Information Research - Vol. 31 No. iConf (2026)

Learning from unknown-unknowns: inconsistency-driven sampling for improving LLM entity matching

Kota Okayama, Ito Hiroyoshi, and Atsuyuki Morishima

DOI: https://doi.org/10.47989/ir31iConf64127

Abstract

Introduction. While large language models (LLMs) demonstrate high performance in entity matching, the '*unknown-unknown*' problem, where models confidently make incorrect predictions, remains a significant challenge. This research focuses on the manifestation of this problem as logical inconsistencies, such as violations of transitivity (e.g., A=B and B=C, but A≠C) across multiple matching decisions.

Method. '*Inconsistent triangles,*' in which the transitive law is violated among three entities, were detected, and scored based on their degree of contradiction. Pairs with higher inconsistency scores were prioritised for annotation, and the resulting labeled data was fed back to the model through fine-tuning or few-shot learning.

Analysis. The proposed method was evaluated on multiple datasets, including Japanese and English data. Its performance was compared against existing baseline methods, such as uncertainty sampling and random sampling, using the pairwise F1 score as the primary evaluation metric.

Results. The experiments revealed that the proposed inconsistency-driven sampling strategy outperformed or achieved comparable performance to existing methods across all datasets.

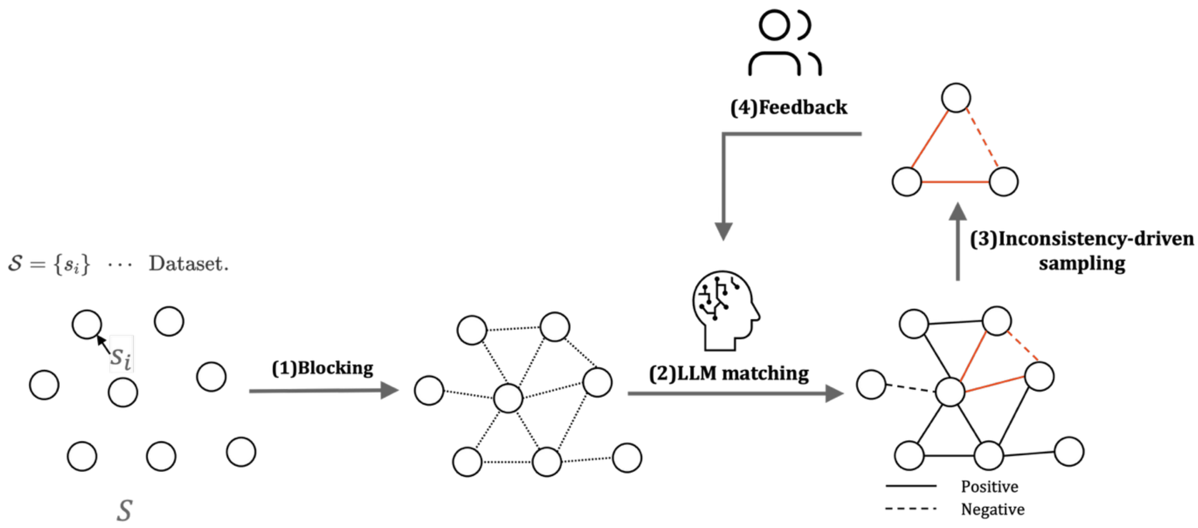
Conclusion. By leveraging inconsistency to actively select training data, our approach achieves learning efficiency, demonstrating improved entity matching performance under the same annotation budget.

Introduction

Entity Matching (EM) is a fundamental operation for identifying the records that refer to the same real-world entity (Barlaug, et al., 2021). Recently, while large language models (LLMs) have shown promising performance in EM, their black-box nature and the persistent ’*unknown-unknown problem*’, where models make highly confident yet incorrect predictions, remain significant challenges (Chung, et al., 2019). Traditional active learning strategies like uncertainty sampling can detect instances where the model is uncertain, but struggle to identify these ’*unknown-unknown*’ errors (Settles, 2010). To clarify this challenge, we define the two components of the ’*unknown-unknown*’ state: Unknown (1) is the underlying error (the incorrect matching decision), and Unknown (2) is the lack of uncertainty (the high confidence score) that shields the error from detection. Our work specifically targets this hidden population of errors.

An approach exists for addressing ‘*unknown-unknown*’ in EM by utilising inconsistencies in transitivity (Ito et al., 2025). This method is based on the insight that high-confidence errors often manifest as logical inconsistencies, specifically violations of the transitive property (i.e., if A=B and B=C, then A=C). By focusing on these inconsistent pairs, they achieved efficient improvements in matching accuracy. However, their validation was limited to classical machine learning models. Consequently, the application of this strategy to modern LLMs has not been systematically investigated, and it remains unclear how to most effectively feedback the information gleaned from these inconsistencies, for instance, through fine-tuning versus few-shot prompting.

Therefore, this research systematically validates whether an ‘*inconsistency-driven*’ sampling strategy can effectively improve LLM accuracy for entity matching under a limited annotation budget, with a particular focus on resolving the ‘*unknown-unknown*’ problem. Figure 1 provides an overview of this strategy. The process takes a set of entities as input and proceeds as follows: (1) Blocking narrows down the candidate pairs of nodes that are likely to match. (2) The LLM calculates a match probability for each candidate pair, constructing an undirected graph where edges are weighted by these probabilities. (3) An inconsistency score, representing the degree of transitive violation, is calculated for each triangle in the graph. The unique set of pairs from the triangles with the highest scores is selected for human annotation. (4) Finally, the human-verified labels are fed back to the LLM.



$\mathcal{S} = \{s_i\}$ ··· Dataset.
s_i
S
(1)Blocking
(2)LLM matching
(3)Inconsistency-driven sampling
(4)Feedback
Positive
Negative

Figure 1. Inconsistency-driven human-in-the-loop LLM entity matching.

This paper describes our method and presents the results of experiments comparing multiple sampling strategies (Inconsistency-driven, Uncertainty, Diversity, ActiveLLM, and Random) and feedback mechanisms (few-shot and fine-tuning (FT)) under the same annotation budget.

Our research questions (RQs) are as follows:

- RQ1: To what extent is the inconsistency-driven approach effective for LLM-based entity matching?
- RQ2: Between few-shot and FT, which feedback strategy is more effective, and under which conditions or datasets?

Contributions and Findings: To address the RQs, we conduct the following:

- We introduce transitive inconsistency sampling to LLM-based EM and clarify its effectiveness by comparing it against other methods across multiple datasets.
- We compare two feedback mechanisms, few-shot and FT, to delineate the conditions under which each is most effective.

Specifically, to answer the two RQs, we conducted experiments using bibliographic (bib), music, person, and product datasets, evaluating differences in F1 scores. Regarding RQ1, our results show that while the maximum performance achievable by an LLM alone varies by dataset and task setting, the inconsistency-driven selection strategy consistently yields improvements in all cases. For RQ2, we found that FT (specifically, fine-tuning on inconsistency-derived samples) was particularly effective across most datasets, while few-shot prompting was also beneficial under certain conditions. These findings indicate that for tackling the challenge of inconsistency in LLMs for EM, a strategy that prioritises the annotation of high-value examples based on transitive logic is highly effective.

Related work

This section reviews related research to clarify the positioning of our study.

EM methods using classical machine learning and LLMs

Existing EM methods include classical approaches based on string distance, token matching, or rule synthesis (Benjelloun et al., 2009; Cohen et al., 2003; Jaro, 1989) machine learning techniques that combine feature engineering with classifiers like Random Forest (Das et al., 2017; Gokhale et al., 2014); Human-in-the-Loop systems that incorporate human judgment through crowdsourcing and similarity learning based on metric learning (Peeters et al., 2022; Osawa et al., 2021; Takashi et al., 2019). More recently, methods using LLMs as the matcher have emerged, establishing them as a powerful approach to EM tasks (Peeters et al., 2023a).

Methods for improving matching model performance

Efforts have also been made to automatically enhance LLM performance for specific tasks. For instance, Ji et al. (2025) proposed autonomous learning, which simulates the process of human learning from reference materials. Unlike our study, which assumes human involvement in the improvement process, this method may be unable to learn from information the model does not recognise it lacks. Another active learning method, ActiveLLM (Bayer et al., 2025), uses an LLM as a selector for training data but also suffers from the inability to capture the LLM's '*unknown-unknowns*'. From the perspective of prompt engineering, Wang et al. (2025) investigated performance improvements by comparing three strategies for EM: Matching (pairwise classification), Comparing (comparison between two candidates), and Selecting (selection from a list). The directions for improvement are diverse, also including Retrieval-Augmented Generation (RAG) to integrate external knowledge (Peng et al., 2023) and fine-tuning specialised for EM (Steiner et al., 2025).

Methods for improving EM models using transitivity

Other attempts have been made to boost EM performance by manually assisting or refining an LLM's reasoning. The research on inference assistance and consistency by Zhu et al. (2020) involves manual corrections based on the transitive property, presupposing a crowdsourcing environment. Our work differs in that it integrates the transitive property into active learning as an indicator for *'inconsistency detection and selection,'* thereby increasing the informational value of annotated examples.

As another example of applying transitivity to EM models for active learning, Ito et al. (2025) proposed a method to address the *'unknown-unknown'* problem in machine learning matchers that is, the issue of making high confidence but incorrect predictions. Their method detects *'inconsistencies'* where matching results among three entities violate the transitive property and prompts human intervention by presenting the pairs causing the contradiction. This allows for the efficient discovery of instances where the model is confidently wrong, which are often overlooked by conventional active learning methods like uncertainty sampling.

Positioning of this study

The novelty of this research lies in positioning an LLM as the primary matcher and investigating whether its performance can be improved by selecting pairs for annotation based on a transitivity-derived inconsistency score. Furthermore, we explore how to most effectively provide this feedback to the LLM.

Problem definition

To explain the problem addressed in this paper, we first present concrete examples of the input and output. The following are two entities from a music dataset (Köpcke et al., 2010). Each record has attributes such as title, length, artist, album, year:

```
data:
    title: 003-She's My Best Friend
    length: 6m 0sec
    artist: Lou Reed
    album: Coney Island Baby (1976)
    year: 'null'
```

```
data:
    title: She's My Best Friend
    length: '360360'
    artist: Lou Reed
    album: Coney Island Baby
    year: '1976'
```

Figure 2. Input examples.

Given this input, a matcher f, implemented with an LLM, returns a matching decision (either 'Yes' or 'No') and its corresponding confidence score. A 'Yes' decision corresponds to a match probability of 0.5 or greater, while a 'No' decision corresponds to a probability less than 0.5. Specifically, the output is as follows:

- When determined to be the same entity (a match): Yes. Match Probability: 0.95
- When determined to be different entities (not a match): No. Match Probability: 0.20

We define the problem mathematically. Let S be the set of entities and $\mathcal{X} = \{(s_i, s_j) | i < j\}$ be the set of all possible pairs. The label set is $\mathcal{Y} = \{\text{Match}, \text{Unmatch}\}$. The universe of all correctly labeled pairs is $\mathcal{U}_n \subset \mathcal{X} \times \mathcal{Y}$, from which we have a small, labeled subset $\mathcal{L}_m \subset \mathcal{U}_n$ ($m \ll \| \mathcal{X} \|$). Our matcher $f: \mathcal{X} \to \mathcal{Y}$ is a function from a hypothesis class $\mathcal{F}$ that also provides a match probability $P(s_i, s_j)$. Our objective is to design a selection strategy $Q: \mathcal{U}_n \to \mathcal{L}_m$ to train the most accurate matcher f

possible under a limited annotation budget (Ito et al., 2025). In essence, we aim to select the optimal labeled set $\mathcal{L}_m$.

$$\underset{\mathcal{L}_m \subseteq \mathcal{U}_n}{\operatorname{argmax}} \frac{1}{n} \sum_{(x,y) \in \mathcal{U}_n} \delta \left(f(x) = y \vee (x,y) \in \mathcal{L}_m \right) \quad (1)$$

The goal is to increase the overall accuracy by having the model correctly predict unlabeled pairs while obtaining the ground truth for difficult examples via $\mathcal{L}_m$. In other words, we aim to sample pairs that the model is likely to misclassify.

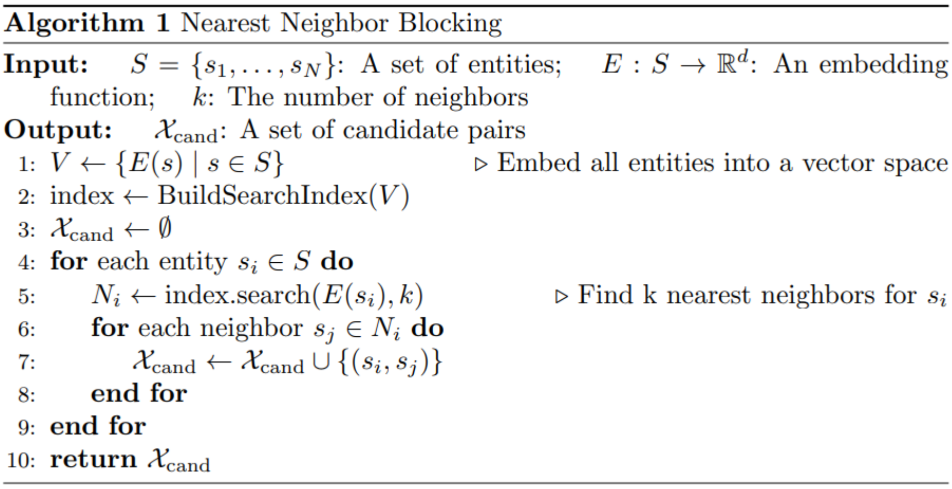
Method

Framework overview

This section details the overall framework for inconsistency-driven active learning, as illustrated in Figure 1. The workflow proceeds as follows: (1) Blocking is applied to narrow down the set of candidate pairs to those most likely to match. (2) Using the current model, match probabilities are inferred for all candidate pairs, from which *'inconsistent triangles'* that violate the transitive property are calculated and extracted. (3) Leveraging these inconsistent triangles, pairs exhibiting a high degree of inconsistency are prioritised for human annotation. (4) The newly annotated pairs are then fed back to the model.

Blocking

To reduce the computational cost of entity matching, a blocking process is first applied to remove non-matching pairs, thereby efficiently narrowing down the set of candidate pairs. As blocking is not the primary focus of this research, we adopt a standard blocking method based on nearest neighbor search using general-purpose embedding representations. The procedure is shown in Algorithm 1.



Algorithm 1 Nearest Neighbor Blocking

Input: $S = \{s_1, \ldots, s_N\}$: A set of entities; $E : S \to \mathbb{R}^d$: An embedding function; k: The number of neighbors

Output: $\mathcal{X}_{\text{cand}}$: A set of candidate pairs

```
1:  $V \leftarrow \{E(s) \mid s \in S\}$                ▷ Embed all entities into a vector space
2:  index ← BuildSearchIndex($V$)
3:  $\mathcal{X}_{\text{cand}} \leftarrow \emptyset$
4:  for each entity $s_i \in S$ do
5:      $N_i \leftarrow$ index.search($E(s_i), k$)        ▷ Find k nearest neighbors for $s_i$
6:      for each neighbor $s_j \in N_i$ do
7:          $\mathcal{X}_{\text{cand}} \leftarrow \mathcal{X}_{\text{cand}} \cup \{(s_i, s_j)\}$
8:      end for
9:  end for
10: return $\mathcal{X}_{\text{cand}}$
```

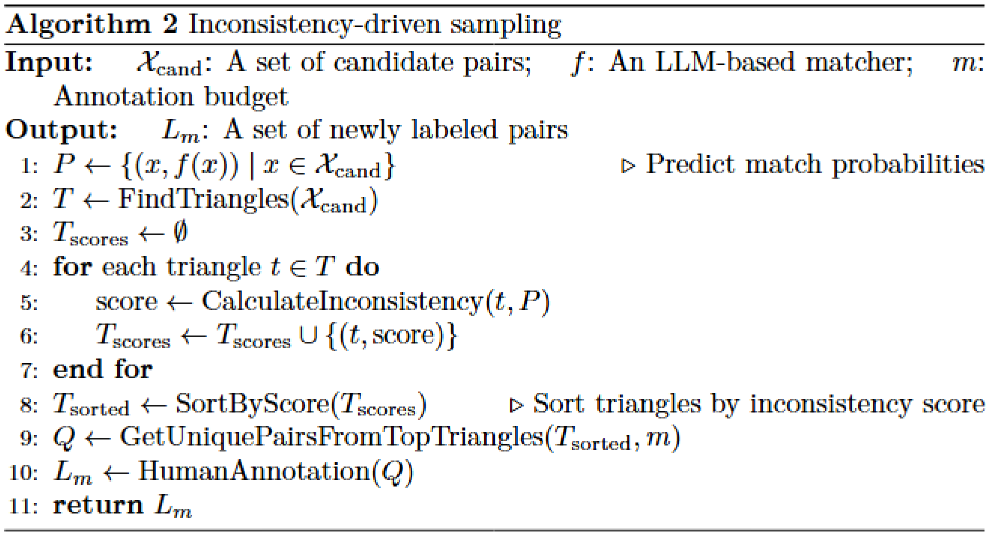
Algorithm 1. Blocking algorithm.

First, the algorithm takes as input the set of all entities $S = \{s_1, \ldots, s_N\}$, an embedding function E, and the number of neighbors k to search for each entity. The algorithm initially transforms all entities $s \in S$ into a d-dimensional vector space using the embedding function E, generating a set of vectors V (Line 1). Next, it constructs an index from these vectors to enable efficient nearest neighbor search (Line 2).

After the index is built, the algorithm searches for the k nearest neighboring entities s_j for each entity's embedding vector $E(s_i)$ (Line 4). In this study, we used the L2 distance between the embedding vectors. The pair (s_i, s_j), consisting of the original entity s_i and a found neighbor s_j, is then added to the candidate pair set $\mathcal{X}_{\text{cand}}$ (Line 6). By repeating this process for all entities, a refined set of candidate pairs $\mathcal{X}_{\text{cand}}$ is obtained, containing only pairs with a high likelihood of matching.

Inconsistency-driven sampling

The inconsistency-driven sampling algorithm, corresponding to part (3) of Figure 1, is presented in Algorithm 2.



```
Algorithm 2 Inconsistency-driven sampling
Input:   $\mathcal{X}_{\text{cand}}$: A set of candidate pairs;   $f$: An LLM-based matcher;   $m$:
    Annotation budget
Output:   $L_m$: A set of newly labeled pairs
 1: $P \leftarrow \{(x, f(x)) \mid x \in \mathcal{X}_{\text{cand}}\}$          ▷ Predict match probabilities
 2: $T \leftarrow \text{FindTriangles}(\mathcal{X}_{\text{cand}})$
 3: $T_{\text{scores}} \leftarrow \emptyset$
 4: for each triangle $t \in T$ do
 5:     score $\leftarrow \text{CalculateInconsistency}(t, P)$
 6:     $T_{\text{scores}} \leftarrow T_{\text{scores}} \cup \{(t, \text{score})\}$
 7: end for
 8: $T_{\text{sorted}} \leftarrow \text{SortByScore}(T_{\text{scores}})$      ▷ Sort triangles by inconsistency score
 9: $Q \leftarrow \text{GetUniquePairsFromTopTriangles}(T_{\text{sorted}}, m)$
10: $L_m \leftarrow \text{HumanAnnotation}(Q)$
11: return $L_m$
```

Algorithm 2. Inconsistency-driven sampling.

This algorithm takes the candidate pair set $\mathcal{X}_{\text{cand}}$, an LLM matcher f, and an annotation budget m as input. First, it uses the initial model f to predict the match probability for every candidate pair $x \in \mathcal{X}_{\text{cand}}$, storing the results as P.

Next, the algorithm moves to its core component: the inconsistency calculation. It identifies all triangular structures $t = \{(s_a, s_b), (s_b, s_c), (s_c, s_a)\}$ formed by three entities $\{s_a, s_b, s_c\}$ within the candidate pair set (Line 2).

Figure 3 illustrates the core concept of a logically inconsistent triplet, which our sampling strategy targets. In this figure, solid edges represent matched pairs (Match), and dashed edges represent non-matched pairs (Unmatch). A contradictory triangle is explicitly defined as a triplet where the model predicts two matches and one non-match (e.g., Match-Match-Unmatch), violating the transitive property. For each identified triangle t, it calculates an *'inconsistency score'* using the predicted probabilities $p(a, b), p(b, c), p(c, a)$ corresponding to its three edges (Lines 4-7). This score is designed to quantify the degree of logical inconsistency within the triplet, thereby detecting the contradictory triangles as visually represented in Figure 3. This score is defined by

$$\text{score} = \sum_{(\text{i,j,k}) \in \text{cyc(a,b,c)}} p(i, j) p(j, k) \big(1 - p(k, i)\big) \tag{2}$$

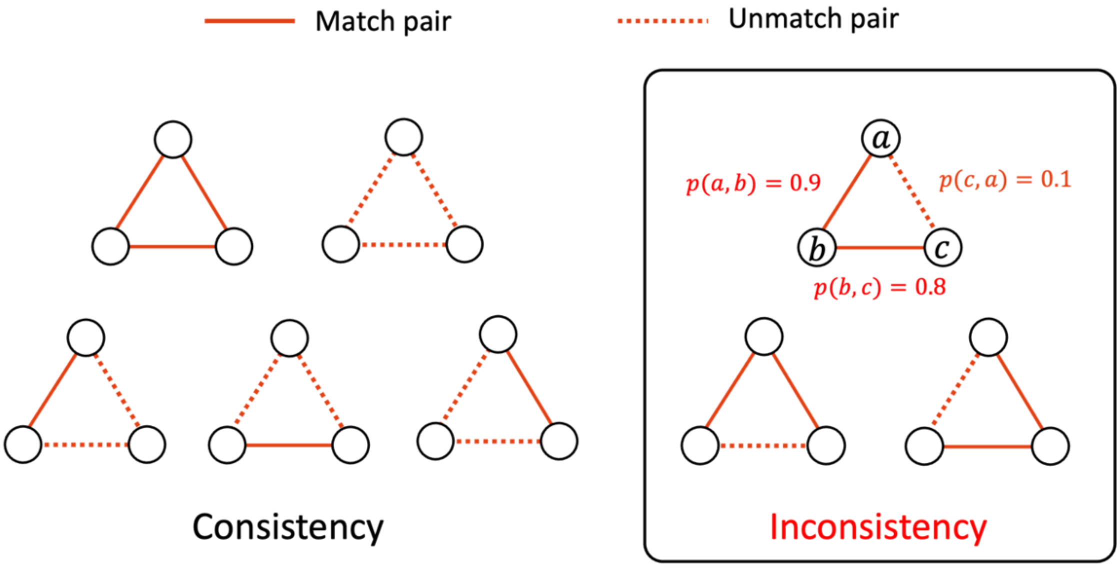


Figure 3. Inconsistent data examples.

Here, cyc(a,b,c) represents the set of (a,b,c) and its cyclic permutations, (b,c,a) and (c,a,b). After calculating the inconsistency scores for all triangles, the algorithm sorts them in descending order. It then populates the query set by selecting pairs from the most inconsistent triangles until the annotation budget m is met. These pairs are considered to be the most '*informative*' samples, where the current model is likely uncertain or incorrect.

Finally, the selected query set is presented to a human annotator to obtain ground truth labels, creating the labeled set Q (Line 8-10). This process aims to maximise the model's performance within the constraints of a limited annotation budget.

Feedback strategy(few-shot/FT)

As illustrated in Figure 1, our research framework follows a cycle of ***selection* → *annotation* → *model update* → *re-inference***. This study compares two primary strategies for feeding the annotated data $\mathcal{L}_m$ back to the model (part 4 of the figure): ***Few-shot*** learning and ***Fine-tuning* (FT)**.
A common aspect of both strategies is that pairs with high inconsistency scores are prioritised for annotation, and a predetermined number of annotated examples are used as '*training data*'. While each strategy requires specific settings for sample sizes and hyperparameters, the configurations used in our experiments will be detailed in the next chapter.

In the Few-shot strategy, several examples from this training data (e.g., pairs with the highest inconsistency scores) are selected and embedded into the prompt as contextual information during inference. This adjusts the model's responses for a specific task without altering its weights. In contrast, Fine-tuning (FT) uses the training data to directly update the weights of the base LLM, thereby generating a model specialised for the task.

In both strategies, to ensure a fair performance evaluation, the training data used for feedback and the evaluation data used for measuring performance are treated as mutually exclusive (disjoint) sets.

Experiments

Dataset

This section describes the experiments conducted to answer RQ1 and RQ2, along with their results. The evaluation was performed using five datasets covering diverse domains.

Persons (Köpcke et al., 2010) is a relatively clean dataset consisting of English personal attributes, used to evaluate baseline matching performance. ***Bibliorecords*** is a Japanese bibliographic dataset provided by public libraries in Japan, used to measure the model's robustness against notational variance. ***Music*** (Köpcke et al., 2010) is an English music dataset with numerous missing values and noisy attributes, used to assess the ability to handle noisy, real-world data.

Furthermore, two datasets from the product domain were selected. ***Amazon-Walmart*** (Hasso Plattner Institute, 2025) targets product data from different e-commerce sites. ***WDC-Products*** (Peeters et al., 2023b) is a benchmark based on actual product data collected at a large scale from web archives. While this benchmark comprises, multiple datasets based on different metrics, our study uses the version ***WDC-Products 80%*** where 80% of the data consists of corner cases (Peeters et al., 2023b).

Table 1 shows an overview of each dataset and the attributes used in the experiments.

Domain (Language)	Dataset	Attributes
Persons (English)	***Persons*** (Köpcke et al., 2010)	name, surname, suburb, postcode
Bibliorecords (Japanese)	***Bibliorecords*** (public libraries in Japan)	title, author, publisher, date
Music (English)	**Music** (Köpcke et al., 2010)	artist, title, album, year, length
Products (English)	***Amazon-Walmart*** (Hasso Plattner Institute, 2025)	title, price, brand, modelno
Products (English)	**WDC-Products** 80% (Peeters et al., 2023b)	brand, title, description, price

Table 1. Dataset attributes.

In our experiments, we used a subset of approximately 3,000 records sampled from each of these datasets. We divided each data set into 2,000 records for training and 1,000 records for evaluation. We concatenated the attributes in Table 1 into one string when we give it to LLM in prompts. In addition to those attributes, every record in the datasets has a special attribute 'Cluster ID.' If a pair of records in the same data set has the same Cluster ID, we consider that the records are *matched* in the evaluation phase in our experiment.

Table 2 summarises the statistics for the created evaluation subsets.

Dataset	Records	Clusters	Avg. Size	Matches
Persons	1000	637	1.57	1570
Bibliorecords	1001	292	3.43	1817
Music	1002	491	2.04	888
Amazon-Walmart	1000	963	1.04	38
WDC-Products	1000	500	2.00	500

Table 2. Experimental dataset statistics.

Experimental setting

Given a matcher, a data set, a sampling strategy with the annotation budget m and a feedback strategy, our experimental workflow proceeds as follows:

1. Blocking: We apply Algorithm 1 to the set of pairs of data items from the dataset. Here, we employ an embedding-based nearest neighbor search. Specifically, using the embedding vectors obtained from text-embedding-ada-002, we utilise FAISS's (Johnson et al., 2019) IndexFlatL2 to identify the top 10 candidates ($k = 10$) for each record based on Euclidean distance (L2).
2. Measuring the performance of the initial matcher f_0: We computer F1 scores of applying f_0 to the set of blocked pairs.
3. Apply the sampling strategy with m to the set of blocked pairs to select m pairs.
4. Update the model f_0 with a feedback strategy p to obtain f_1
5. Measuring the performance of the initial matcher f_1 .

Implementation details are as follows:
Candidate Pair Filtering (Blocking): To filter candidate test pairs, we employ an embedding-based nearest neighbor search. Specifically, using the embedding vectors obtained from text-embedding-ada-002, we utilise FAISS's (Johnson et al., 2019) IndexFlatL2 to identify the top 10 candidates (k=10) for each record based on Euclidean distance (L2).

Model and Fine-tuning: The base model used is gpt-4o-mini-2024-07-18. For fine-tuning, we use the OpenAI API. The annotation results, containing record pairs and their labels, are converted into a JSONL format with system, user, and assistant roles to create the training data. We fine-tune the model for 3 epochs with representative hyperparameter settings of a batch size of 1 and a learning rate of 1.8.

Baselines and comparative methods

In the experiment, we provided feedback to the LLM as follows:

1. ***Fine-tuning (FT):*** This method directly updates the LLM's parameters using pairs detected through the transitivity inconsistency described previously. The training data was created by converting the annotated pairs into a JSONL format with system, user, and assistant roles (see Table 3 for an example).

Component	Prompt Text
System	You are expert music entity matcher. Answer 'Yes' (≥0.5) or 'No' (<0.5) + similarity score (0.0-1.0)
User	Music 1: 'Valley of the Kings (Mysteries of Egypt)' by Sam Cardon Music 2: 'Valley of the Kings' by Sam Cardon
Assistant	Yes Score: 1.0

Table 3. Fine-tuning data examples

2. **Few-shot:** This method presents a small number of correct examples (in-context learning examples) within the LLM's prompt. This allows the model to learn the task definition and output format from the context. In this study, we prioritise and embed difficult cases considered to have high learning value, such as instances where the model previously caused a transitivity violation. Specifically, we incorporate four inconsistent triangles as few-shot examples (see Table 4 for an example).

Component	Prompt Text
System	You are expert music entity matcher. Answer ‘Yes’ (≥0.5) or ‘No’ (<0.5) + similarity score (0.0-1.0)
Few-shot	Model's incorrect predictions: (Record A, record B): Yes, Score: 0.9, (Record B, record C): Yes, Score:0.8, (Record C, record A): No, Score: 0.1 This was a logical violation (inconsistency). correct judgments: Correct Answer for (A, B): `Yes, Score: 1.0` Correct Answer for (B, C): `Yes, Score: 1.0` Correct Answer for (A, C): `Yes, Score: 1.0`
User	Music 1: ‘Valley of the Kings (Mysteries of Egypt)’ by Sam Cardon Music 2: ‘Valley of the Kings’ by Sam Cardon
Assistant	Yes Score: 1.0

Table 4. Few-shot learning examples

To validate the effectiveness of the proposed ***Inconsistency-driven sampling***, we establish the following baseline and comparative methods:

3. ***Zero-shot***: We evaluate the initial performance of the pretrained LLM without applying any active learning or fine-tuning. This serves as the primary performance baseline in our study.
4. ***Random sampling***: This is the most fundamental sampling method, where data is selected randomly from the set of all candidate pairs. In our study, a specified number of pairs are randomly drawn from the list of all candidates and used for fine-tuning. This is used as a comparative standard to evaluate the effectiveness of other active learning strategies.
5. ***Uncertainty sampling*** (Settles, 2010): This method prioritises data for which the model is most uncertain about its prediction. Given a model's match probability $p(x)$ for a record pair x, the uncertainty of that pair is defined by how close its score is to the decision boundary of 0.5. In our research, we calculate uncertainty using the formula: $U(s_i, s_j) = |P(s_i, s_j) - 0.5|$. Pairs with the smallest uncertainty value—i.e., those where the model's judgment is most ambiguous—are preferentially selected and used for fine-tuning.
6. ***Selecting strategy*** (Wang et al., 2025): This method uses a prompt format that presents multiple candidate records to the LLM and asks it to select the ones that match an anchor record. While the reference study prompted the LLM to return only a single matching candidate number from a list, our study allows the LLM to return multiple matching candidate numbers.
7. ***ActiveLLM*** (Bayer et al., 2025): This method treats the LLM itself as an active learning agent, prompting it to self-select the next data points to be annotated. In our implementation, we first randomly sample 200 candidate pairs, from which we prompt GPT-4o to select 32 pairs for annotation, ensuring that previously selected pairs are not chosen again. This process is repeated until the target number of pairs is reached, and the selected data is used for fine-tuning. This approach focuses on explicitly defining the LLM as a data selection expert (an active learner) within the prompt and delegating the selection task to it.

To ensure a fair comparison, the number of unique pairs selected by each sampling strategy is normalised to match that of the Inconsistency-driven sampling method.

Evaluation setup

The primary metric reported is the pairwise F1 score derived from the clusters. The annotation budget is based on 'the number of unique pairs contained within 100 inconsistent triangles, and this number of unique pairs is aligned across all comparative methods to ensure a fair comparison. Additionally, for the proposed method, we compare results with and without adjusting the positive-negative balance. The experimental protocol is based on seed=42 and a batch size of 1, with comparisons primarily conducted over a single iteration. The feedback mechanisms compared are few-shot and fine-tuning (FT). For blocking, we use embedding-based nearest neighbors to filter test pairs (text-embedding-ada-002, k=10, FAISS IndexFlatL2, L2 distance). However, as the main focus of this paper is the comparison of sampling strategies, the primary evaluation axis is the F1 score.

Results

Main performance evaluation

To evaluate the effectiveness of our proposed Inconsistency-driven sampling, we compared its performance against baseline and existing active learning methods across multiple datasets (Bib, Music, Person, Amazon-Walmart, WDC-product). The F1 scores after sampling 100 inconsistent triangles are shown in Table 5. The number of unique pairs for this budget was approximately 450 for the Bib, Music, Person, and WDC datasets, and approximately 200 pairs for the Amazon-Walmart dataset.

Table 5 shows the model performance after fine-tuning (FT) with the data selected by each sampling strategy. The experimental results confirm that our proposed method consistently achieves performance comparable or superior to that of existing methods (Zero-shot, Random, Uncertainty, ActiveLLM) across multiple data types.

Strategy	Persons	Bibrecords	Music	Amazon-Walmart	WDC-Products
Zero-shot (default)	0.8849	0.9894	0.9680	0.8182	0.6103
Random	0.9803	0.9688	0.9833	0.7170	0.5421
Uncertainty	0.9955	**0.9932**	0.9855	**0.9459**	0.6299
ActiveLLM	0.9745	0.9929	0.9916	0.1242	0.1879
Selecting	0.9070	0.9677	0.9887	0.3684	0.4336
Ours (Few-shot)	0.9305	0.9826	0.9787	0.9067	0.6126
Ours (Fine-tuning)	**1.0000**	**0.9932**	**0.9960**	0.8861	**0.6820**

Table 5. Experiment: Comparison of f1 value for different sampling strategies (100 triangles)

On the Bib dataset, FT with the proposed method achieved an F1 score of 0.9932, which was on par with Uncertainty sampling, the highest-performing comparative method. On the Music dataset, the proposed method recorded an F1 score of 0.9960, clearly outperforming all other methods. Similarly, on the WDC-product dataset, the proposed method recorded the highest F1 score of 0.6820.

Conversely, on the Amazon-Walmart dataset, while the uncertainty method showed the highest performance (0.9459), the performance improvement from our method was not as high as in other datasets. We will analyse the cause of this in the following section.

Analysis of the quantity of inconsistent triangles

To investigate the reasons for the performance variance of the proposed method across different datasets, we analysed the '*quantity*' and '*quality*' of inconsistent triangles within each dataset.

Relationship between the number of inconsistent triangles and performance

Table 6 shows the total number of inconsistent triangles detected from all candidate pairs in each training dataset.

Datatype	Total pairs	Total triangles	# Of Inconsistent triangles
Persons (k=15)	21,179	65,820	**418**
Bibrecords (k=15)	23,406	60,356	**295**
Music (k=15)	22,450	15,505	**304**
Amazon-Walmart	14,024	24,032	**20**
WDC-Products	13,034	30,681	**2,089**

Table 6. The number of inconsistent triangles in training data

As shown in Table 6, there is a significant variance in the number of detected inconsistent triangles among the datasets. Notably, the Amazon-Walmart dataset, where the performance of FT was limited in the previous section, contains only 20 inconsistent triangles—an extremely small number compared to the other datasets.

This scarcity of inconsistencies can be attributed to the dataset's structural properties. As detailed in Table 2, the Amazon-Walmart dataset is extremely sparse; it contains 963 distinct clusters for 1,000 records, with an average cluster size of just 1.04 and only 38 total matching pairs. In such a dataset, where most entities do not have a matching counterpart, the structural opportunity for three entities to form a '*triangle*' of candidate pairs is inherently rare. This result suggests that the effectiveness of the proposed method is contingent upon the sufficient existence of '*inconsistent triangles*' to serve as a learning source within the dataset. In cases like Amazon-Walmart where inconsistencies are sparse, the information content gained from sampling is limited, which may hinder performance improvement. It is also important to note that since our method samples pairs in descending order of the score from Equation 2, the sampled pairs do not necessarily constitute a transitivity violation themselves.

Analysis of the nature of inconsistencies

Furthermore, an analysis of the nature of the inconsistencies revealed that the majority of contradictions detected in the high-precision datasets were caused by a specific pattern: misclassifying a triplet that should be (true, true, true) as (true, true, false). This is, in other words, an error caused by a ***false negative***. This finding suggests that our method can effectively detect the '*unknown-unknown*' errors that the model makes with high confidence.

Discussion

In this section, we explicitly address the research questions proposed in the Introduction based on our experimental results.

RQ1: To what extent is the inconsistency-driven approach effective for LLM-based entity matching?

Our experiments substantiate the claim that inconsistency-driven sampling is universally effective. As summarised in Table 5, the proposed method achieved superior F1 scores compared to baselines across all four domains (Bibliographic, Music, Person, and WDC Product). Most notably, on the Person dataset, the inconsistency-driven approach refined the model to achieve a perfect F1-score of 1.0. This empirically proves that prioritising samples violating transitivity allows the model to correct its internal '*unknown-unknown*' errors more effectively than random or standard uncertainty sampling.

RQ2: Between few-shot and FT, which feedback strategy is more effective, and under which conditions or datasets?

Regarding the comparison of feedback strategies, our analysis reveals that Fine-Tuning (FT) is generally more effective, but the optimal strategy depends on dataset characteristics. While the Introduction noted the general superiority of FT, our detailed results clarify the boundary conditions:

- Dominance of FT: FT with inconsistency-derived samples yielded the highest performance on the majority of datasets (e.g., Bibliographic, Music, Person, and WDC Product), confirming its robustness for general entity matching tasks.
- The '*sparse inconsistency*' condition; however, a key finding is the exception observed in datasets with fewer inconsistent triangles (e.g., Walmart-Amazon). In these specific cases, few-shot prompting outperformed FT. This indicates that when high-value contradictory examples are scarce, directly providing them in the prompt is more data-efficient than updating model parameters.

Limitations and future work

While the effectiveness of our proposed method has been demonstrated, several limitations and directions for future work exist.

First, the efficacy of inconsistency sampling is contingent upon the sufficient existence of inconsistent triangles among candidate pairs. If a dataset is extremely clean or the relationships between entities are sparse, opportunities for inconsistency detection may decrease, potentially limiting the information gain from sampling. Many mainstream entity matching datasets are structured around one-to-one matching or contain only a few matching pairs per entity, scenarios where the advantages of our method may be less pronounced.

Second, our evaluation relies on a commercial LLM API, which is susceptible to external factors such as performance fluctuations due to unannounced model updates. However, the active learning framework itself is adaptable to other LLMs.

Third, our results suggest that transitivity violations primarily arise from cases where a pair that should be true is incorrectly predicted as false. The current sampling method, which considers all pairs within high-scoring inconsistent triangles as candidates, is inefficient as it may include pairs that the model already correctly classifies. A strategy to resolve this would be to prioritise sampling the pairs that directly cause the inconsistency, specifically those with false predictions that violate the transitive property (e.g., predicting the pair (A, C) as false when (A, B) and (B, C) are predicted as true). In the future, we plan to implement this pinpointed sampling strategy for LLMs and verify whether it can maximise model performance with an even smaller annotation budget.

Conclusion

In this research, we proposed a novel active learning strategy for LLM-based entity matching that focuses on transitivity inconsistencies. The results of our evaluation experiments across multiple datasets demonstrate that this method stably outperforms or achieves comparable performance to existing methods like random sampling and uncertainty sampling. When combined with fine-tuning as a feedback mechanism, its effectiveness in improving model performance was particularly confirmed.

These findings substantiate the validity of using the logical consistency of an LLM's judgments as a sampling criterion and suggest a new direction for future active learning research. Future work will involve expanding our validation by applying the method to more models, testing it on larger and more diverse datasets, and exploring sampling strategies that incorporate other logical constraints.

Acknowledgements

This work was supported in part by JST CREST (JPMJCR22M2), Grants-in-Aid for Scientific Research (22H00508).

About the authors

Kota Okayama is a bachelor student at College of Knowledge and Library Sciences, University of Tsukuba. Currently, he is working on human-AI collaboration in entity matching. He can be contacted at: kota.okayama.2025b@gmail.com

Hiroyoshi Ito is an Assistant Professor at the University of Tsukuba's Center for Artificial Intelligence Science and the Faculty of Library, Information and Media Science. He obtained his Doctor of Engineering degree from the University of Tsukuba in March 2020. His research focuses on Human-in-the-Loop machine learning, graph data mining, and time-series data analysis. He can be contacted at: ito@slis.tsukuba.ac.jp

Atsuyuki Morishima is Professor at Institute of Library, Information and Media Science and Center for Artificial Intelligence Reseach (C-AIR) at University of Tsukuba, Japan. His research interests include computational division of labor, data-centric human-machine computations, data integration, and digital libraries. He has contributed to conferences and journals in diffirent communities such as databases, human computation, digital libraries and information science. He can be contacted at: morishima-office@slis.tsukuba.ac.jp

References

Barlaug, N., & Gulla, J. A. (2021). Neural networks for entity matching: A survey. ACM Computing Surveys, 15(3), 1–34. https://doi.org/10.1145/3442200

Bayer, M., Lutz, J., & Reuter, C. (2025). ActiveLLM: Large Language Model-based Active Learning for Textual Few-Shot Scenarios. arXiv preprint arXiv:2405.10808.

Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S. E., & Widom, J. (2009). Swoosh: a generic approach to entity resolution. The VLDB Journal, 18(1), 255-276.

Chung, Y., Haas, P.J., Upfal, E., Kraska, T. (2019). Unknown examples & machine learning model generalization.

Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. In IIWeb (Vol. 3, pp. 73-78).

Das, S., G.C., P.S., Doan, A., Naughton, J.F., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V., Park, Y. (2017). Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services. In: Proceedings of the 2017 ACM International Conference on Management of Data. pp. 1431–1446. https://doi.org/10.1145/ 3035918.3035960

Gokhale, C., Das, S., Doan, A., Naughton, J.F., Rampalli, N., Shavlik, J., Zhu, X. (2014). Corleone: Hands-off crowdsourcing for entity matching. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. pp. 601–612. https://doi.org/10.1145/2588555.2588576

Hasso Plattner Institute. (2025). Amazon-Walmart Product Matching Dataset. Retrieved from https://hpi.de/naumann/projects/repeatability/datasets/amazon-walmart-dataset.html (Accessed 2025-08-29).

Huang, Z. (2024). Disambiguate Entity Matching using large language models through Relation Discovery. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (pp. 5551–5555). Association for Computing Machinery. https://doi.org/10.1145/3665601.3669844

Ito, H., Koizumi, T., Yoshimoto, R., Fukushima, Y., Harada, T., & Morishima, A. (2025). Inconsistency-driven approach for human-in-the-loop entity matching. Information Research an International Electronic Journal, 30(iConf), 1024–1038.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association, 84(406), 414-420.

Ji, K., Chen, J., Gao, A., Xie, W., Wan, X., & Wang, B. (2025). Unlocking LLMs' Self-Improvement Capacity with Autonomous Learning for Domain Adaptation. In Findings of the Association for Computational Linguistics: ACL 2025 (pp. 21051–21067). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-acl.1084

Johnson, J., Douze, M., Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data 7(3), 535–547.

Köpcke, H., Thor, A., & Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. Proceedings of the VLDB Endowment, 3(1-2), 484-493.

Osawa, N., Ito, H., Fukushima, Y., Harada, T., Morishima, A. (2021). Bubble: A quality-aware human-in-the-loop entity matching framework. In: The 5th IEEE Workshop on Human-in-the-loop Methods and Future of Work in Big-Data (IEEE HMData2021). pp. 3557–3565. https://doi.org/ 10.1109/BigData52589.2021.9672002

Peeters, R., Bizer, C. (2022). Supervised contrastive learning for product matching. In: Companion Proceedings of the Web Conference 2022. pp. 248–251 https://doi.org/10.1145/3487553.3524254

Peeters, R., Der, R. C., & Bizer, C. (2023). WDC Products: A Multi-Dimensional Entity Matching Benchmark. arXiv preprint arXiv:2301.09521.

Peeters, R., Steiner, A., & Bizer, C. (2023). Entity matching using large language models. arXiv preprint arXiv:2310.11244.

Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., Huang, Q., Liden, L., Yu, Z., Chen, W., & Gao, J. (2023). Check Your Facts and Try Again: Improving large language models with External Knowledge and Automated Feedback. arXiv preprint arXiv:2302.12813.

Settles, B. (2010). Active learning literature survey. In: Active Learning Literature Survey. University of Wisconsin-Madison, https://minds.wisconsin.edu/bitstream/handle/1793/60660/TR1648.pdf

Steiner, A., Peeters, R., and Bizer, C. 2025. Fine-tuning large language models for Entity Matching. arXiv preprint arXiv:2409.08185.

Takashi, H., Yukihiro, F., Sho, S., Misato, T., Ryuji, Y., Atsuyuki, M. (1993). Advancement of bibliographic identification using a crowdsourcing system. Proceedings of the 9th Asia-Pacific Conference on Library & Information Education and Practice (A-LIEP 2019) pp. 71–82.

Wang, T., Chen, X., Lin, H., Chen, X., Han, X., Sun, L., Wang, H., & Zeng, Z. (2025). Match, Compare, or Select? An Investigation of large language models for Entity Matching. In Proceedings of the 31st International Conference on Computational Linguistics (pp. 96–109). Association for Computational Linguistics.

Zhu, Y., Liu, H., Wu, Z., Du, Y. (2020). Relation-aware neighborhood matching model for entity alignment. https://arxiv.org/abs/2012.08128

Appendix

Full prompt templates

The following Table 7 is the complete prompt template used for the pairwise entity matching task. The interaction was structured using a System Message and a User Message, standard for the OpenAI API.

Component	Prompt Text
System	‘You are an expert at determining whether two product records refer to essentially the same product’. ‘First, please clearly answer 'Yes' if you believe the two product records refer to the same product, or 'No' otherwise’. ‘Next, provide a confidence score from 0.0 (completely different) to 1.0 (completely identical) indicating your certainty in this judgment’. ‘Your judgment must strictly follow these rules:’ ‘- If the confidence score is 0.5 or higher, your answer must be 'Yes”. ‘- If the confidence score is below 0.5, your answer must be 'No'.’
User	‘Please determine whether the following two product records refer to essentially the same product.’ ‘Product 1: {record_info_1}’ ‘Product 2: {record_info_2}’ ‘Do these refer to the same product? Answer:’

Table 7. Full prompt templates.

Fewshot prompt template

The following Table 8 is the fewshot prompt template used for the pairwise entity matching task. The text below demonstrates the English translation of the system instructions used in our experiments.

Component	Prompt Text
System	'You are an expert at determining whether two product records refer to essentially the same product'. 'First, please clearly answer 'Yes' if you believe the two product records refer to the same product, or 'No' otherwise'. 'Next, provide a confidence score from 0.0 (completely different) to 1.0 (completely identical) indicating your certainty in this judgment'. 'Your judgment must strictly follow these rules:' '- If the confidence score is 0.5 or higher, your answer must be 'Yes'.' '- If the confidence score is below 0.5, your answer must be 'No'.' 'Furthermore, your judgment must be logically consistent. For example, if A and B are the same, and B and C are the same, then A and C must also be the same. Below are examples of transitivity violations from your past judgments. Please avoid such inconsistent judgments.' '[Example of Inconsistent Judgment 1]' 'Record A: {fewshot_record_info_1}' 'Record B: {fewshot_record_info_2}' 'Record C: {fewshot_record_info_3}' 'Your previous judgment was:' • 'Pair (A, B) → Yes (Score: 1.00)' • 'Pair (B, C) → Yes (Score: 1.00)' • 'However, Pair (A, C) → No (Score: 0.00)' 'This contradicts the transitivity rule (If A=B and B=C, then A=C)'. 'The correct judgment is:' • 'Pair (A, B) → Mismatch' • 'Pair (B, C) → Match' • 'Pair (A, C) → Mismatch' '[Example of Inconsistent Judgment 2]' … '[Example of Inconsistent Judgment 3]' …
User	'Please determine whether the following two product records refer to essentially the same product'. 'Product 1: {record_info_1}' 'Product 2: {record_info_2}' 'Do these refer to the same product? Answer:'

Table 8. Fewshot full prompt templates.

Input entity record format

The entity records were transformed from their original structured format into a plain text string before being inserted into the {record_info_1} and {record_info_2} placeholders of the User Message

The general structure for a single entity record is:

record_info	[Column Name]: [Value]\n [Column Name]: [Value]...

The record in the prompt was generated from the source data using the following exact string representation:

Specific Example (from the WDC Product Dataset):

Record_info_example	Product Name: 16 GB USB-Stick Kingston Data Traveller G4 (USB3.0) \n Brand: Kingston\n Description: None\n Price: 7.90