Learning from Natural Language Explanations for Generalizable Entity Matching

Anonymous ACL submission

Abstract

Entity matching is the task of linking records from different sources that refer to the same real-world entity. Past work has primarily treated entity linking as a standard supervised learning problem. However, supervised entity matching models often do not generalize well to new data, and collecting exhaustive labeled training data is often cost prohibitive. Further, recent efforts have adopted LLMs for this task in few/zero-shot settings, exploiting their general knowledge. But LLMs are prohibitively expensive for performing inference at scale for real-world entity matching tasks.

As an efficient alternative, we re-cast entity matching as a conditional generation task as opposed to binary classification. This enables us to "distill" LLM reasoning into smaller entity matching models via natural language explanations. This approach achieves strong performance, especially on out-of-domain generalization tests (↑10.85% F-1) where standalone generative methods struggle. We perform ablations that highlight the importance of explanations, both for performance and model robustness.

1 Introduction

Entity matching, also known as *record linkage* or *data deduplication*, refers to matching records from different sources which refer to the same underlying entity, in the absence of unique identifiers. This is a practically important task across a diverse set of domains, e.g., database management, healthcare, customer relationship management, and financial services; in such applications, normalizing entities to realize a unified view of data is imperative.

Most prior work on entity matching has adopted supervised techniques, training a model to link entities within a particular domain. Performing pair-wise comparison on all record pairs is computationally prohibitive, especially on large scale datasets; typical entity resolution pipelines therefore perform *blocking* followed by *matching* (Li et al., 2020; Wang et al., 2023a). The former step entails identifying candidate record pairs which may reference the same entity, while in the latter one attempts to infer whether this candidate is indeed a match.

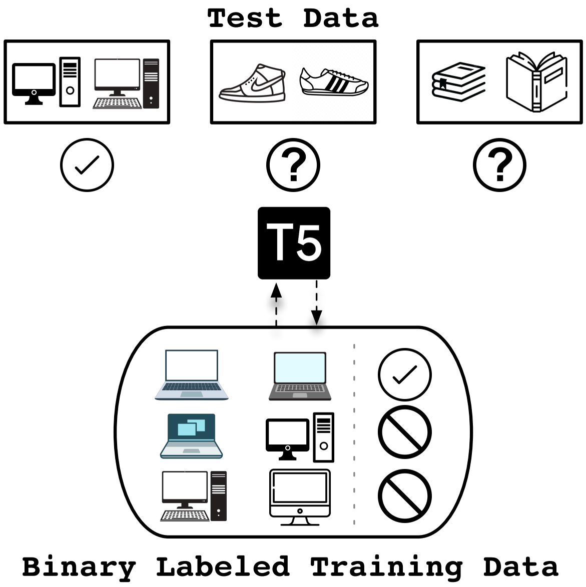

Figure 1: An example of the generalization problem in entity matching: A model trained on a dataset of computers (e.g., WDC-Computers) is tested on instances taken from a corpus comprising shoes (WDC-Shoes).

Assuming a supervised setting for this task is limiting in a few key ways. First, collecting human supervision is inherently expensive. Second and relatedly, training an entity matching model in one "domain" (in this work, a domain is a product category) via explicit supervision will yield a model which is unlikely to readily transfer to other domains. For example, a model trained to match camera models based on descriptions is unlikely to generalize well to linking laptops (nevermind non-electronics). But collecting annotations linking products in all possible categories is not feasible. This has motivated work on transferable models for entity matching across domains (Trabelsi et al., 2022; Tu et al., 2022c,a; Chai et al., 2023).

One way to address the generalization problem may be to use general-purpose LLMs "zero-shot", via prompting and/or lightweight fine-tuning. Given the generality of such models, it is intuitive that they may be more robust to domain shifts when matching entities. Moreover, an as-yet unexplored potential benefit of LLMs for this task is their ability to provide (natural language) "reasoning" for their outputs; this may permit fast manual verification of linkages, and therefore instill confidence in model outputs. Aside from this, we later show that the richer signal in generated label "rationales" (or *explanations*) allows for improved model distillation, consistent with recent findings on other tasks (Ho et al., 2022).

A downside of LLMs is inference cost; applying such models to very large datasets—and continuously to new data as it is produced—is expensive. A comparatively tiny database with just one-thousand entities can yields a million (1k $\times$ 1k) candidate pairs, translating to thousands of dollars in inference costs.[1] We therefore explore *model distillation for entity matching*. In particular, we elicit "reasoning" alongside outputs for entity matching tasks from massive LLMs, and use this to train a modestly sized LM for entity matching such that it can also provide supporting rationales.[2] We show that despite its small size, the resultant model achieves strong performance. Moreover, our ablations highlight the importance of rationalization for robust entity matching, i.e., generalization.

Our contributions are as follows. (1) We frame entity matching as a conditional generation task and show that relatively small seq2seq models perform comparably to non-generative models when tested on in-domain instances. However, both approaches suffer significant loss in performance when tested on out-of-domain instances. (2) We show how augmenting entity matching training datasets with chain-of-thought style reasoning (explanations) obtained from larger models results in significant gains on out-of-domain instances. (3) We perform comprehensive ablations on LLM-generated "explanations" to tease out which aspects of these explanations affect downstream model performance. These findings may have implications for other tasks.

[1] openai.com/pricing

[2] This is a type of distillation, but differs from traditional approaches (Hinton et al., 2015) in that we are distilling only "reasoning" abilities, and not capabilities on the task itself.

	Flan-T5 (*base*)	DITTO (*RoBERTa-base*)	Mistral-7B LLM (*Instruct*)
Training Method	**Supervised**	**Supervised**	**ICL Few-shot**
Abt-Buy	**89.92**	89.33	31.11
Amazon-Google	**76.23**	75.58	25.54
Walmart-Amazon	**87.40**	86.76	18.53
Beer	93.33	**94.37**	32.91
iTunes-Amazon	93.09	**97.06**	41.88
WDC-Computers	**92.08**	91.70	43.27
WDC-Cameras	**91.25**	91.23	45.31
WDC-Watches	93.72	**95.69**	53.94
WDC-Shoes	**90.20**	88.07	51.64

Table 1: Comparison of performance (F-1 scores) for prior work (Li et al., 2020) with recent generative models (Chung et al., 2022) under full supervision (except on Mistral-7B LLM) on the task of entity matching under binary labeled (BL) data.

2 Entity Matching via Text Generation

We treat entity matching as a conditional text generation task. For a dataset of N entity pairs $x_i = (\texttt{entity\_a}_i, \texttt{entity\_b}_i)$, we model the probability of generating classification label (e.g., "match"/"no match") as a string $y_i = < y_i^1, y_i^2 \cdots y_i^T >$, conditioned on a context string $\mathcal{C}_i$. Formally:

$$p_{\text{LM}}(y_i|\mathcal{C}_i, x_i) = \prod_{t=1}^{T} p(y_i^t|\mathcal{C}_i, x_i, y_i^{1\cdots t-1})$$

This is the standard conditional language modeling objective. During training, we use "teacher-forcing", i.e., condition production of outputs ("match" or "not") on reference prefixes.

2.1 Data

We use 9 publicly available entity matching datasets (Köpcke et al., 2010; Konda et al., 2016) used for evaluation in similar prior work (Li et al., 2020; Peeters and Bizer, 2023a). These datasets span several domains, allowing us to assess out-of-domain performance by testing a model trained on one type of data on examples from a another. Each dataset contains entity pairs from structured tables. We follow the *input* linearization strategy and train/validation/test splits from Li et al. (2020). Under this linearization scheme each input candidate entity pair is serialized as a sequence of tokens:

$$\texttt{[entity}_a\texttt{] [COL] <attr> [VAL] ...}$$
$$\texttt{[entity}_b\texttt{] [COL] <attr> [VAL]...}$$

In our generative setting, a single training instance

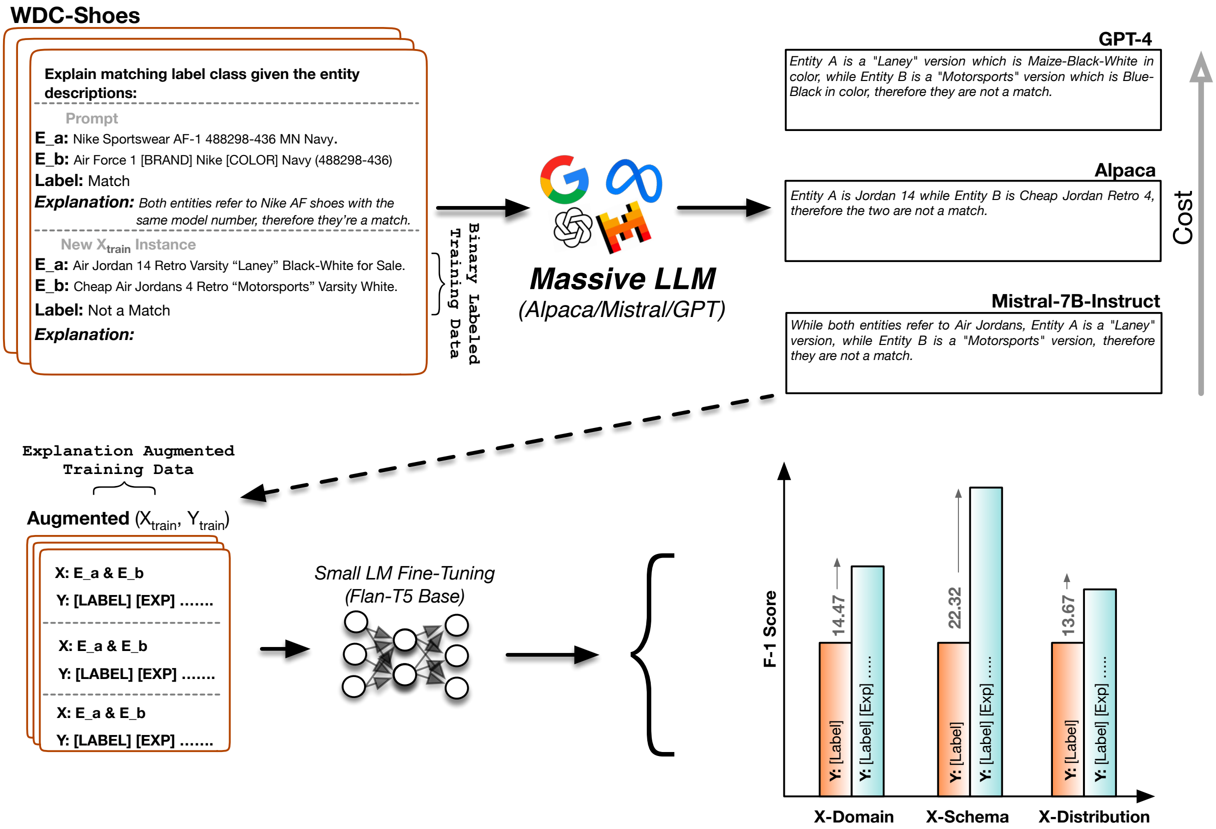

Figure 2: We propose augmenting binary labeled (BL) training data of entity matching datasets with Chain-of-Thought style natural language explanations from large models before fine-tuning smaller, more robust generative models. We use the time needed to generate explanation-augmented (EA) training data on a typical Amazon EC2 P3 instance as a proxy for cost in case of Mistral (Jiang et al., 2023) and Alpaca (Taori et al., 2023) models, and the total cost of OpenAI's API usage in case of GPT-* models. Using this approach, we realize significant performance gains in a variety of out-of-domain test settings.

then becomes a pair of input text with entity attributes, and a linearized output target string[3]:

```
Input [entity_a] [COL] <Title> [VAL] Nike Air
Jordans 2007 ... [entity_b] [COL] <Title> Air
Jordans by Nike [COL] <MANUF_YEAR> [VAL] 2007
...
Target  Match
```

We provide additional full length examples and dataset-specific instances in Appendix B.

2.2 Small LMs, SOTA Performance

We start by evaluating baseline generative models to standard datasets. Table 1 summarizes our findings from these experiments. Generally, we find that even smaller generative models (e.g., FlanT5-base) perform comparably to (and even occasionally outperform) their non-generative counterparts (e.g., DITTO). We also provide results from zero/ICL few-shot experiments using much larger generative models (1B+ parameters) in Appendix E. However, deploying such large models at scale would be prohibitively expensive. Therefore, we focus on smaller models in this work.

To quantify performance on *out-of-domain* data, we consider three experimental settings representative of practical conditions under which entity matching models may be deployed.

Cross Domain Train the model on entity pairs belonging to one domain (e.g., consumer electronics products) and test its performance on another domain (e.g., shoes). Training on the Amazon-Google dataset and testing model performance on WDC-Shoes is one example of this setting.

Cross Schema Entities in the test data may have different attributes, not seen in training, even if the data is from the same domain and derived from the same source. Datasets used to test cross-schema robustness are *not* mutually exclusive from (and may overlap with) cross-domain train-test data pairs.

[3] DITTO (Li et al., 2020) follows a non-generative approach and therefore does **not** require linearized strings as *output targets*.

Type	Training Data	Tested On	F-1 (BL)	F-1 (EA_{Alpaca})	F-1 ($EA_{Mistral}$)	$\nabla(EA_{Mistral\text{-}7B} - BL)$ ($\uparrow$)
X-Domain	Amazon-Google	Beer	70.27	90.80	92.30	22.03
	Abt-Buy	Beer	68.86	85.11	89.66	21.01
	Walmart-Amazon	Beer	77.77	85.62	89.65	11.88
	WDC-Computers	WDC-Shoes	69.95	76.16	79.18	9.23
		WDC-Watches	80.07	87.23	87.02	6.94
		WDC-Cameras	73.26	91.26	93.77	20.57
	WDC-Shoes	WDC-Computers	67.90	84.01	84.13	16.23
		WDC-Watches	70.34	81.49	84.89	14.55
		WDC-Cameras	73.26	82.27	84.74	11.48
	WDC-Watches	WDC-Computers	73.37	85.43	86.20	12.83
		WDC-Shoes	67.26	80.99	81.70	14.44
		WDC-Cameras	82.59	88.47	89.96	7.37
	WDC-Cameras	WDC-Computers	76.33	86.92	87.71	11.38
		WDC-Watches	74.21	80.20	81.77	7.55
		WDC-Shoes	69.15	78.52	78.04	8.89
X-Schema	iTunes-Amazon	Amazon-Google	21.29	43.45	44.61	23.32
		Walmart-Amazon	20.04	41.81	43.09	23.05
	Walmart-Amazon	iTunes-Amazon	51.72	72.19	75.63	23.91
	Amazon-Google		72.22	91.25	91.21	18.99
X-Distribution	Abt-Buy	Amazon-Google	22.25	38.88	41.42	19.17
		Walmart-Amazon	25.77	46.04	45.09	19.32
	Amazon-Google	Abt-Buy	26.72	49.73	44.64	17.92
		Walmart-Amazon	33.10	47.22	51.61	18.51
	Walmart-Amazon	Abt-Buy	63.75	72.84	67.52	3.77
		Amazon-Google	52.05	55.71	60.20	7.97
	WDC-All	Abt-Buy	69.16	76.58	76.44	7.28
		Amazon-Google	46.12	56.12	59.13	13.01
		Walmart-Amazon	64.09	75.55	76.37	12.28

Table 2: Comparison of FlanT5-base performance when trained without (BL) *and* with explanation-augmented (EA) training data. Broadly, we observe significant gain in model performance when trained with chain-of-thought style explanations elicited from large language models.

Cross Distribution Train and test the model on the same domain (e.g., consumer electronics products) but on entity pairs derived from different sources. For example: Train on Walmart-Amazon dataset, test on the entity pairs of Abt-Buy data.

In every setting we observe, unsurprisingly, degraded model performance ($F\text{-}1_{(BL)}$ in Table 2) compared to in-domain test sets (Table 1). For instance, a model trained on a dataset of WDC-Cameras suffers a drop of ~15 points when tested on a dataset of WDC-Computers. We provide additional results in Appendix D for non-generative models under this cross testing framework. Broadly, consistent with prior work (Tu et al., 2022b), we find that non-generative models fare poorly when tested on out-of-domain data.

We emphasize here that the aforementioned settings frequently occur and are a representative of the practical use-cases of entity matching models. It is often cost-prohibitive to collect and annotate data in large volumes for training domain, distribution, or schema-specific models.

2.3 Eliciting explanations from LLMs to improve smaller LMs

To improve out-of-domain model performance under our testing framework, we propose augmenting the binary labeled training data (BL) used to fine-tune small generative models with *Chain-of-Thought* (CoT) style reasoning explanations (Wei et al., 2022) elicited from much larger language models Mistral-Instruct (Jiang et al., 2023) and Alpaca (Taori et al., 2023). We call this explanation-augmented training data (EA).

We use ICL few-shot prompting strategy to elicit meaningful generalizable CoT-style explanations given a pair of input entities and their corresponding matching label. Consider the following illustrative example from the WDC-Shoes dataset used as a prompt to elicit a *CoT-explanation*.

```
Input [entity_a] [COL] <Title> [VAL] Nike Air
Jordans 2007 ... [entity_b] [COL] <Title> Air
```

Jordans by Nike [COL] <MANUF_YEAR> [VAL] 2007
...
Target Match [explanation] Both entities refer to Nike Air Jordans from 2007, therefore they're a match.

Input [entity$_a$] [COL] <Title> [VAL] New Balance 1080 Running [COL] <MANUF_YEAR> [VAL] 2016 ... [entity$_b$] [COL] <Title> NB Fresh Foam X 1080v13 [COL] <MANUF_YEAR> [VAL] 2016 ...
Target Match [explanation] -

The actual prompts we use consist of two ICL examples (one for each target label type), in addition to the new instance for which we want the model to generate an explanation. An author of this paper wrote the explanations for the two ICL examples used in the prompt. We reproduce these prompts in their entirety in Appendix C. For generating *CoT-style* explanations we used publicly available checkpoints for both Mistral-7B-Instruct[4] and Alpaca.[5] We generated explanations with a maximum length of 128 tokens (minimum of 5 tokens) with top$_k$ sampling ($k = 50$) and nucleus sampling ($p = 0.95$). For every dataset, we found that generating explanations took approximately 2-5 seconds for Mistral-7B-Instruct, and 7-12 seconds on Alpaca-based models.

We consider these model generated *CoT-style* explanations analogous to summaries generated by a model given entity text and a corresponding matching label. We then use these explanations to fine a smaller model (FlanT5-base in our case) and observe considerable gains in cross-domain, cross-schema, and cross-distribution performance (Table 2). We find on average the F-1 score under cross-schema setting increases by 22.32, while for cross-domain and cross-distribution setting the average F-1 score increases by 14.47 and 13.67 respectively. In some instances (e.g., a model trained on WDC-Computers $\rightarrow$ tested on WDC-Cameras), we observe that augmenting the training set with *CoT-style* explanations enables OOD performance comparable to in-domain performance[6].

3 Assessing the usefulness of explanations through ablations

We conduct several ablations, both automated (A–E) and through manual human annotations (H_1 and H_2), to assess the usefulness of generated explanations (which appear to improve the performance of *smaller* entity-matching models). Table 3 summarizes findings from our automated ablations. We will use the following instance from the Abt-Buy dataset as a running example to demonstrate ablations A–E:

Entity A: WD Red 3TB SATA III 3.5" Hard Drive - IntelliPower 64MB Cache WD30EFRX
Entity B: CCL Computers WD Red 1 - 64Mo (NAS) HDD
Label: Not a Match

For this instance, the language model (Mistral-7B-Instruct) generates the following explanation:

Generated: While both entities refer to "WD Red" hard drive, Entity A specifically refers to 3TB SATA III 3.5" drive, while Entity B refers to a drive for use in a Network Attached Storage (NAS) and therefore they are not a match.

For each of the following ablations (A–E), we make targeted changes to the original LLM-generated explanations and then retrain the smaller LM to test the corresponding effects.

A. Junk Substituion We start by substituting LLM-generated explanations by sentences comprising random 'junk' tokens, which are generated at random[7] from the English language vocabulary. We retain the original length of the explanation, e.g., in the example above the LLM-generated explanation is substituted with the following text

Substituted: contour fix nap egregious text nimble perhaps

The aim is to assess whether it is the presence of *meaningful* text (rather than *any* text) that leads to performance gains under the above settings. Aggregate performance under Ablation A drops 28.17%, and this is consistent across train-test pairs.

B. Random Token-Drop We alter the LLM-generated explanations by reducing their length. We start by removing all stop-words from the explanation, then randomly drop tokens to further reduce its length until we reduce the total length by half (50%). In the running example, the LLM-generated explanation might be replaced by the following text

Substituted: entities Red "hard 3TB SATA 3.5" use Attached Storage NAS match.

C. TF-IDF Here we attempt to sample tokens from the LLM-generated explanation to assess if the presence of certain key tokens is all that is needed to realize the observed performance gains. We use TF-IDF (Salton and McGill, 1986) as a measure of word importance. Specifically, we treat

[4] huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[5] crfm.stanford.edu/2023/03/13/alpaca.html

[6] Details on reprehensibility are provided Appendix A.

[7] via NLTK (www.nltk.org)

Type	Training Data	Tested On	F-1 ($EA_{Mistral}$)	Ablations A	B	C	D	E
X-Domain	Amazon-Google	Beer	92.30	72.35	88.94	89.33	79.59	89.85
	Abt-Buy	Beer	89.66	62.99	88.81	87.93	70.01	87.50
	Walmart-Amazon	Beer	89.65	75.25	89.30	91.47	76.29	83.33
	WDC-Computers	WDC-Shoes	79.18	71.31	78.04	72.28	75.37	76.92
		WDC-Watches	87.01	80.12	87.06	82.07	82.99	86.12
		WDC-Cameras	93.77	69.15	91.92	89.86	88.56	90.18
	WDC-Shoes	WDC-Computers	84.13	61.75	79.45	72.07	73.29	81.64
		WDC-Watches	84.89	64.76	78.07	77.63	77.62	81.11
		WDC-Cameras	84.74	72.23	77.61	74.95	77.03	82.61
	WDC-Watches	WDC-Computers	86.20	78.18	84.64	84.99	76.05	85.71
		WDC-Shoes	81.70	64.82	83.25	77.71	73.97	78.62
		WDC-Cameras	89.96	85.92	89.36	88.61	85.25	89.18
	WDC-Cameras	WDC-Computers	87.71	75.58	79.50	79.14	79.83	86.99
		WDC-Watches	81.77	73.36	79.67	78.20	79.16	77.21
		WDC-Shoes	78.04	68.60	74.92	74.09	72.60	75.32
X-Schema	iTunes-Amazon	Amazon-Google	44.61	20.89	32.44	35.57	35.58	35.05
		Walmart-Amazon	43.09	17.14	40.49	39.08	41.16	25.64
	Walmart-Amazon	iTunes-Amazon	75.63	49.53	73.33	77.71	60.21	76.41
	Amazon-Google		91.21	69.56	83.65	83.23	73.07	89.97
X-Distribution	Abt-Buy	Amazon-Google	41.42	24.73	36.56	42.04	27.76	39.64
		Walmart-Amazon	45.09	22.01	44.09	43.84	27.84	40.75
	Amazon-Google	Abt-Buy	44.64	23.31	32.05	45.08	31.29	33.61
		Walmart-Amazon	51.61	29.55	35.47	42.54	36.55	45.08
	Walmart-Amazon	Abt-Buy	67.52	62.81	68.99	68.11	64.91	67.55
		Amazon-Google	60.20	51.92	60.47	58.83	54.27	58.84
	WDC-All	Abt-Buy	76.44	68.48	71.28	72.36	70.21	75.51
		Amazon-Google	59.13	49.74	55.49	55.12	50.56	53.99
		Walmart-Amazon	64.09	62.19	73.81	72.43	67.23	75.28
∇ Aggregate comparison against F-1 ($EA_{Mistral}$)				−26.99	−5.57	−5.69	−14.35	−4.98

Table 3: Comparison of FlanT5-base performance when LLM-generated explanations used during model training are ablated under various conditions – **A.** Junk text substitution, **B.** Random reduction in length, **C.** TF-IDF reduction in length, **D.** Substitution with non-instance specific explanation, **E.** Random corruption of tokens in explanation.

entity descriptions and their corresponding labels as *documents*, and LLM-generated explanations as a *summary* of these. We then sample tokens from the explanation based on the TF-IDF scores of individual tokens until we retain 50% of the original length of the explanation. In the running example, the LLM-generated explanation might be replaced by the following text:

```
Substituted: drive to entity refers while 3tb and
are attached both entities for hard iii in match
nas network not red refer sata specifically
storage
```

Perhaps surprisingly, sampling tokens in this way does *not* help, compared to randomly sampling them like as in (B); the performance degradation is about the same (5.57% vs 5.69%; Table 3).

D. Generic Explanations In this ablation we evaluate whether a *dataset-level* (as opposed to instance-level) explanation yields performance gains. These dataset-wide explanations may or may not be model generated. For our experiments, we use the following manually written explanations:

```
WDC-Cameras Based on the description of two
cameras in Entity A and Entity B, they are (or
are not) a match.
WDC-Shoes  Based on the color, brand, size and
make of the two shoes in Entity A and Entity B
respectively, they are (or are not) a match.
iTunes-Amazon  Based on the artist, genre and
song titles, the two entities here are (or are
not) a match.
```

We find that the aggregate performance (Table 3) declines by $\sim$14%, compared to $\sim$25% when we do not use any explanations, and $\sim$27% using junk text as a substitute (Ablation A).

E. Random Corruption Finally, we evaluate the results when we randomly replace half of the tokens in LLM-generated explanation by a reserved token (<unk>) to gauge whether the performance gains observed with explanations owe to the effective additional compute they permit at inference

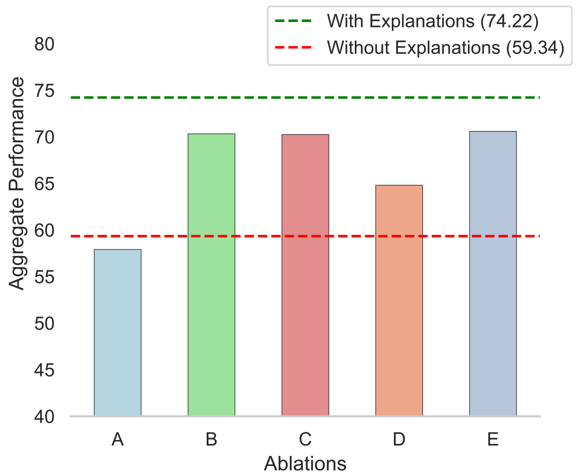

Figure 3: Average F1 on out-of-domain test data when *training* data is ablated under varying conditions.

time. In our example, the LLM-generated explanation is modified to:

Substituted: While <unk> <unk> <unk> to <unk> <unk> <unk> <unk>' hard drive, <unk> <unk> A specifically refers <unk> 3 <unk> SATA III <unk> 3.5 <unk> <unk> <unk> <unk> ity B refers <unk> <unk> drive <unk> <unk> <unk> <unk> <unk> Network <unk> <unk> d <unk> (NAS) <unk> therefore <unk> are not <unk> <unk> match <unk>

While we observe a performance difference on average (Table 3), these differences are inconsistent across settings, contrary to our other ablation results. For instance, under cross-domain setting for `WDC-Cameras` $\rightarrow$ `WDC-Computers`, we observe that Ablation E outperforms both Ablations B and C and is comparable to using unaltered explanations. However, under a cross-schema setting for `iTunes-Amazon` $\rightarrow$ `Walmart-Amazon`, ablation E performs substantially worse than using unaltered explanations. We leave a more comprehensive analysis of this behavior for future work.

In addition to ablations A–E, we conduct two additional experiments with human-interventions to test (1) robustness of models trained with augmented data; and (2) faithfulness of the generated reasoning explanations themselves. Because we generate tens of thousands of explanations (i.e., instance specific explanations for the entire training set for every dataset), collecting human annotations on all instances is cost prohibitive. Instead, we manually select 300 instances from the Abt-Buy dataset to conduct the following two tests.

H_1 Test of Robustness First, we test robustness by randomly selecting 300 entity pairs with a "match" label from the test set. We then make minimal changes to the entity data (descriptions) to convert a "matched" to a "non-matched" pair. These changes are quite minimal, often involving only a token or two (e.g., `Nike`$\rightarrow$`Adidas`) while retaining a majority of token overlap between the entity pair descriptions. This intervention is motivated by the fact that matching models may over-rely on token overlap to classify whether or not the entity pair is match, and whether a trained model is robust to minor perturbations when tested on in-domain data. Consider the following example:

Original: [entity$_a$] Kingston 128GB DataTraveler G3 USB 3.1 Flash drive [entity$_b$] Kingston 128G DT G3 USB 3.1 Flash Drive
Label Match

Edited: [entity$_a$] Kingston 128GB DataTraveler G3 USB 3.1 Flash drive [entity$_b$] Kingston **32G** DT G3 USB 3.1 Flash Drive
Corrected Label Not a Match

Here we have minimally changed the storage capacity of two USB Flash Drives manufactured by the same company, under the same brand/model.

We then run these substituted instances through our models – trained both with ***and*** without LLM-augmented explanations. Our goal here is was to test what percentage of labels correctly flip from "match" to "no-match" in both instances. We're motivated to test this aspect of robustness to determine the degree to which smaller trained models rely on raw token overlap vs the reasoning in LLM-generated explanations.

For the models trained without explanations, we find that $71/300$ (23%) of labels flip, while for the models trained with LLM-augmented explanations, we find that $164/300$ (54%) labels successfully flip to a non-match; this indicates that augmented reasoning in training data makes smaller models more robust to subtle but critical input perturbations.

H_2 Test of Factuality Finally, we investigate the extent to which LLM-generated explanations relate to the underlying entity pair descriptions. To this end we consider generated explanations as analogous to document summaries, i.e., we consider the input entity pair descriptions and their matching label as a *document*, and treat the model generated explanation of the *summary*. We then annotate these explanations for inconsistencies.

Three authors of this paper serve as human annotators and we use the Amazon Mechanical Turk

(MTurk) sandbox as our preferred annotation platform. For every instance, we ask annotators the following two questions related to the types of observed errors in reasoning explanations:

Instrinsic Errors Is the explanation fully derivable from the input entities and their corresponding matching label, irrespective of whether it contains excess information?

Extrinsic Errors Does the explanation contain information in excess of the entity descriptions and their corresponding matching labels? These inconsistencies are often called “hallucinations”.

We collected three annotations per instance and take the majority vote as reference where there is not unanimous agreement. We find that 10.9% of instances contain instrinsic errors, and 15.1% of explanations contain elements unsupported by inputs (“hallucinations”). We observe an inter-rater agreement (Fleiss’s κ) of 0.75 for the question on instrinsic errors and an agreement of 0.86 on the question of extrinsic errors. We provide details on the annotation interface in Appendix F.

4 Related Work

4.1 Deep learning in Entity Resolution

With respect to entity resolution, the core process involves pairwise comparisons to ascertain matching entities. Recent efforts have capitalized on neural methods (including LLMs), including DeepER (Ebraheem et al., 2018), a deep learning-based framework, and DeepMatcher (Mudgal et al., 2018), which exemplifies the integration of deep learning in entity matching. Additionally, active learning strategies have been adapted for entity resolution as detailed in (Kasai et al., 2019).

Other significant contributions include Seq2SeqMatcher (Nie et al., 2019), focusing on sequence-to-sequence matching, and HierMatcher (Fu et al., 2021), which adopts a hierarchical approach. The use of pre-trained language models has also gained traction, as evidenced by methods such as R-SupCon, Ditto, Rotom, and Sudowoodo, discussed in various studies (Brunner and Stockinger, 2020; Peeters et al., 2020; Li et al., 2021; Miao et al., 2021; Wang et al., 2023b). These methods collectively represent the cutting-edge techniques in the realm of entity matching.

Domain Adaptation aims to allow a model trained in one domain to generalize to other domains (Trabelsi et al., 2022; Tu et al., 2022c,a; Sachidananda et al., 2021).

4.2 Reasoning in LLMs

Most recently, Entity Matching via LLMs has shown promising results (Peeters and Bizer, 2023c,b). In these works, both zero-shot and fine-tuning approaches have been explored. Beyond entity matching, in-context learning (ICL) with LLMs has become a dominant strategy, enabling these models to perform tasks with task conditioning and minimal task demonstrations (Brown et al., 2020; Xie et al., 2021). This approach has demonstrated strong performance (Zhao et al., 2021; Liu et al., 2021) and streamlined experimentation with LLMs, as it eliminates the need for model training. However, the adoption of ICL has highlighted the sensitivity of LLMs to prompt selection (Lu et al., 2021; Margatina et al., 2023), making prompt engineering for various tasks a challenging and time-consuming process. Nonetheless, data-driven signals, such as selecting semantically similar demonstrations using text retrievers, have proven to be effective (Lu et al., 2021; Margatina et al., 2023), offering a more systematic approach to prompt engineering.

Chain-of-Thought (CoT) reasoning (Wang et al., 2022; Hoffmann et al., 2022; Chowdhery et al., 2022) has lately emerged as a means to allow LLMs to better perform certain tasks. This approach—which can be elicited via prompting few-shot examples (Kojima et al., 2022)—involves guiding LLMs to generate a sequence of intermediate reasoning steps. Recent efforts have demonstrated the benefits of distilling “reasoning” capabilities in smaller LMs (Shridhar et al., 2023; Wadhwa et al., 2023); our results contribute to this line of work.

5 Conclusions

We proposed a novel model distillation approach to train a small, more-robust model for generalizable entity matching. Eliciting target label rationales from LLMs enables transfer of grounded “reasoning” to the smaller models. Our experiments show this translates to strong performance in diverse settings, outperforming existing models designed for domain adaptation that struggle to generalize. Ablation studies provide insight into the importance of explanation generation for achieving robust matching performance.

Limitations

We have shown that augmenting training data used to train smaller models with natural language explanations elicited from much larger models can yield substantial improvements in out-of-domain test settings. We then assessed the quality and usefulness of said explanations through automated ablations. Finally, we conducted human annotations on a sample of these explanations to quantify error they may contain.

There are some important limitations to these findings. First, we have considered training a model on one domain (or distribution/schema), and then testing it on a set of $N-1$ datasets to evaluate model performance in an OOD setting. This (somewhat extreme) setting sharply exemplifies the sort of domain shift we are interested in studying. But we have not *comprehensively* considered the more traditional OOD setting of training on $N-1$ datasets, and testing on the held out domain (distribution/schema), except while training on `WDC-All` and testing on `Abt-Buy`, `Amazon-Google`, and `Walmart-Amazon`. However, even under the limited circumstances we considered, we saw substantial gains in OOD performance (↑10.86 F-1).

Second, we rely on LLM-generated reasoning explanations to augment our training data. This dependence on externally hosted, proprietary large models could be problematic in certain sensitive domains, for example when working with entity descriptions that contain personally identifiable information (PII) since there is an extensive body of prior research (Hossain et al., 2023; Prakash and Lee, 2023) documenting social biases inherent to LLMs. That said, this dependence is only for *training* data, and one could conceivably use open source LLMs, like we have, capable of CoT in place of proprietary models (e.g. OpenAI).

Third, while we find that distilling CoT-style explanations meaningfully improves small LM performance, our attempts to evaluating the usefulness of said explanations (if any) will require substantial future work. Our ablations do not provide a clear answer as to which aspects of these explanations are useful for downstream performance improvements. For instance, in ablation **D** we use a constant non-instance specific explanation appended to all target outputs (as opposed to instance specific explanation generated from a LLM). In theory, this provides no meaningful ability to classify a given instance over say, junk text. However, we still observe some gains in downstream OOD test performance.

Lastly, we *only* experiment with datasets curated (and sourced) in English and therefore we do not have any insight into the issues that may result in other languages.

Ethical Considerations

Statement of Intended Use Our work broadly relies on open-source datasets derived from e-commerce platforms, where entity attributes consist of heterogeneous descriptive sentences of common everyday consumer products. However, in certain applications of entity resolution like customer profile de-duplication, where entity descriptors involve a human population-level attributes, the underlying data *must* be appropriately de-identified (i.e. anonymized) in the interest of individual privacy. As stated in limitations, we make no attempt to manually edit/oversee the LLM-generated explanations before using them to train smaller LMs, and therefore there is a downstream risk of propagating large model biases.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ursin Brunner and Kurt Stockinger. 2020. Entity matching with transformer architectures-a step forward in data integration. In *EDBT*. OpenProceedings.

Chengliang Chai, Nan Tang, Ju Fan, and Yuyu Luo. 2023. Demystifying artificial intelligence for data preparation. In *Companion of the 2023 International Conference on Management of Data*, pages 13–20.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi

Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *PVLDB*, 11(11):1454–1467.

Cheng Fu, Xianpei Han, Jiaming He, and Le Sun. 2021. Hierarchical matching network for heterogeneous entity resolution. In *IJCAI*, pages 3665–3671.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Tamanna Hossain, Sunipa Dev, and Sameer Singh. 2023. MISGENDERED: Limits of large language models in understanding pronouns. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5352–5367, Toronto, Canada. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. In *ACL*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Pradap Konda, Sanjib Das, Paul Suganthan G. C., AnHai Doan, Adel Ardalan, Jeffrey R. Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeff Naughton, Shishir Prasad, Ganesh Krishnan, Rohit Deep, and Vijay Raghavendra. 2016. Magellan: Toward building entity matching management systems. *Proc. VLDB Endow.*, 9(12):1197–1208.

Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proc. VLDB Endow.*, 3(1–2):484–493.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proc. VLDB Endow.*, 14(1):50–60.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2021. Deep entity matching with pre-trained language models. *PVLDB*, 14(1):50–60.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. *arXiv preprint arXiv:2305.14264*.

Zhengjie Miao, Yuliang Li, and Xiaolan Wang. 2021. Rotom: A meta-learned data augmentation framework for entity matching, data cleaning, text classification, and beyond. In *SIGMOD*, pages 1303–1316.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, and et. al. 2018. Deep learning for entity matching: A design space exploration. In *SIGMOD*.

Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *CIKM*, pages 629–638.

Ralph Peeters and Christian Bizer. 2023a. Entity matching using large language models.

Ralph Peeters and Christian Bizer. 2023b. Entity matching using large language models. *arXiv preprint arXiv:2310.11244*.

Ralph Peeters and Christian Bizer. 2023c. Using chatgpt for entity matching. *arXiv preprint arXiv:2305.03423*.

Ralph Peeters and Christian Bizer. 2023d. Using chatgpt for entity matching.

Ralph Peeters, Christian Bizer, and Goran Glavas. 2020. Intermediate training of BERT for product matching. In *DI2KG@VLDB*.

Nirmalendu Prakash and Roy Ka-Wei Lee. 2023. Layered bias: Interpreting bias in pretrained large language models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 284–295, Singapore. Association for Computational Linguistics.

Vin Sachidananda, Jason Kessler, and Yi-An Lai. 2021. Efficient domain adaptation of language models via adaptive tokenization. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 155–165, Virtual. Association for Computational Linguistics.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073, Toronto, Canada. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. Ul2: Unifying language learning paradigms.

Mohamed Trabelsi, Jeff Heflin, and Jin Cao. 2022. Dame: Domain adaptation for matching entities. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1016–1024.

Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Chengliang Chai, Guoliang Li, Ruixue Fan, and Xiaoyong Du. 2022a. Domain adaptation for deep entity resolution. In *Proceedings of the 2022 International Conference on Management of Data*, pages 443–457.

Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Chengliang Chai, Guoliang Li, Ruixue Fan, and Xiaoyong Du. 2022b. Domain adaptation for deep entity resolution. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 443–457, New York, NY, USA. Association for Computing Machinery.

Jianhong Tu, Xiaoyue Han, Ju Fan, Nan Tang, Chengliang Chai, Guoliang Li, and Xiaoyong Du. 2022c. Dader: hands-off entity resolution with domain adaptation. *Proceedings of the VLDB Endowment*, 15(12):3666–3669.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Runhui Wang, Yuliang Li, and Jin Wang. 2023a. Sudowoodo: Contrastive self-supervised learning for multi-purpose data integration and preparation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1502–1515. IEEE.

Runhui Wang, Yuliang Li, and Jin Wang. 2023b. Sudowoodo: Contrastive self-supervised learning for multi-purpose data integration and preparation. *ICDE*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Appendix

A Experimental settings and reproducibility

We performed all of our experiments on two AWS EC2 P3 instances, each containing 8 NVIDIA V100 (16GB) GPUs. We used the Huggingface library (v4.26.1; Wolf et al. 2020) and publicly available checkpoints of models we used in our experiments. On all datasets except for WDC our best performing models were trained with batch size 16, while for WDC datasets we used a batch size of 8. We use default hyperparameters[8] for model fine-tuning except for learning rate ($10^{-2} - 10^{-6}$), which we vary through hyperparameter tuning. We used the Adam optimizer and set the max epochs to 100 with an early stopping patience of 10 and a validation set F-1 score increase threshhold of 0.02. None of the trained models in any of our experiments required more than 60 epochs.

B Datasets

We select commonly used entity matching datasets in our work. Each dataset is split into training, validation, and test sets using the ratio 3:1:1 – same splits as Li et al. (2020) to provide direct comparisons in our OOD baselines (Table 4):

Abt-Buy This dataset contains product descriptions from e-commerce platforms `Abt.com` and `Buy.com`. A majority of products on either platform can be categorized as consumer electronics. There are a total of $9,575$ instances in the Abt-Buy dataset.

Amazon-Google The Amazon-Google dataset consists mainly of software product offerings e.g. MS Office/Windows. The relevant entity attributes in Amazon-Google include *brand, title* and *price*. There are a total of $11,460$ product pairs.

Walmart-Amazon This is a structured benchmark entity matching dataset in the general consumer products domain containing textual product attributes like *brand, title, model number,* and *price*. Walmart-Amazon consists of $10,242$ product pairs.

iTunes-Amazon Unlike our other datasets, iTunes-Amazon consists of strutured descriptions of songs in the form of textual attributes like *artist, album year,* and *title*. iTunes-Amazon is a relatively small dataset made up of 539 instance pairs.

Beer This dataset contains structured textual attributes of beers from BeerAdvocate and RateBeer. We use the processed version[9] of this dataset with the same train-dev-test splits as Li et al. (2020). There are only 450 pairs in the Beer dataset.

WDC Products The Web Data Commons datasets span a variety of product categories like electronics, apparel, and accessories. WDC provides 4400 manually annotated gold labels from four categories: `computers` $(68,461)$, `cameras` $(42,277)$, `watches` $(61,569)$, and `shoes` $(42,989)$. Each category contains 800 negative and 300 positive *test* pairs. Each instance in all WDC datasets consists of four attributes - *title, description, brand,* and *specTable*.

C Prompts

We use the following prompts as few-shot exemplars corresponding to each dataset *type* to elicit natural language explanations. Inputs and target references are directly extracted from the original training sets while the `explanations` are human-written (by the authors) and were added for the experiments described in section 2.3.

Consumer Electronic Products We use the following prompt for all of the following datasets – Abt-Buy, Amazon-Google, Walmart-Amazon, WDC-Computers, and WDC-Cameras.

```
<s>[INST] Given the following two examples, provide an explanation for the third example for why the two entities do or do not match. [\INST]
Entity A: [NAME] samsung dlp tv stand in black tr72bx [DESCRIPTION] samsung dlp tv stand in black tr72bx designed to fit samsung hlt7288 hlt7288 , hl72a650 , and hl67a650 television sets tempered 6mm tinted glass shelves wide audio storage shelves to accommodate 4 or more components wire management system easy to assemble high gloss black finish [PRICE] 369.0
Entity B: [NAME] samsung tr72b tv stand [DESCRIPTION] glass black [PRICE] 232.14
Label: MATCH
Explanation: Both entities refer to samsung TV stand in black and therefore have substantially similar specifications, therefore they're a match. </s>
Entity A: [NAME] canon high capacity color ink cartridge color ink cl51 [DESCRIPTION] canon high capacity color ink cartridge cl51 compatible with pixma ip6210d , ip6220d , mp150 , mp170 and mp450 printers [PRICE] 35.0
```

[8] `huggingface.co/docs/transformers/model_doc/flan-t5`

[9] `pages.cs.wisc.edu/~anhai/data1/deepmatcher_data/Structured/Beer/exp_data`

Entity B: [NAME] canon pg-40 twin pack black ink cartridge 0615b013 [DESCRIPTION] black [PRICE]
Label: NOT A MATCH
Explanation: Entity A refers to color ink cartridge while Entity B is a blank ink cartridge, therefore they are not a match. </s>

Shoes We use the following prompt for WDC-Shoes. The examples here are randomly selected from the WDC-Shoes training data.

<s> [INST]Given the following two examples, provide an explanation for the third example for why the two entities do or do not match.[/INST]
Entity A: [NAME] Nike Sportswear Air Force 1 - Midnight Navy'en Mens Shoes Nike Navy 488298-436 en
Entity B: [NAME] "Nike Air Force 1 '07 Low midnight navy / white (488298-436)"eu (488298-436) | Bludshop.com" eu
Label: MATCH
Explanation: Both entities refer to Nike Air Force shoes, navy in color with the same model number 488298-436, therefore they're a match.</s>
Entity A: [NAME] "Air Jordan 14 Retro Low "Laney" Varsity Royal/Varsity Maize-Black-White For Sale"en-US Sale | Cheap Jordans 2017"en-US
Entity B: [NAME] "Cheap Air Jordan 4 Retro "Motorsports" White/Varsity Blue-Black Sale"en-US Sale | Cheap Jordans 2017"en-US
Label: NOT A MATCH
Explanation: While both entities refer to cheap Air Jordan shoes, Entity A is a Laney version which is Maize-Black-White in color, while Entity B is a Motorsports version which is Blue-Black in color, therefore they are not a match.</s>

Music We use the following prompt for iTunes-Amazon. The examples here are randomly selected from the iTunes-Amazon training data.

<s> [INST] Given the following two examples, provide an explanation for the third example for why the two entities do or do not match. [\INST]
Entity A: [SONG_NAME] Extra Extra Credit [ARTIST_NAME] Wiz Khalifa [ALBUM_NAME] Flight School [GENRE] Hip-Hop/Rap , Music [PRICE] 0.99 [COPYRIGHT] 2009 Rostrum Records [TIME] 4:03 [RELEASED] 17-Apr-09
Entity B: [SONG_NAME] Extra Extra Credit [Explicit] [ARTIST_NAME] Wiz Khalifa [ALBUM_NAME] Flight School [Explicit] [GENRE] Rap & Hip-Hop [PRICE] 0.99 [COPYRIGHT] 2013 Mad Decent [TIME] 4:03 [RELEASED] April 17 , 2009
Label: MATCH
Explanation: Both entities are songs with the same name, artist and album.</s>
Entity A: [SONG_NAME] Illusion (feat . Echosmith) [ARTIST_NAME] Zedd [ALBUM_NAME] True Colors [GENRE] Dance , Music, Electronic [PRICE] 1.29 [COPYRIGHT] 2015 Interscope Records [TIME] 6:30 [RELEASED] 18-May-15
Entity B: [SONG_NAME] Papercut [feat . Troye Sivan] [ARTIST_NAME] Zedd [ALBUM_NAME] True Colors [GENRE] Dance & Electronic [PRICE] 1.29 [COPYRIGHT] (C) 2015 Interscope Records [TIME] 7:23 [RELEASED] May 18 , 2015
Label: NOT A MATCH
Explanation: While both entities refer to songs with the same artist, they have clearly different names and therefore, are not a match.</s>

Beer We use the following prompt for Beer dataset.

<s> [INST] Given the following two examples, provide an explanation for the third example for why the two entities do or do not match.[\INST]
Entity A: [NAME] Honey Basil Amber [MANUFACTURER] Rude Hippo Brewing Company [STYLE] American Amber / Red Ale [ABV] 7.40
Entity B: [NAME] Rude Hippo Honey Basil Amber [MANUFACTURER] 18th Street Brewery [STYLE] Amber Ale [ABV] 7.40
Label: MATCH
Explanation: Both entities refer to Honey Basil Amber beer with the same ABV, therefore they're a match.</s>
Entity A: [NAME] Brew Kahuna NW Red Ale [MANUFACTURER] Sky High Brewing [STYLE] American Amber / Red Ale [ABV] 5.20
Entity B: [NAME] Brew Bus Detour Series : Rollin Dirty Red Ale - Wood Aged [MANUFACTURER] Cigar City Brewing [STYLE] Irish Ale [ABV] 5
Label: NOT A MATCH
Explanation: Entity A refers to Beer manufactured by Sky High Brewing while Entity B refers to Beer manufactured by Cigar City Brewing, and they have different names, therefore they are not a match.</s>

D OOD Performance in Neural Entity Matching

We conduct baseline experiments using our testing framework (cross-domain, cross-distribution, and cross-schema) on both generative (FlanT5) and non-generative (DITTO – based on RoBERTa) methods. Table 4 summarizes our results. We observe significant decline in performance under both methods, with RoBERTa-based DITTO (Avg F-1: 55.28) faring slightly worse than FlanT5 (Avg F-1: 59.28).

Our results on non-generative models like DITTO are in-line with prior work in the area where Tu et al. (2022b) first highlight the issue of domain adaptation and the challenge of *reusing* labeled source data where there might be a change in distribution or domain at test time.

E Zero-Shot Entity Matching with LLMs

In addition to training and testing smaller seq2seq models we also provide results from few-shot prompting on larger language models (# parameters > 7B). We emphasize here again that in *any* practical entity matching context, deployment of such larger models is infeasible due the sheer number of comparisons involved. For instance, a *small*

Type	Training Data	Tested On	F-1 BL_{DITTO}	F-1 $BL_{FlanT5\text{-}Base}$
X-Domain	Amazon-Google	Beer	70.27	63.10
	Abt-Buy	Beer	68.86	55.29
	Walmart-Amazon	Beer	77.77	59.12
	WDC-Computers	WDC-Shoes	69.95	65.18
		WDC-Watches	80.07	80.98
		WDC-Cameras	73.26	70.51
	WDC-Shoes	WDC-Computers	67.90	65.11
		WDC-Watches	70.34	74.47
		WDC-Cameras	73.26	72.90
	WDC-Watches	WDC-Computers	73.37	75.34
		WDC-Shoes	67.26	67.22
		WDC-Cameras	82.59	81.16
	WDC-Cameras	WDC-Computers	76.33	75.83
		WDC-Watches	74.21	73.92
		WDC-Shoes	69.15	61.73
X-Schema	iTunes-Amazon	Amazon-Google	21.29	21.48
		Walmart-Amazon	20.04	18.75
	Walmart-Amazon	iTunes-Amazon	51.72	50.82
	Amazon-Google		72.22	76.17
X-Distribution	Abt-Buy	Amazon-Google	22.25	19.15
		Walmart-Amazon	25.77	28.99
	Amazon-Google	Abt-Buy	26.72	25.55
		Walmart-Amazon	33.10	23.78
	Walmart-Amazon	Abt-Buy	63.75	58.11
		Amazon-Google	52.05	39.18
	WDC-All	Abt-Buy	69.16	67.22
		Amazon-Google	46.12	41.37
		Walmart-Amazon	64.09	64.88

Table 4: Comparison of OOD test performance under our framework for FlanT5-base (Chung et al., 2022) and non-generative DITTO (Li et al., 2020) when trained on binary labeled (BL) training data. Broadly, we observe significant degradation in model performance under both models.

product catalog of $1,000$ products can, in worst case scenario, lead to $1,000,000$ pair comparisons – this requires efficiency and, as a practical matter, low deployment costs. Nevertheless, we feel it is important to contextualize our work under ICL few-shot settings on LLMs given their current relevance. We use the same prompts as provided in Appendix C, with one example of each class and test five (Taori et al., 2023; Jiang et al., 2023; Almazrouei et al., 2023; Chung et al., 2022; Tay et al., 2023) instruction tuned models.

Table 5 summarizes these results. Generally, we find that all the models we test under-perform trained smaller LMs. We also observe certain behaviors while prompting LLMs where in some cases (see Alpaca tested on the Beer dataset) we get unusually high recall while getting very low precision measurements, indicating that models may excessively rely on token overlap as a proxy for entity matches. This is in line with prior work where Peeters and Bizer (2023d) use ChatGPT for Entity Matching and observe similar behavior. We do not experiment with different prompts and/or chain-of-thought style explanations under these few-shot settings since that is beyond the scope of this work.

F Human Evaluation (H_2)

We conduct Test of Factuality evaluation on Amazon Mechanical Turk (AMT) – a popular platform for workers (both experts and non-experts) to perform "micro-tasks" (in our case, instance annotations) on explanations generated by the Mistral-7B model on 300 instances of the Abt-Buy dataset. Figure 4 illustrates the interface provided to annotators where they're asked the two factuality-related questions and are presented with binary choices.

	Alpaca (7B)			Mistral-7B-Instruct			Falcon-Instruct (7B)			FlanT5-XXL (11B)			Flan-UL2 (20B)		
	P	R	F-1	P	R	F-1	P	R	F-1	P	R	F-1	P	R	F-1
Abt-Buy	12.33	77.61	21.28	16.49	52.6	25.11	14.77	50.81	22.89	15.23	91.30	26.11	85.74	42.41	56.75
Amazon-Google	11.91	89.29	21.02	15.50	72.64	25.54	12.67	70.41	21.48	20.75	80.27	32.98	74.66	48.3	58.65
Walmart-Amazon	10.31	83.81	18.37	10.74	75.40	18.53	11.52	85.36	20.30	18.14	72.09	28.99	92.21	36.88	52.69
Beer	18.91	100.00	31.81	20.01	92.85	32.91	10.58	100.00	19.14	9.65	89.30	17.42	13.5	94.12	23.61
iTunes-Amazon	15.61	95.66	26.84	28.32	87.59	42.80	11.57	98.47	20.71	15.46	77.77	25.79	20.69	85.12	33.29
WDC-Computers	29.74	84.24	43.96	32.49	64.76	43.27	29.59	91.20	44.68	23.71	82.45	36.83	92.55	60.41	73.10
WDC-Cameras	30.57	85.40	45.02	33.08	72.24	45.31	26.99	90.16	41.54	36.05	87.77	51.11	80.51	61.97	70.03
WDC-Watches	35.49	85.36	50.14	34.47	75.68	47.37	11.17	83.18	19.70	34.19	85.44	48.84	84.13	68.82	75.71
WDC-Shoes	32.79	62.24	42.95	32.51	78.35	51.64	36.43	75.19	49.08	29.22	65.09	29.22	75.48	50.17	60.28

Table 5: ICL Few Shot performance without any model training.

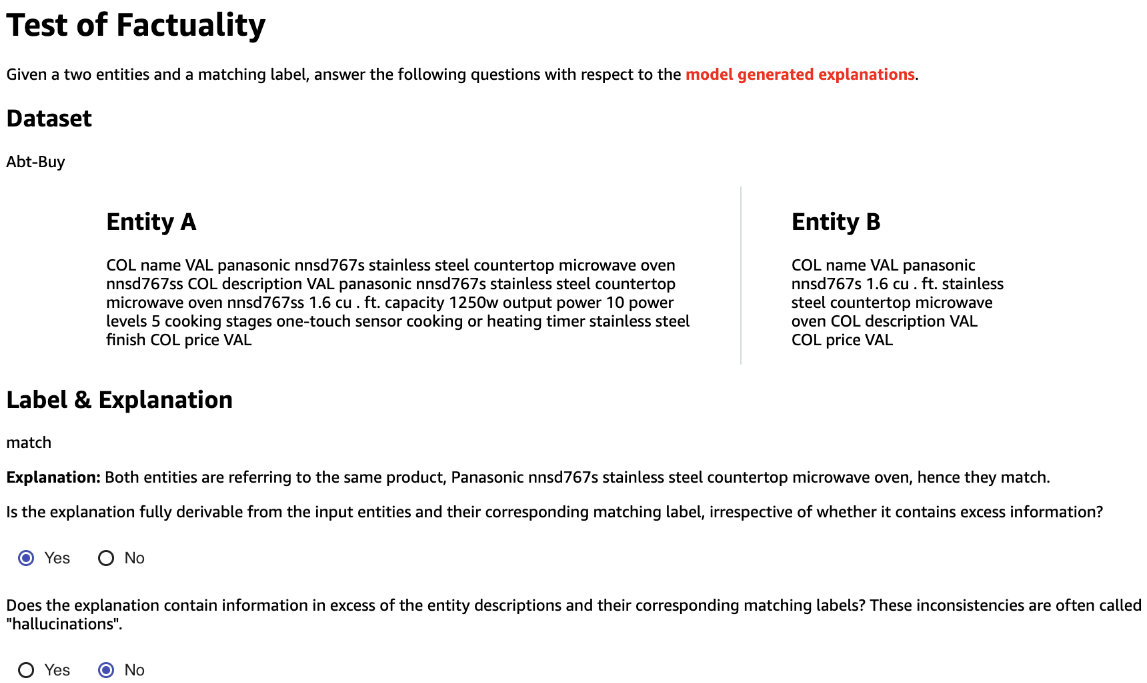

Figure 4: Interface to conduct Test of Factuality annotations on instances taken from the Abt-Buy dataset. Each model-generated (Mistral-7B; Jiang et al. (2023)) explanation is tested for intrinsic and extrinsic errors.