JCED outperforms state-of-the-art methods (Raha, HoloDetect, etc.) by 140% in average F1-score and 13.76% in detection efficiency. Ablation studies verify the complementary gains of all modules, and JCED exhibits superior robustness under dynamic error rates ranging from 5% to 30%. This research offers a novel technical solution for low-resource data governance.

Keywords

Data Governance, Data Quality, Error Detection, Tabular Data, Low Labelling Learning

1. Introduction

With the development of the Internet, massive data has been accumulated, and its analytical value can be tapped through data processing. As a basic and challenging

step in relational data tasks such as entity matching and data integration, data cleaning directly determines data quality. Data errors will mislead downstream decisions and cause huge economic losses to enterprises [1]-[4]. Real-world datasets contain roughly 5% erroneous records [5] [6], driving widespread academic and industrial research on data cleaning, which is split into error detection (ED) [7]-[15] and error correction (EC) [16] [17].

This work targets tabular data error detection. State-of-the-art methods suffer three critical defects under low-label scenarios: they rely on manual settings and cannot automatically model nonlinear column dependencies; they only adopt context-independent embedding; and biased sampling distorts data distribution and impairs generalization. The specific defects are as follows. First, table attributes usually have nonlinear logical correlations. Existing methods capture them through manually defined rules and parameters, which require a lot domain labor [7] [8]. Therefore, this paper designs a dependency modeling module to automatically quantify column correlations. Second, mainstream methods convert table text into embeddings for sample screening [7] [8] [15]. Most of them only use context-independent Word2Vec, which ignores context semantic changes, leading to suboptimal performance of context-aware tabular data [15]. This paper combines Word2Vec and context-sensitive Sentence Transformer to obtain joint semantic representation. Third, existing sampling strategies give priority to high-frequency and high-information samples, and rarely select low-frequency samples, resulting in serious sampling bias. The model overfits frequent error patterns and cannot be generalized to rare error types when labels are limited. Therefore, this paper studies a sampling bias elimination scheme. Fourth, traditional low-label detection methods have obvious limitations. HoloDetect requires 5% - 10% manual labels through data augmentation; Raha [7] uses clustering and label propagation, but will produce noisy labels, and its accuracy and recall are unstable. Based on the above situation, this paper proposes a tabular error detection model JCED, which can maintain a high F1 score when annotations are scarce.

To tackle the above limitations, this paper puts forward Joint Contextual Error Detector (JCED), an end-to-end tabular error detection model with four core modules:

1) Distance Correlation Coefficient (DCOR) based logical grouping module: DCOR captures both linear and nonlinear attribute correlations to automatically group columns and depict comprehensive dependency structures without manual configuration.

2) Multi-model joint representation module: Fuses context-agnostic Word2Vec and context-aware Sentence Transformer to overcome the single embedding's deficiency in modeling contextual semantics.

3) Two-stage intra-cluster hard pair sampling module: Stage one clusters embeddings and selects optimal clusters via scoring functions; stage two mines informative normal-error sample pairs and outliers within target clusters to fully exploit limited labeled data.

4) Inverse Propensity Score (IPS) weighted binary classifier: Eliminates sampling bias induced by unbalanced sample selection and stabilizes model generalization.

2. Related Work

This subsection reviews state-of-the-art error detection (ED) approaches.

ED is a core task for data quality optimization and has been widely researched [10]. Existing methods attain average F1 scores above 0.85 on real-world datasets, yet their performance degrades drastically under scarce labeled samples [7]-[9] [11]. Industrial ED tools including Trifacta [18] and Tamr [19] have been commercialized, but nearly all mainstream ED techniques demand massive expert annotations to maintain high F1 values.

Conventional ED falls into two categories: quantitative methods [20]-[22] that detect anomalies via data statistical distributions, and qualitative methods [23] [24] relying on handcrafted rules, patterns or external knowledge [25]-[28]. Limited by heterogeneous mixed error types in relational tables, both families suffer low recall.

Recently, machine learning and deep learning-based ED methods have attracted wide interest [4] [7] [8] [29]-[31]. HoloDetect [8] and Raha [7] represent current SOTA models. HoloDetect mines data semantic and syntactic features with pre-defined consistency rules to locate errors. Raha runs multiple ED algorithms to generate feature encodings for error modeling and delivers comparable accuracy, but its multi-algorithm configuration pipeline is extremely time-intensive.

Different from prior work, the proposed JCED model removes the reliance on multiple detection algorithms and manual pre-configuration. It adopts distance correlation-based logical grouping, and fuses context-independent and context-aware embeddings in a joint representation module to capture contextual semantics. Equipped with a two-stage hard-sample sampling strategy and inverse propensity score weighted binary classifier, JCED achieves superior F1 and efficiency with only a small number of labeled records, as verified in Section 5.3.

3. Problem Statement

This section outlines the error detection problem and discusses how the JCED model addresses it.

This paper defines cases where the cell value $D[i,j]$ in a spreadsheet differs from its corresponding true value $D*[i,j]$ as negative classes, and cases where they match as positive classes. The goal of error detection is to assign the most probable positive or negative class label to each cell value $D[i,j]$ within the given dataset D. A cell value is considered correctly classified if it is assigned to the positive class when $D[i,j]=D*[i,j]$, or to the negative class when $D[i,j]\neq D*[i,j]$; conversely, it is deemed incorrectly classified if assigned to the negative class when $D[i,j]=D*[i,j]$, or to the positive class when $D[i,j]\neq D*[i,j]$. Specifically, the study aims to develop a sampling strategy that identifies the most promising training samples to maximize the F1-score. The precise definition and calculation

methodology of the F1-score metric are detailed in Section 5.2.

However, as noted in the introduction, manual labeling for model training is typically highly labor-intensive, and our research objective is to achieve high error detection accuracy using a small number of labeled instances. Accordingly, this paper employs the JCED model for error detection tasks to enhance performance while reducing annotation costs.

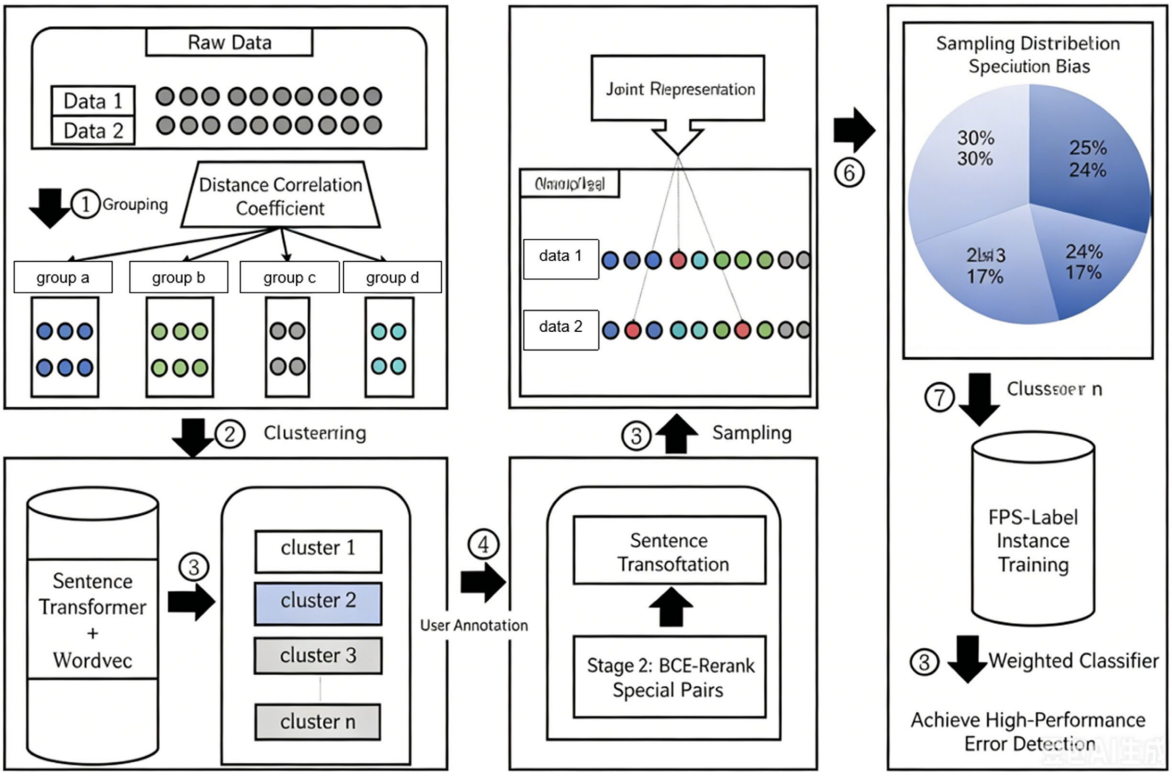

Figure 1. Flowchart of the JCED model.

Figure 1 shows the framework of JCED, which consists of a grouping module, a joint representation module, a special case pair sampling module, and an inverse propensity score correction module.

4. JCED Model

4.1. Grouping Module

Merging datasets with unrelated columns injects extraneous embedded signals. Since such appended information weakly correlates with the core dataset, embedding vectors suffer severe degradation in representation capacity. This underscores the necessity of algorithms for screening high-information features. Widely adopted feature selection techniques cover mutual information (MI) [32], chi-square test, Pearson correlation coefficient, and Random Forest feature importance scoring. For the grouping module, this work proposes a DCOR-based column selection strategy dubbed CS. It leverages the distance correlation coefficient (DCOR) to quantify pairwise column dependencies and filter out redundant

columns, accelerating data query efficiency. Concretely, CS ranks columns by their DCOR against the target column m: a higher DCOR for column n denotes a tighter inherent linkage to m, a prerequisite for robust event detection (ED) analysis.

The first step in the grouping module is to construct a distance matrix: initially, for two sets of samples $X=(x_1,\ldots,x_n)$ and $Y=(y_1,\ldots,y_n)$, Calculate the Euclidean distance matrices between them separately.

$$a_{ij}=\|x_i-x_j\|, b_{ij}=\|y_i-y_j\| \tag{1}$$

Subsequently, double centering is applied to the distance matrix: each matrix entry is subtracted by its corresponding row mean and column mean, followed by addition of the global matrix mean. This operation eliminates bias within the distance matrix, as formalized in the formula below.

$$A_{ij}=a_{ij}-a_i-a_j+a \tag{2}$$

$$B_{ij}=b_{ij}-b_i-b_j+b \tag{3}$$

They denote a_i, a_j, a_{ij}, the row mean, column mean and global mean, respectively. We further compute distance covariance and distance variance based on the double-centered distance matrix. Distance covariance acts as a core metric for measuring the similarity of distance distributions between two vectors, while distance variance characterizes the internal distance variation within a single vector. The calculation formulas are given below:

$$\mathrm{dCov}^2(X,Y)=\frac{1}{n^2}\sum_{i,j=1}^{n}A_{ij}B_{ij} \tag{4}$$

$$\mathrm{dVar}^2(X)=\frac{1}{n^2}\sum_{i,j=1}^{n}A_{ij}^2,\mathrm{dVar}^2(Y)=\frac{1}{n^2}\sum_{i,j=1}^{n}B_{ij}^2 \tag{5}$$

Finally, the distance correlation coefficient is computed from the distance covariance $\mathrm{dCor}(X,Y)$ and distance variance. As a standardized metric for measuring dependence between two random vectors, it takes values in \([0, 1]\): a value of 0 signifies full independence, while 1 corresponds to a deterministic functional relationship.

$$\mathrm{dCor}(X,Y)=\frac{\mathrm{dCov}(X,Y)}{\sqrt{\mathrm{dVar}(X)\mathrm{dVar}(Y)}} \tag{6}$$

Using the dCor metric, the DS strategy selects columns $a\in\mathcal{F}$ for each contaminated column a_k where $\mathcal{F}(a_k)$ exceeds the threshold α, thereby forming the set of relevant columns S_t as follow.

$$S_t=a\mid\mathcal{F}(a_k;a)\geq\alpha \tag{7}$$

In this work, α is set to 0.5 to strike a balance between adequate coverage of relevant columns and high information density. This configuration preserves critical features while mitigating information loss [33]. Under this strategy, columns with large dCor values that exhibit the strongest relevance are selected to form individual data clusters. Such a design guarantees strong internal correlations

within each cluster, enabling the derived embedding model to fully capture contextual semantic information.

4.2. Joint Representation Module

Driven by the widespread adoption of computing systems, natural language processing (NLP) has attracted extensive research interest. Practical demands spanning machine translation, speech recognition, and information retrieval have raised ever-stricter standards for the NLP capabilities of computing devices. Linguistic modeling constitutes an indispensable prerequisite for machines to comprehend natural language. NLP modeling paradigms have evolved from handcrafted rule-based systems to statistical approaches, known as statistical language models (SLMs). Representative SLM techniques include n-gram models, neural networks, and log-linear models. Nevertheless, SLM training is plagued by persistent obstacles: high dimensionality, word similarity modeling, weak generalization ability, and performance degradation. Resolving these bottlenecks has consistently fueled the evolution of statistical language models.

Within the landscape of SLM research, Google released Word2Vec in 2013 as an efficient word embedding training framework. Given a raw text corpus, Word2Vec leverages optimized training objectives to map discrete words into dense vector space, offering a foundational tool for downstream NLP research. It learns static word embeddings via two core architectures: Skip-Gram and Continuous Bag-of-Words (CBOW). Though Word2Vec produces context-invariant word representations—each word retains a fixed vector regardless of surrounding context—it achieves competitive performance across diverse NLP benchmarks.

Sentence Transformers, built upon recurrent neural network (RNN) foundations, generate fixed-length dense embeddings for full sentences and support direct text similarity computation. Its word encoding pipeline adopts a canonical encoder-decoder framework: the encoder compresses source text into context vectors, while the decoder reconstructs target representations from these latent codes. This architecture inherently introduces an information bottleneck between encoding and decoding stages. The model converts every lexical token into a uniform-dimension embedding that encapsulates both lexical semantics and contextual cues, which can be readily deployed on tasks such as text classification, sentiment analysis, and machine translation.

Context-independent embeddings deliver stable baseline features, whereas context-aware embeddings capture fine-grained context-specific semantic nuances. This work integrates both embedding paradigms to construct a joint representation module. The module inherits the efficiency and universal applicability of static embeddings while harnessing the strong context-sensitive semantic modeling capacity of dynamic embeddings.

The joint representation module first preprocesses tabular data. Suppose Table X contains m rows and n columns, among which the first k columns store textual content and the remaining n-k columns hold numerical values. The detailed pro-

cessing workflow is specified as follows:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} & x_{1,k+1} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,k} & x_{2,k+1} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,k} & x_{m,k+1} & \cdots & x_{m,n} \end{bmatrix} \tag{8}$$

Here, $x_{i,j}$ represents the data in row i and column j.

Next, use Sentence Transformers for embedding to extract text columns from the table data.:

$$T = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,k} \end{bmatrix} \tag{9}$$

Then use Sentence Transformers to encode the text column.

$$E_{T1} = \text{SentenceTransformer}(T) \in R^{m \times d_t}$$

Here, dt represents the output dimension of Sentence Transformer. The text columns are then encoded using Word2Vec.:

$$E_{T2} = \text{WordVec}(T) \in R^{m \times d_w} \tag{10}$$

Which d_w is Output Dimension of word2vec.

Merge the embeddings from Sentence Transformer and Word2Vec:

$$E_T = [E_{T1}; E_{T2}] \in R^{m \times (d_t + d_w)} \tag{11}$$

Extract Number Column:

$$E_N = \begin{bmatrix} x_{1,k+1} & \cdots & x_{1,n} \\ x_{2,k+1} & \cdots & x_{2,n} \\ \vdots & \ddots & \vdots \\ x_{m,k+1} & \cdots & x_{m,n} \end{bmatrix} \in R^{m \times (n-k)} \tag{12}$$

Union embedding of text columns and feature splicing of numerical columns,

$$E_T = [E_{T1}; E_{T2}] \in R^{m \times (d_t + d_w)} \tag{13}$$

Define the Query, Key, and Value matrices:

$$Q = E, K = E, V = E \in R^{m \times (d_t + d_w + (n-k))} \tag{14}$$

Calculate Attention Score:

$$\text{Scores} = \frac{QK^T}{\sqrt{d_k}} \tag{15}$$

Here, d_k represents the dimensionality of the key, typically given by $d_k = d_t + d_w + n - k$. The Softmax function is applied to obtain the attention weight distribution:

$$\alpha = \text{softmax}(\text{Scores}) \in R^{m \times m} \tag{16}$$

Finally, calculate the weighted sum to obtain the output:

$$\text{Output} = \alpha V \in R^{m\times(d_t+d_w+(n-k))} \tag{17}$$

4.3. Sampling Module

JCED is primarily designed to sustain high F1 scores under scarce labeled samples. To address this goal, we propose a two-stage instance pair sampling strategy, which makes full use of limited labeled data and accurately extracts the most representative sample pairs within each cluster. This two-stage sampling scheme extends the work described in Section 4.2. Briefly, the joint representation module converts clustered tabular data into embedding vectors, followed by DBSCAN clustering to partition the embeddings into n clusters. Given k labeled instances, we conduct k/2 sampling iterations. The overall strategy comprises two successive stages: cluster grouping sampling and discriminative special pair sampling.

Stage 1 Grouping Sampling: We first calculate the score of each cluster via the cluster scoring function and select the cluster with the maximum score among all n clusters, as formulated below:

$$S(C) = \alpha \cdot \frac{n_a}{n} + (1-\alpha)\cdot(1-\text{WCSS}(C)) \tag{18}$$

n_a indicates the number of labeled data points in this cluster， WCSS represents the within-cluster sum of squares, used to measure the compactness or variability of data points within a cluster..

The cluster score function evaluates the quality of each cluster based on a combination of the aforementioned three variables. To assess the homogeneity of data sets, this study introduces the concept of within-cluster sum of squares (WCSS) into the cluster score function [34]. WCSS effectively measures the variability or compactness of a data set by representing the sum of squared deviations between each data point and the cluster center. Specifically, the calculation formula for WCSS is as follows::

$$\text{WCSS} = \sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij}-c_j)^2 \tag{19}$$

where: n denotes the number of data points within a cluster, m represents the number of features, x_{ij} denotes the feature j value of the data point i, and c_j denotes the feature j value of the cluster center..

The cluster scoring function comprehensively considers cluster size, the number of labeled data points, and the compactness of data points within a cluster, assigning a score to each cluster. A higher score indicates greater representativeness and makes a cluster more suitable for selection as a sampling target.

Stage 2 sampling: The second stage of sampling involves identifying the most relevant data points and outliers using the BCE-Rerank model. This model identifies both the most relevant data points among all entries in the highest-scoring cluster and the outlier data within that cluster. To begin, similarity scores are calculated for each data point x_i by applying the BCE-Rerank model to compute mu-

tual similarity scores $s\left(x_i, x_j\right)$ between data points. The specific formula is as follows:

$$s\left(x_i, x_j\right) = \text{bce-rerank}_{\text{model}}\left(x_i, x_j\right) \tag{20}$$

Subsequently, we reorder all data points in descending order according to their similarity scores. Data points with higher similarity scores correspond to higher data relevance, whereas those with the lowest scores are regarded as outliers. We then pair the most relevant data point with the outlier to construct a special sample pair. This completes a single sampling iteration and yields one paired data sample.

Furthermore, the introduction of the BCE-Rerank model substantially boosts data retrieval performance. The proposed two-stage sampling strategy effectively screens highly informative tabular data, which further improves the accuracy of error detection.

4.4. Classifier Module

The Inverse Preference Score (IPS) was first proposed by Cornell University in a 2016 ICML conference paper [35]. As a pioneering method for mitigating bias in recommendation systems, IPS is designed to eliminate inherent selection bias. A fundamental bottleneck of recommendation systems lies in sparse observational data: only a small fraction of user-item interactions are observable, while most potential interactions remain unrecorded. Such incomplete data severely degrades the predictive performance and estimation accuracy of recommendation models. To tackle this problem, IPS estimates implicit user preferences for items and adaptively calibrates sample weights, enabling unbiased model evaluation even on biased and incomplete datasets.

Similar selection bias issues also prevail in data error detection tasks. Such bias not only restricts the generalization capability of detection algorithms but also undermines their performance in identifying specific error types. When trained on imbalanced datasets, conventional models tend to prioritize dominant categories during optimization, thereby failing to fully capture feature information from minority samples. To improve the fairness and overall detection accuracy of the model, targeted optimization strategies are essential to alleviate selection bias.

Accordingly, this paper introduces the IPS mechanism into error detection algorithms to address the aforementioned bias problem. We first calculate the propensity score of each sample and construct a corresponding weight matrix, which is subsequently integrated into the standard cross-entropy loss function of the binary classifier. This yields an IPS-enhanced binary classification model. The improved model reduces over-reliance on samples from prevalent clusters and strengthens feature learning from underrepresented clusters, achieving balanced and comprehensive error detection performance. For rigorous theoretical verification, we further derive the expected loss function of the proposed model. We assume that the true label y_i and prediction probability p_i of each sample follow

an independent and identical distribution, and each sample possesses an observational probability $p(x_i)$.

Under ideal balanced sampling conditions, the expected value of the standard cross-entropy loss function for a binary classifier is:

$$E[L] = E\left[-\sum_{i=1}^{N}\left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)\right] \tag{21}$$

Since y_i and p_i are independent and identically distributed, the expectation can be decomposed as:

$$E[L] = -\sum_{i=1}^{N} E\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{22}$$

When sampling data exhibits selection bias, the observation probability $p(x_i)$ for each sample is not uniformly distributed. Assuming the observation probability of sample x_i is $p(x_i)$, the expected loss function for the entire dataset is given by:

$$E[L_{biased}] = E\left[-\sum_{i=1}^{N}\left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)\right] \tag{23}$$

However, since the observation probability $p(x_i)$ is not uniform, the actual number of observed samples may be biased. Therefore, the expected loss function for the entire dataset can be expressed as:

$$E[L_{biased}] = -\sum_{i=1}^{N} p(x_i) E\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{24}$$

To demonstrate that the expected sum of the loss function under biased conditions differs from that under equilibrium conditions, we can compare them as follows:

$$E[L_{biased}] = -\sum_{i=1}^{N} p(x_i) E\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{25}$$

$$E[L] = -\sum_{i=1}^{N} E\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{26}$$

Obviously, unless $p(x_i) = \frac{1}{N}$ for all i, then $E[L_{biased}] \neq E[L]$.

To eliminate bias, this paper employs IPS (Importance Sampling) weights $w(x_i) = \frac{1}{p(x_i)}$. The expected loss function for the weighted dataset is as follows:

$$E[L_{biased}] = E\left[-\sum_{i=1}^{N} p(x_i) \frac{1}{p(x_i)}\left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)\right] \tag{27}$$

After simplification:

$$E[L_{biased}] = E\left[-\sum_{i=1}^{N}\left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)\right] \tag{28}$$

it is $E[L]$ precisely the expected value of the standard cross-entropy loss function.

Based on the above analysis, we draw the following conclusion: when selection bias exists, the expected loss function $E[L_{biased}]$ for the entire dataset differs from the standard cross-entropy loss function $E[L]$. After transformation, the expected loss function $E[L_{biased}]$ derived from all data weighted by IPS scores becomes identical to the standard cross-entropy loss function $E[L]$.

In other words, by introducing appropriate weights (such as IPS weights), the impact of selection bias can be eliminated, aligning the weighted expected loss function with the standard expected loss function. This resolves the sampling bias problem in error detection algorithms caused by prioritizing labeled instances with high information content, allowing the weighted corrected binary classifier to learn each error pattern more uniformly.

5. Experimental Evaluation

This section presents a comprehensive experimental evaluation of the proposed JCED model against multiple state-of-the-art baseline methods. With elaborately designed experiments, this study primarily answers the following research questions:

1) Which feature selection method can best boost the overall performance of JCED?

2) Which clustering algorithm contributes most significantly to the performance improvement of JCED?

3) Which sampling strategy yields optimal performance for the JCED framework?

4) Under limited labeling budgets, how does JCED outperform baseline algorithms in detection performance and computational efficiency?

5) With the increase of data error rates, how does JCED perform in terms of detection accuracy and efficiency compared with baseline methods?

Experimental analyses in this section enable a thorough investigation of the core advantages and practical capabilities of JCED. Meanwhile, the empirical results verify the model's effectiveness and comparative superiority over conventional error detection baselines.

The remainder of this section is organized as follows. Section 5.1 introduces the experimental datasets, experimental setup, and evaluation metrics. Section 5.2 elaborates on the experimental implementation, presents quantitative and qualitative results, and provides in-depth result analysis. Section 5.3 concludes the key experimental findings and summarizes core insights.

5.1. Experimental Data

This study comprehensively evaluated JCED using five real-world datasets (hospital [7], flight [36], beer consumption [37], Rayyan [38], and movie [39]), which have been widely utilized in the relevant literature [7] [8] [15].

All five datasets come with a ground-truth version that labels each cell as erroneous or correct, providing the labels used for training and evaluation. For the

error-rate experiment (**Figure 7**), we controlled the error rate by rule-based random error injection: for each dataset we reproduced its own original error types (typos, missing values, format and value-domain violations) and randomly perturbed cells according to these rules until each target rate (5%, 10%, 15%, 20%, and 30%) was reached.

5.2. Experimental Design and Evaluation Criteria

This paper conducts a comprehensive comparison between JCED and various baseline methods, including machine learning-based approaches such as RAHA [7] and ED2 [11]; rule-based techniques like HoloDetect [8]; knowledge base-driven methods such as KATARA [25]; and ensemble methods such as dBoost [40] and min-K [10].

The selected baseline methods represent state-of-the-art comprehensive solutions for data error detection. Each method integrates distinct techniques and algorithms to address detection tasks with unique optimization paradigms.

This study adopts precision, recall, and F1-score as core evaluation metrics to quantitatively assess the detection performance of the proposed JCED model, and further utilizes running time to measure its computational efficiency.

Precision denotes the proportion of true positive (TP) samples among all samples predicted as positive, including both true positives and false positives (FP). It is calculated as follows:

$$\text{Precision} = \text{TP} / \left(\text{TP} + \text{FP}\right)$$

Recall quantifies the proportion of correctly identified true positive samples relative to all actual positive samples, covering all positive cases excluding false negatives (FN). Its mathematical formula is defined as:

$$\text{Recall} = \text{TP} / \left(\text{TP} + \text{FN}\right)$$

Precision and recall present an inherent trade-off in practical model optimization: improving one metric usually leads to a decline in the other. To achieve balanced and comprehensive detection performance, the F1-score is adopted as a synthetic evaluation indicator. As the harmonic mean of precision and recall, the F1-score integrates the advantages of the two metrics and delivers a holistic assessment of classifier performance. A higher F1-score indicates superior overall detection capability. The standard F1-score formula is given below:

$$\text{F1} - \text{score} = 2\text{TP} / \left(2\text{TP} + \text{FP} + \text{FN}\right)$$

In addition to predictive performance, running time is introduced as an auxiliary evaluation metric to evaluate the computational efficiency of all compared methods.

To guarantee experimental reliability, all experiments are repeated ten times, with the average results taken as the final evaluation data. All experiments are conducted on a server equipped with a 16-core 2.60 GHz CPU and 32 GB RAM, running the Ubuntu 20.04 LTS system. For fair comparison, JCED selects 5 - 25

labeled instances via the sampling module introduced in Section 4.2, while all baseline methods adopt their inherent sampling strategies to sample labeled data under identical labeling budget constraints.

5.3. Experimental Analysis

Prior to analyzing the performance discrepancy between JCED and baseline models, this section first conducts ablation evaluations on the feature selection, clustering, and sampling strategies adopted in JCED. Further experiments are performed to assess model performance under varying outlier ratios.

Feature Selection Methods. The logical grouping module integrates and evaluates multiple feature selection approaches. **Figure 2** presents the comparison results across five datasets, where the x-axis denotes the number of labeled instances and the y-axis corresponds to the F1-score. The compared methods include the distance correlation coefficient (DCOR), Pearson correlation coefficient, mutual information (MI), and Random Forest feature importance scoring.

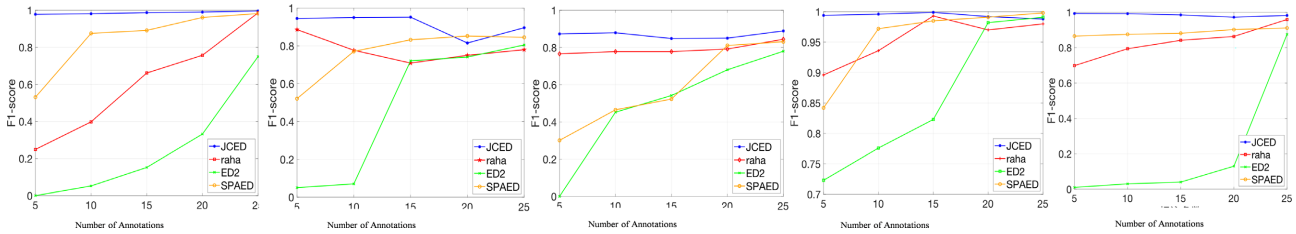

Figure 2. Comparison of experimental results for feature selection methods.

Experimental results verify that DCOR surpasses the Pearson, MI, and Random Forest methods by average performance margins of 8.63%, 2.53%, and 10.52%, respectively. Across all datasets, DCOR yields the slowest and most stable performance growth as the number of labeled instances increases from 5 to 25. Taking the Flight dataset as an example, the F1-score of DCOR increases by merely 7.6% (from 0.792 to 0.852) within this labeling range, which is considerably lower than the 12.3% growth of the Pearson method (0.723 - 0.812) and the 7.4% growth of the Random Forest method (0.741 - 0.796). This indicates that DCOR delivers competitive performance with extremely limited labeled data and is insensitive to variations in sample volume, which validates its practical applicability for low-label-resource error detection scenarios.

Clustering Methods. The joint representation module is adopted to evaluate multiple clustering algorithms. **Figure 3** presents comparative experimental results on five datasets, where the x-axis denotes the number of labeled instances and the y-axis corresponds to F1-scores. The compared algorithms include K-Means, DBSCAN, MeanShift, BP neural network, and spectral clustering.

Experiments under different labeling budgets demonstrate that DBSCAN achieves optimal overall performance on most datasets. It outperforms MeanShift, K-Means, and the BP neural network by average gains of 3.3%, 2.2%, and 4.0%, respectively. The prominent performance of DBSCAN stems from its density-

based clustering mechanism, which can automatically identify cluster centers and boundaries and exhibits strong robustness against noise samples. In contrast, K-Means requires a pre-defined cluster number and cannot effectively capture complex data structures. MeanShift suffers from performance degradation on datasets with intricate distributions. Spectral clustering is highly susceptible to similarity metrics, where different measurement strategies lead to unstable results. Although BP neural networks possess powerful feature learning ability, they rely heavily on sufficient training data and elaborate parameter tuning to obtain satisfactory clustering performance.

Sampling Methods. Ablation experiments are further conducted to evaluate different sampling strategies in the proposed module. **Figure 4** illustrates the quantitative comparison results across five datasets, with the number of labeled instances and F1-scores serving as evaluation axes. The baseline sampling strategies include heuristic sampling and random sampling, which are compared with the proposed TSD sampling method.

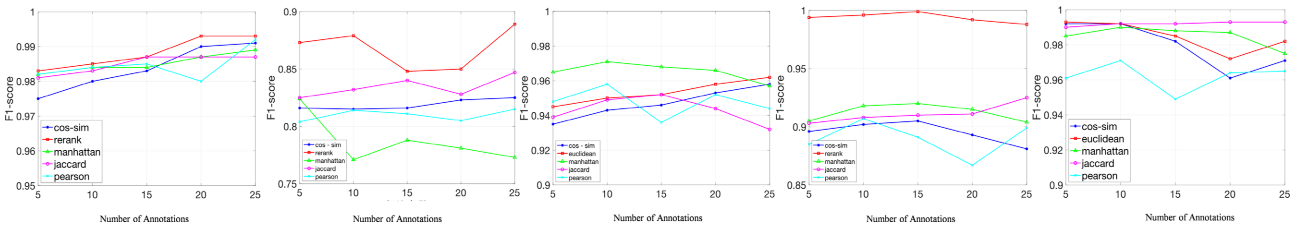

Figure 3. Comparison results of clustering algorithms.

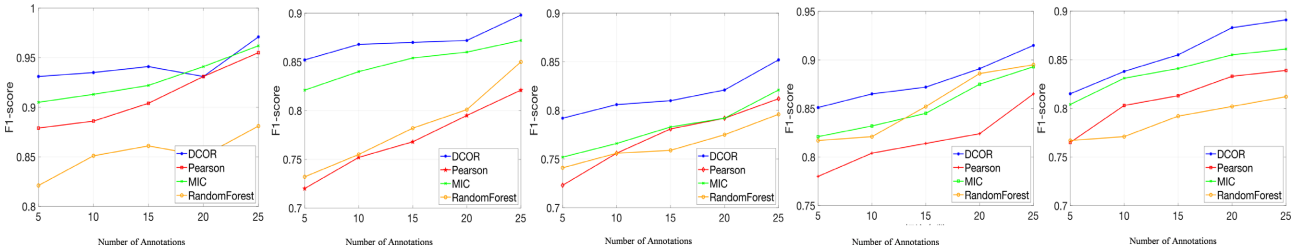

Figure 4. Comparison of experimental results for different sampling method Experiments conducted with varying numbers of labeled instances.

Experimental results with varying labeled sample sizes demonstrate that the TSD sampling method achieves significant performance superiority. It outperforms heuristic sampling and random sampling by average margins of 3.63% and 2.02%, respectively. On the Beers dataset, TSD exhibits outstanding detection performance: it achieves an F1-score of 0.994 with only 5 labeled samples, compared with 0.893 for heuristic sampling and 0.932 for random sampling. Even when the number of labeled samples increases to 25, TSD maintains a high F1-score of 0.988, while heuristic and random sampling only reach 0.911 and 0.925, respectively. These results demonstrate that TSD consistently outperforms the two baseline methods at all labeling levels on the Beers dataset.

Overall, TSD achieves excellent performance across most datasets, benefiting from its adaptive sampling mechanism. It dynamically optimizes sampling strat-

egies according to dataset characteristics and effectively screens high-information samples, thereby providing high-quality data for subsequent clustering procedures. Heuristic sampling shows unique advantages for datasets with specific structural features yet suffers from limited generalization across diverse datasets due to its strong task specificity. By contrast, random sampling delivers relatively moderate overall performance but maintains stable applicability for datasets with homogeneous distributions, serving as a reliable baseline for evaluating complex sampling algorithms.

Performance Comparison with State-of-the-Art Error Detection Models. **Figure 5** compares the performance of the proposed JCED model with several state-of-the-art error detection baselines, including Raha, SPAED, and ED2, across all five datasets.

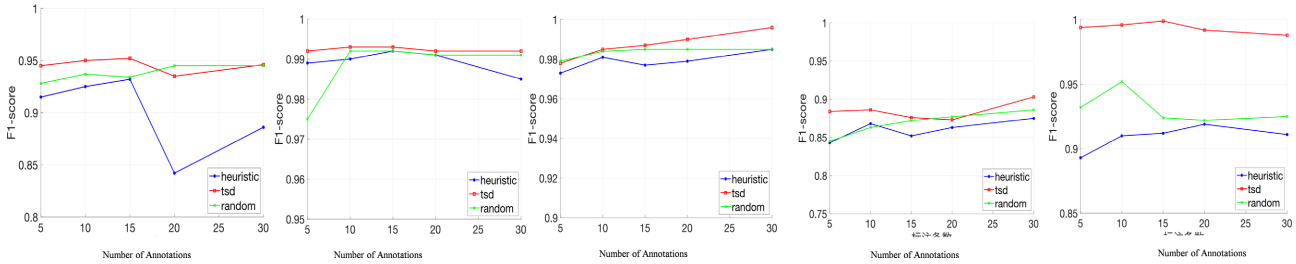

Figure 5. Comparison results with the baseline error detection model.

Quantitative results under different annotation budgets reveal that JCED surpasses Raha by an average margin of 140% across all datasets. On the Hospital dataset, JCED achieves an F1-score of 0.978 with only 5 annotated samples, whereas Raha obtains merely 0.249. When the annotation number increases to 25, JCED's F1-score further improves to 0.996, while Raha only slightly rises to 0.986. This verifies that JCED maintains a persistent and prominent performance advantage over Raha at all labeling levels, enabling more accurate error detection. A similar performance trend is observed on the Flight, Beers, Rayyan, and Movies datasets. In particular, JCED achieves a substantial average improvement of 1336% over ED2. On the Hospital dataset, ED2 yields a near-zero F1-score with 5 annotations, while JCED reaches 0.978; at 25 annotations, ED2's performance declines to 0.750, whereas JCED climbs to 0.996. On the Rayyan dataset, ED2 obtains an F1-score of 0.050 with 5 annotations, in sharp contrast to JCED's 0.945. When annotations increase to 25, the two methods rise to 0.805 and 0.961, respectively. These results fully demonstrate the inferior performance of the ED2 model on the tested datasets.

In comparison with SPAED, JCED achieves an average performance gain of 27.7%. On the Beers dataset with 5 labeled samples, JCED reaches an F1-score of 0.994, outperforming SPAED's 0.842. At 25 labeled samples, SPAED achieves a slightly higher score of 0.998, while JCED still maintains a competitive level of 0.988. Although SPAED can occasionally match JCED's performance under specific settings, JCED exhibits stable superiority across most labeling scales. On the

Movies dataset, JCED achieves F1-scores of 0.993 and 0.993 with 5 and 25 labels, respectively, while SPAED only obtains 0.865 and 0.911 under the same settings. These comparative results validate JCED's superior adaptability and stability in handling diverse data scenarios.

Time Efficiency Analysis. Figure 6 summarizes the runtime of all compared error detection models on five datasets under annotation numbers of 5, 10, 15, 20, and 30. Experimental results indicate that JCED improves time efficiency by 13.76%, 61.5%, and 58.64% on average compared with Raha, ED2, and SPAED, respectively, demonstrating its prominent computational efficiency.

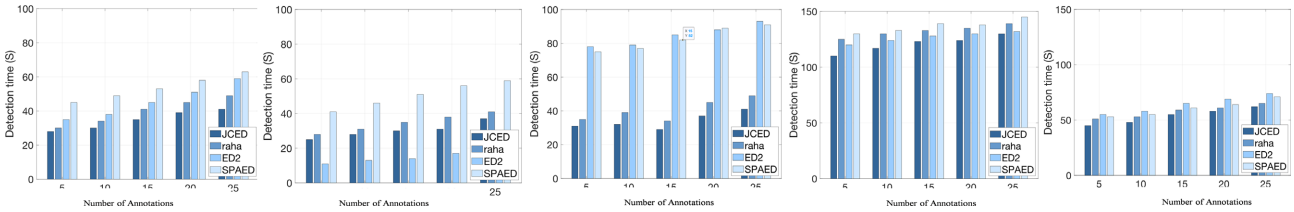

Figure 6. Comparison of time efficiency between the baseline error detection model and the proposed model

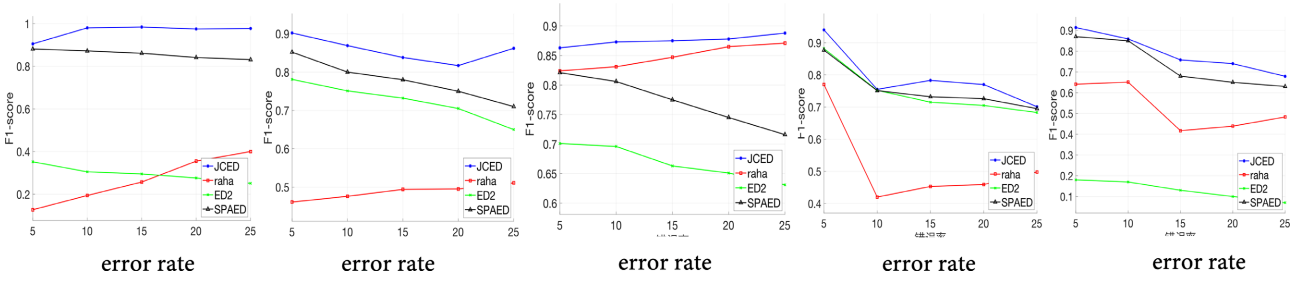

Figure 7. Comparison of error detection model performance under different error rates

Robustness Analysis Under Varying Error Rates. Figure 7 evaluates the model performance of JCED and baseline methods under varying data error rates (5%, 10%, 15%, 20%, and 30%). The results show that JCED achieves average performance improvements of 102%, 169%, and 9.7% over Raha, ED2, and SPAED, respectively.

Across all five datasets, JCED maintains stable and superior performance as the error rate increases from 5% to 30%. On the Hospital dataset, JCED's F1-score decreases slightly from 0.95 to 0.88, with a decline rate of only 7.4%, while Raha presents a larger performance drop of 12.3%. On the Flight dataset, JCED's F1-score declines by 8.7% (from 0.92 to 0.84), whereas ED2 experiences a significant drop of 17.3% (from 0.75 to 0.62). On the Beers dataset, JCED's F1-score decreases by 7.4% (from 0.94 to 0.87). In contrast, other baseline models show more severe performance degradation: SPAED decreases by 12.9% (0.85 to 0.74), Rayyan by 7.5% (0.93 to 0.86), Raha by 14.6% (0.82 to 0.70), and ED2 by 16.7% (0.72 to 0.60). On the Movies dataset, the baseline method exhibits a 6.3% performance decline.

Overall, JCED's average F1-score decline across all datasets is merely 7.5%, with the drop on each individual dataset lower than 10%. By contrast, Raha, ED2, and SPAED suffer average performance reductions of 13.8%, 16.5%, and 11.2%, re-

spectively. These results strongly validate the outstanding robustness and stability of the proposed JCED model against increasing data error rates.

5.4. Ablation Experiment

Table 1. Ablation experiments on different datasets.

Algorithm	hospital	flight	beers	rayyan	movie
withno-Grouping Module	0.948	0.836	0.934	0.948	0.923
withno-Sentence Transformer	0.938	0.831	0.935	0.944	0.927
withno-WordVec	0.930	0.825	0.924	0.934	0.901
withno-Sampling Module	0.955	0.847	0.914	0.942	0.911
withno-Classifier Module	0.975	0.871	0.974	0.963	0.942
with-All	**0.986**	**0.884**	**0.999**	**0.974**	**0.959**

Table 1 presents the quantitative results of module-level ablation experiments across five datasets, where the optimal results are highlighted in bold. The comprehensive ablation analysis verifies that the full JCED model consistently outperforms all its variant models with individual components removed, demonstrating the robustness and overall efficacy of the complete framework.

Quantitative statistics demonstrate the individual contributions of each module. The grouping module yields an average performance improvement of 4.6% across all datasets. As a core component of the joint representation module, Sentence Transformer enhances contextual semantic modeling capability, bringing a 5.0% average performance gain. Although Word2Vec generates static word embeddings with limited contextual perception and fixed window constraints, it still delivers a 6.4% average improvement by providing stable lexical feature representations. Additionally, the proposed sampling module contributes a 5.1% performance boost, while the classifier module achieves a moderate average gain of 1.6%.

Further dataset-specific analysis reveals that removing any module induces apparent performance degradation, and the contribution degree of each component varies significantly across diverse datasets. The grouping module achieves the largest gain of 6.9% on the Beers dataset but only a marginal improvement of 2.7% on the Rayyan dataset. Similarly, Sentence Transformer and Word2Vec attain prominent performance increments of 6.8% and 8.1% on the Beers dataset, while their improvements decrease to 3.2% and 4.3% on the Rayyan dataset, respectively. For the sampling module, the performance gain reaches 9.3% on the Beers dataset but drops to 3.2% on the Hospital dataset. The classifier module yields a maximum improvement of 2.6% on the Beers dataset and a minimum gain of 1.1% on the Hospital dataset.

These results quantitatively validate the differentiated effectiveness of each JCED module for distinct data characteristics, confirming the model's strong

adaptability to complex and diverse datasets. Overall, every embedded component is indispensable to the final performance of JCED. The ablation study not only verifies the superiority of the complete model framework but also clarifies the complementary interactions among different modules. This provides valuable insights for subsequent model optimization, cross-domain adaptation, and advanced error detection research.

5.5. Conclusions

This study proposes JCED, an end-to-end tabular error detection framework tailored for low-label scenarios. By integrating four complementary modules—distance-correlation-based column grouping, hybrid context-adaptive embedding representation, two-stage informative instance sampling, and IPS-bias-corrected classification—JCED establishes a robust and generalizable error detection paradigm. Unlike individual module improvements, the core innovation stems from their inherent synergy, which substantially enhances detection reliability under limited annotation budgets. Extensive experiments on multiple public datasets validate the key conclusions as follows:

1) JCED achieves outstanding detection performance with sparse labeled samples and exhibits strong robustness against diverse data error rates.

2) The full combination and coordination of all embedded modules contribute to the model's optimal overall performance, demonstrating clear synergistic effects.

3) The collaborative grouping and joint embedding learning effectively capture structural and contextual table features. The optimized sampling strategy ensures reliable detection under label scarcity, while IPS weighting effectively alleviates sampling bias, further stabilizing and improving the final detection performance.

Funding

This study is supported by the Special Project of Scientific and Technological Support for Social Governance and Smart Society of the National Key R&D Program of China (Grant No. 2024YFC3306900).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Min, S., Lewis, M., Zettlemoyer, L. and Hajishirzi, H. (2022) MetaICL: Learning to Learn in Context. *Proceedings of the* 2022 *Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, Seattle, July 2022, 2791-2809. https://doi.org/10.18653/v1/2022.naacl-main.201

[2] Wu, Y., Miao, X., Nan, Z., Zhang, J., He, J. and Yin, J. (2024) Effective and Efficient Multi-View Imputation with Optimal Transport. *IEEE Transactions on Knowledge and Data Engineering*, **36**, 6029-6041. https://doi.org/10.1109/tkde.2024.3387439

[3] Guo, Z.M. and Zhou, A.Y. (2002) Research on Data Quality and Data Cleaning: A

Survey. *Journal of Software*, **13**, 2076-2082.

[4] Wang, P. and He, Y. (2019) Uni-Detect: A Unified Approach to Automated Error Detection in Tables. *Proceedings of the* 2019 *International Conference on Management of Data*, Amsterdam, 30 June-5 July 2019, 811-828. https://doi.org/10.1145/3299869.3319855

[5] Panko, R.R. (1998) What We Know about Spreadsheet Errors. *Journal of Organizational and End User Computing*, **10**, 15-21. https://doi.org/10.4018/joeuc.1998040102

[6] Powell, S.G., Baker, K.R. and Lawson, B. (2009) Errors in Operational Spreadsheets: A Review of the State of the Art. 2009 42*nd Hawaii International Conference on System Sciences*, Waikoloa, 5-8 January 2009, 1-8.

[7] Mahdavi, M., Abedjan, Z., Castro Fernandez, R., Madden, S., Ouzzani, M., Stonebraker, M., *et al.* (2019) Raha: A Configuration-Free Error Detection System. *Proceedings of the* 2019 *International Conference on Management of Data*, Amsterdam, 30 June-5 July 2019, 865-882. https://doi.org/10.1145/3299869.3324956

[8] Heidari, A., McGrath, J., Ilyas, I.F. and Rekatsinas, T. (2019) HoloDetect: Few-Shot Learning for Error Detection. *Proceedings of the* 2019 *International Conference on Management of Data*, Amsterdam, 30 June-5 July 2019, 829-846. https://doi.org/10.1145/3299869.3319888

[9] Pham, M., Knoblock, C.A., Chen, M., Vu, B. and Pujara, J. (2021) SPADE: A Semi-Supervised Probabilistic Approach for Detecting Errors in Tables. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Montreal, 19-27 August 2021, 3543-3551. https://doi.org/10.24963/ijcai.2021/488

[10] Abedjan, Z., Chu, X., Deng, D., Fernandez, R.C., Ilyas, I.F., Ouzzani, M., *et al.* (2016) Detecting Data Errors: Where Are We and What Needs to Be Done? *Proceedings of the VLDB Endowment*, **9**, 993-1004. https://doi.org/10.14778/2994509.2994518

[11] Neutatz, F., Mahdavi, M. and Abedjan, Z. (2019) ED2: A Case for Active Learning in Error Detection. *Proceedings of the* 28*th ACM International Conference on Information and Knowledge Management*, Beijing, 3-7 November 2019, 2249-2252. https://doi.org/10.1145/3357384.3358129

[12] Chen, B.L., Wang, T.J., Ren, L.N. and Huang, R.Z. (2023) Method for Chinese Grammar Error Detection Integrating ELECTRA and Text Local Information. *Computer Engineering*, **49**, 304-311.

[13] Guo, K.X., Wang, H.J. and Bai, Z.X. (2022) A Word-Level Text Error Detection Model Integrating Multi-Channel CNN and BiGRU. *Computer Engineering*, **48**, 63-70.

[14] Ilyas, I.F. and Chu, X. (2015) Trends in Cleaning Relational Data: Consistency and Deduplication. *Foundations and Trends® in Databases*, **5**, 281-393. https://doi.org/10.1561/1900000045

[15] Abdelaal, M., Ktitarev, T., Städtler, D. and Schöning, H. (2024) Saged: Few-Shot Meta Learning for Tabular Data Error Detection. *International Conference on Extending Database Technology* (*EDBT*'24), Paestum, 25-28 March 2024, 386-398.

[16] Mahdavi, M. and Abedjan, Z. (2020) Baran: Effective Error Correction via a Unified Context Representation and Transfer Learning. *Proceedings of the VLDB Endowment*, **13**, 1948-1961. https://doi.org/10.14778/3407790.3407801

[17] Rekatsinas, T., Chu, X., Ilyas, I.F. and Ré, C. (2017) Holoclean: Holistic Data Repairs with Probabilistic Inference. *Proceedings of the VLDB Endowment*, **10**, 1190-1201. https://doi.org/10.14778/3137628.3137631

[18] Bleifuß, T., Kruse, S. and Naumann, F. (2017) Efficient Denial Constraint Discovery with Hydra. *Proceedings of the VLDB Endowment*, **11**, 311-323. https://doi.org/10.14778/3157794.3157800

[19] Beskales, G., Ilyas, I.F., Golab, L. and Galiullin, A. (2013) On the Relative Trust between Inconsistent Data and Inaccurate Constraints. 2013 *IEEE* 29*th International Conference on Data Engineering* (*ICDE*), Brisbane, 8-12 April 2013, 541-552. https://doi.org/10.1109/icde.2013.6544854

[20] Dasu, T. and Loh, J.M. (2012) Statistical Distortion: Consequences of Data Cleaning. *Proceedings of the VLDB Endowment*, **5**, 1674-1683. https://doi.org/10.14778/2350229.2350279

[21] Wu, E. and Madden, S. (2013) Scorpion: Explaining Away Outliers in Aggregate Queries. *Proceedings of the VLDB Endowment*, **6**, 553-564. https://doi.org/10.14778/2536354.2536356

[22] Prokoshyna, N., Szlichta, J., Chiang, F., Miller, R.J. and Srivastava, D. (2015) Combining Quantitative and Logical Data Cleaning. *Proceedings of the VLDB Endowment*, **9**, 300-311. https://doi.org/10.14778/2856318.2856325

[23] Dallachiesa, M., Ebaid, A., Eldawy, A., Elmagarmid, A., Ilyas, I.F., Ouzzani, M., *et al.* (2013) NADEEF: A Commodity Data Cleaning System. *Proceedings of the* 2013 *ACM SIGMOD International Conference on Management of Data*, New York, 22-27 June 2013, 541-552. https://doi.org/10.1145/2463676.2465327

[24] Khayyat, Z., Ilyas, I.F., Jindal, A., Madden, S., Ouzzani, M., Papotti, P., *et al.* (2015) BigDansing: A System for Big Data Cleansing. *Proceedings of the* 2015 *ACM SIGMOD International Conference on Management of Data*, Melbourne, 31 May-4 June 2015, 1215-1230. https://doi.org/10.1145/2723372.2747646

[25] Kandel, S., Paepcke, A., Hellerstein, J. and Heer, J. (2011) Wrangler: Interactive Visual Specification of Data Transformation Scripts. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, 7-12 May 2011, 3363-3372. https://doi.org/10.1145/1978942.1979444

[26] Abedjan, Z., Morcos, J., Ilyas, I.F., Ouzzani, M., Papotti, P. and Stonebraker, M. (2016) DataXFormer: A Robust Transformation Discovery System. 2016 *IEEE* 32*nd International Conference on Data Engineering* (*ICDE*), Helsinki, 16-20 May 2016, 1134-1145. https://doi.org/10.1109/icde.2016.7498319

[27] Fan, W., Li, J., Ma, S., Tang, N. and Yu, W. (2011) Towards Certain Fixes with Editing Rules and Master Data. *The VLDB Journal*, **21**, 213-238. https://doi.org/10.1007/s00778-011-0253-7

[28] Chu, X., Morcos, J., Ilyas, I.F., Ouzzani, M., Papotti, P., Tang, N., *et al.* (2015) KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing. *Proceedings of the* 2015 *ACM SIGMOD International Conference on Management of Data*, Melbourne, 31 May-4 June 2015, 1247-1261. https://doi.org/10.1145/2723372.2749431

[29] Krishnan, S., Wang, J., Wu, E., Franklin, M.J. and Goldberg, K. (2016) ActiveClean: Interactive Data Cleaning for Statistical Modeling. *Proceedings of the VLDB Endowment*, **9**, 948-959. https://doi.org/10.14778/2994509.2994514

[30] Visengeriyeva, L. and Abedjan, Z. (2018) Metadata-Driven Error Detection. *Proceedings of the* 30*th International Conference on Scientific and Statistical Database Management*, Bozen-Bolzano, 9-11 July 2018, 1-12. https://doi.org/10.1145/3221269.3223028

[31] Huang, Z. and He, Y. (2018) Auto-Detect: Data-Driven Error Detection in Tables.

Proceedings of the 2018 *International Conference on Management of Data*, Houston, 10-15 June 2018, 1377-1392. https://doi.org/10.1145/3183713.3196889

[32] Shannon, C.E. (1948) A Mathematical Theory of Communication. *Bell System Technical Journal*, **27**, 379-423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[33] Wang, H., Khoshgoftaar, T.M. and Van Hulse, J. (2010) A Comparative Study of Threshold-Based Feature Selection Techniques. 2010 *IEEE International Conference on Granular Computing*, San Jose, 14-16 August 2010, 499-504. https://doi.org/10.1109/grc.2010.104

[34] Kay, P. and Regier, T. (2003) Resolving the Question of Color Naming Universals. *Proceedings of the National Academy of Sciences*, **100**, 9085-9089. https://doi.org/10.1073/pnas.1532837100

[35] Schnabel, T., Swaminathan, A., Singh, A., Chandak, N. and Joachims, T. (2016) Recommendations as Treatments: Debiasing Learning and Evaluation. *International Conference on Machine Learning*, New York, 19-24 June 2016, 1670-1679.

[36] Li, X., Dong, X.L., Lyons, K., Meng, W. and Srivastava, D. (2015) Truth Finding on the Deep Web: Is the Problem Solved? arXiv: 1503.00303.

[37] Ward, F. (2020) Beer Dataset Analysis.

[38] Ouzzani, M., Hammady, H., Fedorowicz, Z. and Elmagarmid, A. (2016) Rayyan—A Web and Mobile App for Systematic Reviews. *Systematic Reviews*, **5**, Article No. 210. https://doi.org/10.1186/s13643-016-0384-4

[39] Konda, P., Das, S., C., P.S.G., Doan, A., Ardalan, A., Ballard, J.R., *et al.* (2016) Magellan: Toward Building Entity Matching Management Systems over Data Science Stacks. *Proceedings of the VLDB Endowment*, **9**, 1581-1584. https://doi.org/10.14778/3007263.3007314

[40] Pit-Claudel, C., Mariet, Z., Harding, R. and Madden, S. (2016) Outlier Detection in Heterogeneous Datasets Using Automatic Tuple Expansion.