# *KAER*: A Knowledge Augmented Pre-Trained Language Model for Entity Resolution

**Liri Fang**[1*], **Lan Li**[1*], **Yiren Liu**[2], **Vetle I. Torvik**[1], **Bertram Ludäscher**[1]

[1] School of Information Sciences, University of Illinois Urbana-Champaign
[2] Informatics, University of Illinois Urbana-Champaign
lirif2@illinois.edu, lanl2@illinois.edu, yirenl2@illinois.edu, vtorvik@illinois.edu, ludaesch@illinois.edu

## Abstract

Entity resolution has been an essential and well-studied task in data cleaning research for decades. Existing work has discussed the feasibility of utilizing pre-trained language models to perform entity resolution and achieved promising results. However, few works have discussed injecting domain knowledge to improve the performance of pre-trained language models on entity resolution tasks. In this study, we propose **K**nowledge **A**ugmented **E**ntity **R**esolution (*KAER*), a novel framework named for augmenting pre-trained language models with external knowledge for entity resolution. We discuss the results of utilizing different knowledge augmentation and prompting methods to improve entity resolution performance. Our model improves on Ditto, the existing state-of-the-art entity resolution method. In particular, 1) *KAER* performs more robustly and achieves better results on "dirty data", and 2) with more general knowledge injection, *KAER* outperforms the existing baseline models on the textual dataset and dataset from the online product domain. 3) *KAER* achieves competitive results on highly domain-specific datasets, such as citation datasets, requiring the injection of expert knowledge in future work.

## Introduction

Recent studies using *transformer-based Pre-trained Language Models* (PLMs) have shown their strong ability to perform various types of NLP tasks (Min et al. 2021). However, few studies have discussed the application of PLMs in the domain of data cleaning (Li et al. 2020; Narayan et al. 2022; Vos, Döhmen, and Schelter 2022). Entity resolution is a common data cleaning task that aims to identify the entries referring to the same real-world entities within or across databases (Christen 2012).

Most existing techniques on entity resolution assume the same schema for records from different sources (Elmagarmid, Ipeirotis, and Verykios 2007). However, in many situations, raw records are obtained from heterogeneous sources and use different schema (Enríquez et al. 2017; Li et al. 2021b). In addition, source data is often from varied domains (e.g., publications, online products, musicians) and in different formats (e.g., numerical, textual, geolocations). All of these increase the difficulty for practitioners to perform entity resolution tasks without prior knowledge of the domain-specific information about the data. Thus, we hypothesize that enhancing the external knowledge at the schema and entity level can improve entity resolution tasks.

With transformer-based PLMs, recent studies draw increasing attention to entity resolution problems (Li et al. 2020; Trabelsi, Heflin, and Cao 2022). However, current studies show that the performance might not be ideal when simply inputting the serialized entity pairs into PLMs for classification. Ditto (Li et al. 2020) injects domain information: pre-defined entity types (i.e., PRODUCT and NUM), and standardizes the numerical formats to improve the performance before feeding the serialized entity pairs into PLMs.

We push this idea further by injecting more external knowledge at the schema and entity level. Knowledge injection at the schema level aims to infer the fine-grained semantic types (e.g., ALBUM, ARTIST, PUBLISHER) for each column based on data values. For the entity level, entity mentions are identified from WikiData and annotated in the initial text with semantic type information of the linked entities. In addition, different formats used to inject external knowledge into the initial entity pairs may vary the performance of PLMs. Thus, three prompting methods are further explored in this study: space, slash, and constrained tuning of PLMs.

To summarize, starting from state-of-the-art method Ditto (Li et al. 2020), we propose a framework for **K**nowledge **A**ugmented **E**ntity **R**esolution (KAER):

- using **C**olumn **S**emantic **T**ype (CST) inference and **E**ntity **L**inking (EL) in order to inject domain-specific information as additional signals to pre-trained language models.
- leveraging three prompting methods to better augment the acquired knowledge to PLMs.
- analyzing the effectiveness of different combinations of knowledge injection and prompting methods on entity resolution tasks from different domains and data types.

[*] These authors contributed equally.

Accepted to Workshop on Knowledge Augmented Methods for Natural Language Processing, in conjunction with AAAI 2023.

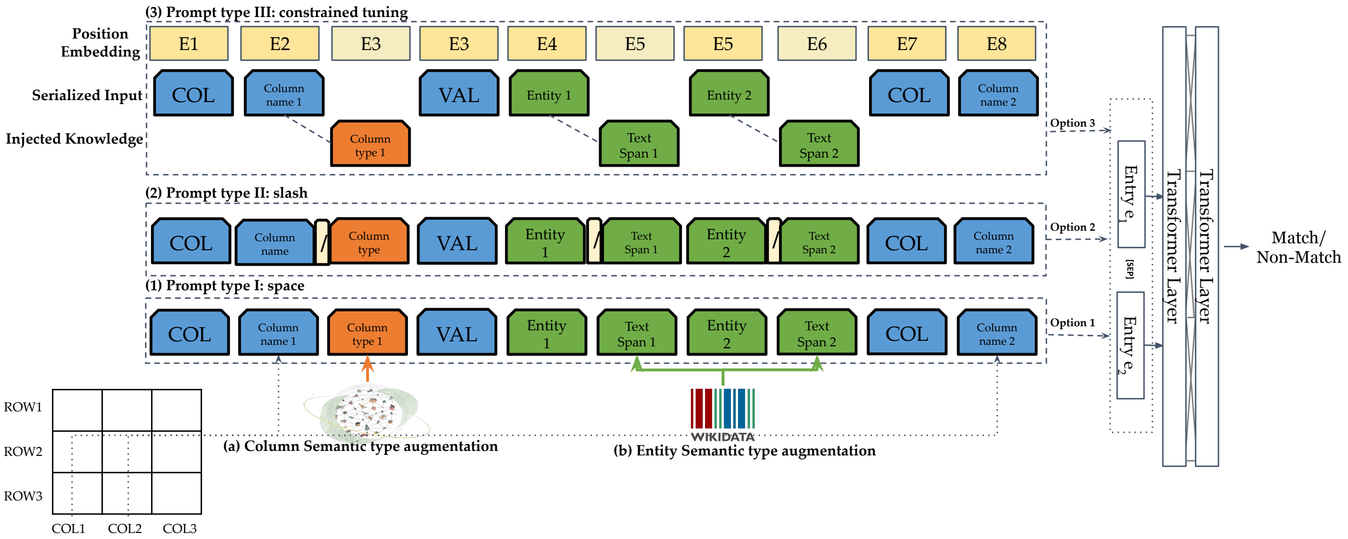

Figure 1: The framework of KAER. First, (a) column semantic type augmentation and (b) entity semantic type augmentation are used to inject domain knowledge from both column and entity levels of the input table. Then three options of prompting types are provided after external knowledge injection: (1) using space; (2) using "/"; (3) using constrained tuning. With the knowledge injection and prompting methods, the serialized sequence is fed into the PLMs.

# Related Work

## Pretrained Model for Entity Resolution

A few recent works apply transformer-based PLMs to entity resolution tasks. Paganelli et al. (2022) discover that simply fine-tuning BERT can benefit matching/non-matching classification tasks and recognize the input sequence as a pair of records. Li et al. (2021a) leverage siamese network structure with PLMs to improve the efficiency of PLMs during the blocking phase. Ditto by (Li et al. 2020) is the state-of-the-art entity matching system based on PLMs, i.e., RoBERTa. In addition, Ditto provides a deeper language understanding for entity resolution by injecting domain knowledge, summarizing the key information, and augmenting with more difficult examples for training data.

## Knowledge Augmentation

Recent studies on semantic column type augmentation and entity linking can inject external domain knowledge into PLMs at the schema and entity levels. Semantic column-type augmentation can inject domain-specific knowledge for columns with/without existing column names. Existing studies (Hulsebos et al. 2019; Suhara et al. 2022) use deep learning approaches to detect semantic data types at the column level. Entity linking (Li et al. 2020) refers to linking entity mentions appearing in natural language text with their corresponding entities in an external knowledge base, e.g., Wikidata. Ayoola et al. (2022) introduces an entity linking method by fine-tuning a PLM over Wikidata, which is used for entity linking in this study.

"prompting" is the method of conditioning the language model Liu et al. (2021). Kojima et al. (2022) propose Zero-shot-CoT, a single zero-shot prompt that highlights that the performance of the language model has been affected by different templates of the prompt. Furthermore, the method to inject the identified external knowledge into PLMs matters. Liu et al. (2020) propose the model K-BERT adding soft-position encoding and visible matrix to the augmented input sequence to reduce knowledge noise injected into the original input and corresponding models.

# Methodology

## Notation of Entity Resolution Task

The formulation of the entity resolution task in this paper is based on the current work by Li et al. (2020). The input of the entity resolution task consists of a set $M \subseteq D_1 \times D_2$, where $D_1$ and $D_2$ are two sets of data entry collections that contain duplicated entries. For each data entry, $e \in (col_i, val_i)_{1 \leq i \leq N}$ where $N$ is the number of columns, and $(e_1, e_2) \in M$. The task discussed in this paper focuses on: for each data entry pair $(e_1, e_2) \in M$, determine whether $e_1$ and $e_2$ refer to the same data entity.

## Model Overview

This study introduces a new framework for entity resolution: KAER (See Figure 1). KAER uses PLMs for entity resolution and contains three modules for knowledge augmentation: a) column semantic type augmentation, b) entity semantic type augmentation, and c) three options of prompting types. The following sections first describe the PLMs for entity resolution and then each module.

**Pre-trained language model for entity resolution.** Following the work by Li et al. (2020), KAER uses RoBERTa as the backbone model. For each data entry pair, $(e_1, e_2)$, the text context of column names and values of $e_1$ and $e_2$ are serialized and concatenated as the input for PLMs. The [CLS] token position is used to classify whether $e_1$ and $e_2$ refer to the same entity. The loss for optimizing the classification objective is:

$$\ell = -log\ p(y|s(e_1, e_2)) \tag{1}$$

where y denotes whether $e_1$ and $e_2$ refer to the same entity, and $s(\cdot, \cdot)$ denotes the serialization and transformation of entity pairs with knowledge injection and prompting methods.

$$s(e_i, e_j) ::= \text{[CLS]}\ serialize(e_i)\ \text{[SEP]}\ serialize(e_j)\ \text{[SEP]} \qquad (2)$$

where $serialize(\cdot)$ serializes each data entry.

$$\begin{aligned} serialize(e_i) ::=&\text{[COL]}\ f(col_1, pt)\ \text{[VAL]}\ g(val_1, pt)\ ... \\ &\text{[COL]}\ f(col_N, pt)\ \text{[VAL]}\ g(val_N, pt) \end{aligned} \qquad (3)$$

where $f(col_i, pt)$ denotes the semantic column type injection with prompting method $pt$, and $g(val_i, pt)$ denotes the entity linking injection with prompting method $pt$. The following sections describe column semantic type injection, entity linking, and three prompting methods.

**Column semantic type augmentation.** Column type prediction aims to predict the semantic type of each column by considering the intra-column context, so that fine-grained semantic information can be augmented to the framework. For example, "name" as a column name can provide more semantic information if injected with the semantic type "song". Here, KAER uses Sherlock (Hulsebos et al. 2019) for column semantic type prediction. In detail, *Sherlock* is a multi-input deep neural network pre-trained on 686,765 data columns retrieved from the VizNet corpus by matching 78 semantic data types from DBpedia to column headers (Hulsebos et al. 2019).

**Entity semantic type augmentation.** Entity semantic type augmentation leverages the entity linking method to identify all entity mentions from a given knowledge base (KB) within the target text sequence. More specifically, 1) to identify the text span of each entity mention $m \in M$, and 2) map each mention to the entity set from an external KB $\mathcal{M} : M \rightarrow E$. Thus, additional knowledge from the KB is used to augment the input text records. In this study, WikiData is used as the external KB since it covers a wide range of domains and is suitable for entity resolution over different datasets. KAER uses the state-of-the-art method proposed by Ayoola et al. (2022), which uses a RoBERTa model jointly pre-trained over entity typing and entity description modeling objectives to inject coarse entity types.

### Prompting methods

**Template-based prompting.** After acquiring domain knowledge, we experiment using text-based templates to combine the knowledge with the initial text input. Two different characters are used to concatenate the original entity mention/column name, i.e. slash ("/") and space. The slash symbol is chosen because it contains a semantic meaning equivalent to 'or' in general web text.

**Prompting with constrained tuning** Inspired by hard-soft prompt hybrid tuning (Han et al. 2021), prompting with constrained tuning is used to generate the token embedding by controlling the visible area of the augmented knowledge only to its corresponding entity mentions or column names. This type of domain knowledge injection modifies the initial input sequence in two dimensions, including the token position and token semantics. More specifically, the token from the initial input sequence has a different absolute position when the template-based prompting is concatenated. In addition, the injected knowledge for a specific entity mention is shared with the surrounding tokens in the same sequence. These two modifications might introduce erroneous knowledge and overload to the framework.

To limit the erroneous knowledge and overload, we leverage constrained tuning, i.e., soft-position encoding and visible matrix, as an additional input of our backbone model, inspired by KBert (Liu et al. 2020). Here, the knowledge-injected sequence is constructed into a tree structure, where the injected knowledge subsequences are considered branches. In this way, as shown in Figure 1, the soft-position encoding assigns the initial tokens the same position encoding in the knowledge-injected sequence as those in the initial input sequence, and assigns the injected knowledge tokens the position in the branches. The visible matrix $V \in \mathcal{R}^{N \times N}$ is a binary matrix, where the injected knowledge token is only visible to itself and the corresponding entity mention or column name tokens, and set as 1, denoted as follows:

$$V_{ij} = \begin{cases} 1, & \text{if } token_i \text{ and } token_j \text{ co-occur.} \\ 0, & \text{otherwise.} \end{cases} \qquad (4)$$

where the co-occurrence of $token_i$ and $token_j$ indicates that the tokens co-occur in the initial input sequences or co-occur as head and tail in domain knowledge injection. Here, the head means the entity mention/column name, and the tail indicates the injected knowledge text.

# Experiments

## Dataset

KAER is evaluated on the Magellan datasets (Das et al. 2022) across various domains. The overview of the datasets used in this study is shown in Table 2. The Magellan datasets contain three types of datasets: dirty dataset, structured dataset, and textual dataset. The dirty dataset is generated from the structured dataset by randomly removing attribute values and appending the initial values to another randomly selected attribute (Li et al. 2020).

## Results

The experimental results are presented in Table 1.

*Knowledge augmentation performs better on smaller datasets.* According to the experimental results, incorporating external knowledge can improve the performance on entity resolution tasks. In particular, the proposed knowledge augmentation methods outperform the two baseline models in a data-scarce context. For example, **KAER (RoBERTa+EL+/)** outperforms both baseline models Roberta and Ditto by 9.09% and 21.44% ↑ on dataset Dirty/iTunes-Amazon, but not with statistic significance. Meanwhile, it statistically significantly outperforms both baseline models 24.45% and 39.15% ↑ on the dataset Structured/iTunes-Amazon. The two baseline models perform worse on smaller datasets because the language model might not be able to learn enough information to distinguish between different entities when fine tuning, given fewer training instances correctly. Prompting with

| | Dirty | | Structured | | | | Textual |
|---|---|---|---|---|---|---|---|
| **Models** | **DBLP**<br>**-**<br>**GoogleScholar** | **iTunes**<br>**-**<br>**Amazon** | **iTunes**<br>**-**<br>**Amazon** | **Amazon**<br>**-**<br>**Google** | **DBLP**<br>**-**<br>**ACM** | **DBLP**<br>**-**<br>**GoogleScholar** | **Abt**<br>**-**<br>**Buy** |
| RoBERTa | 95.81 | 72.73 | 61.76 | 73.12 | 98.77 | 95.32 | 88.41 |
| Ditto | 95.54 | 60.38 | 47.06 | 74.44 | 98.55 | 95.85 | 88.89 |
| KAER ( Roberta + CST ) | 95.47 | 62.96 | 80.70**++ | 68.70 | 98.77 | 95.05 | 92.27**++ |
| KAER ( Roberta + CST + / ) | 95.74 | 54.90* | **89.66**++** | 74.85 | **98.99** | 95.14 | **92.61**++** |
| KAER ( Roberta + CST + PCT) | 95.20 | 72.00 | 57.14 | **76.25** | 98.09* | 95.62 | 78.24**++ |
| KAER ( Roberta + EL ) | 95.12 | 50.00* | 61.54 | 73.77 | 98.54 | 95.72 | 89.11 |
| KAER ( Roberta + EL + / ) | 95.75 | **81.82** | 86.21**++ | 74.02 | 98.43 | **96.11*** | 88.19 |
| KAER ( Roberta + EL + PCT) | 94.80**+ | 77.78+ | 67.69 | 75.63 | 97.04**+ | 95.29 | 73.80**++ |
| KAER ( Roberta + CST + EL ) | **95.87** | 47.06* | 59.26 | 72.28 | 98.43 | 95.65 | 89.66 |
| KAER ( Roberta + CST + EL + / ) | 95.63 | 63.33* | 57.14+ | 75.15 | 98.34 | 95.67 | 89.81 |
| KAER ( Roberta + CST + EL + PCT) | 94.24**++ | 71.11+ | 61.29 | 71.37* | 96.26**++ | 94.93+ | 67.20**++ |

Table 1: Experimental results using different knowledge injection methods, measured by F1 score. The two baseline models: RoBERTa (without domain knowledge injection); and Ditto (RoBERTa and Ditto's general domain knowledge injection methods). Other models represent KAER with various knowledge injection and prompting methods combined with RoBERTa. "+CST" indicates **C**olumn **S**emantic **T**ype injection."+EL" indicates knowledge injection with **E**ntity **L**inking. "+/" represents prompting with a slash. "+PCT" represents **P**rompting with **C**onstrained **T**uning, i.e., soft positions and visible matrix. The paired t-test is processed on between KAER injections and two baseline models, i.e., RoBERTa(*) and Ditto(+). ** and ++ represent 99% confidence level, and * and + represent 95% confidence level.

| Data Type | Dataset | Domain | Size | # Positive | # Attr. |
|---|---|---|---|---|---|
| Structured | Amazon-Google | software | 11,460 | 1,167 | 3 |
| | iTunes-Amazon | music | 539 | 132 | 8 |
| | DBLP-ACM | citation | 12,363 | 2,220 | 4 |
| | DBLP-GoogleScholar | citation | 28,707 | 5,347 | 4 |
| Dirty | iTunes-Amazon | music | 539 | 132 | 8 |
| | DBLP-ACM | citation | 12,363 | 2,220 | 4 |
| | DBLP-GoogleScholar | citation | 28,707 | 5,347 | 4 |
| Textual | Abt-Buy | product | 9,575 | 1,028 | 3 |

Table 2: Dataset Summary

PCT achieves better performance in datasets with smaller text lengths. For example, **KAER (RoBERTa+CST+PCT)** reaches the best F1 score (76.25%) without statistic significance on dataset Structured/Amazon-Google, which contains the smallest length of serialized data entry. Moreover, **KAER (RoBERTa+CST+PCT)** on Structured/Amazon-Google makes a significant difference compared to both baseline models, at 99% confidence level.

*Knowledge augmentation does not perform well on datasets from certain domains.* The datasets used in our experiments span several different domains. Using our proposed knowledge augmentation methods (CST and EL), records from certain domains, such as music and online product, benefit from more accurate retrieved knowledge. For example, **KAER (RoBERTa+CST+/)** achieves the best F1-score with statistic significance, i.e., 92.61%, on the textual dataset Abt-Buy from the online product domain. **KAER (RoBERTa+EL+/)** achieves the best F1-score (81.82%) without statistical significance on the musical dataset Dirty/iTunes-Amazon. One potential reason for this notable improvement is that the entities mentioned in the datasets from these domains are more general. However, publication datasets from domains like citation require more domain-specific or even expert knowledge rather than general commonsense. Our knowledge injection methods are primarily designed for retrieving general knowledge. Thus, for datasets, like Structured/DBLP-ACM, **KAER (RoBERTa+CST)** performs as well as Roberta and significantly makes a difference with Ditto at 95% confidence level, and **KAER (RoBERTa+CST+/)** achieves the best F1-score 98.99% without statistic significance. But all the other KAER models perform worse. For future improvement, additional external expert knowledge from the citation domain should be incorporated for knowledge injection.

*Knowledge augmentation is affected by data quality.* KAER, with certain knowledge augmentation methods, outperforms the baseline models on dirty datasets. For instance, **KAER (RoBERTa+CST+EL)** slightly exceeds RoBERTa, and Ditto on Dirty/DBLP-GoogleScholar. However, the F1 scores of all the other injections on the Dirty/DBLP-GoogleScholar cannot compete with the results on the baseline models (RoBERTa and Ditto). One reason might be that the misleading predictions based on the dirty input by CST result in semantic noise and propagate to the PLMs. On the other hand, knowledge augmentation with PCT can advance the robustness of the model dealing with low-quality injections. For instance, **KAER (RoBERTa+CST+PCT)** obtains better F1 score than **KAER (RoBERTa+CST)** (72.00% vs. 62.96%). Similar improvement applies to **KAER (RoBERTa+EL+PCT)** vs. **KAER (RoBERTa+EL)** and **KAER (RoBERTa+CST+EL+PCT)** vs. **KAER (RoBERTa+CST+EL)**.

## Conclusion

In this study, we introduced a novel framework, KAER, for entity resolution problems by augmenting PLMS with external knowledge using prompting techniques. The framework is evaluated over a set of datasets from varied domains and data types (i.e., dirty, structured, and textual) and improves performance in certain applications over the baseline models. Further analysis revealed the effectiveness of different domain knowledge injection and prompting methods on various datasets from different domains and data types. The results show a promising direction of injecting knowledge to improve entity resolution performance using PLMs. Future studies can improve upon our method by introducing more domain-specific knowledge.

## Acknowledgement

The experiments are conducted on the NCSA Delta supercomputing system, supported by the National Science Foundation under Grant No. OAC-2005572. This work is supported by the National Institute on Aging of the NIH (Award Number P01AG039347). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## References

Ayoola, T.; Tyagi, S.; Fisher, J.; Christodoulopoulos, C.; and Pierleoni, A. 2022. ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking. In *Proceedings of the 2022 NAACL: Human Language Technologies: Industry Track, NAACL 2022*, 209–220.

Christen, P. 2012. The Data Matching Process. In *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Data-Centric Systems and Applications, 23–35. Berlin, Heidelberg: Springer. ISBN 978-3-642-31164-2.

Das, S.; Doan, A.; G. C., P. S.; Gokhale, C.; Konda, P.; Govind, Y.; and Paulsen, D. 2022. The Magellan Data Repository. https://sites.google.com/site/anhaidgroup/projects/data.

Elmagarmid, A. K.; Ipeirotis, P. G.; and Verykios, V. S. 2007. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1): 1–16. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Enríquez, J.; Domínguez-Mayo, F.; Escalona, M.; Ross, M.; and Staples, G. 2017. Entity reconciliation in big data sources: A systematic mapping study. *Expert Systems with Applications*, 80: 14–27.

Han, X.; Zhao, W.; Ding, N.; Liu, Z.; and Sun, M. 2021. PTR: Prompt Tuning with Rules for Text Classification. *CoRR*, abs/2105.11259.

Hulsebos, M.; Hu, K. Z.; Bakker, M. A.; Zgraggen, E.; Satyanarayan, A.; Kraska, T.; Demiralp, Ç.; and Hidalgo, C. A. 2019. Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on KDD 2019*, 1500–1508. ACM.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. *CoRR*, abs/2205.11916.

Li, B.; Miao, Y.; Wang, Y.; Sun, Y.; and Wang, W. 2021a. Improving the Efficiency and Effectiveness for BERT-based Entity Resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15): 13226–13233. Number: 15.

Li, X.; Talburt, J. R.; Li, T.; and Liu, X. 2021b. When Entity Resolution Meets Deep Learning, Is Similarity Measure Necessary? In *Advances in Artificial Intelligence and Applied Cognitive Computing*, 127–140. Springer.

Li, Y.; Li, J.; Suhara, Y.; Doan, A.; and Tan, W.-C. 2020. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14(1): 50–60.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *CoRR*, abs/2107.13586.

Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; and Wang, P. 2020. K-BERT: Enabling Language Representation with Knowledge Graph. In *Proceedings of the AAAI*, volume 34, 2901–2908.

Min, B.; Ross, H.; Sulem, E.; Veyseh, A. P. B.; Nguyen, T. H.; Sainz, O.; Agirre, E.; Heintz, I.; and Roth, D. 2021. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. *CoRR*, abs/2111.01243.

Narayan, A.; Chami, I.; Orr, L. J.; and Ré, C. 2022. Can Foundation Models Wrangle Your Data? *CoRR*, abs/2205.09911.

Paganelli, M.; Buono, F. D.; Baraldi, A.; and Guerra, F. 2022. Analyzing how BERT performs entity matching. *Proceedings of the VLDB Endowment*, 15(8): 1726–1738.

Suhara, Y.; Li, J.; Li, Y.; Zhang, D.; Demiralp, c.; Chen, C.; and Tan, W.-C. 2022. Annotating Columns with Pre-Trained Language Models. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, 1493–1503. Association for Computing Machinery.

Trabelsi, M.; Heflin, J.; and Cao, J. 2022. DAME: Domain Adaptation for Matching Entities. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 1016–1024. Virtual Event AZ USA: ACM. ISBN 978-1-4503-9132-0.

Vos, D.; Döhmen, T.; and Schelter, S. 2022. Towards Parameter-Efficient Automation of Data Wrangling Tasks with Prefix-Tuning. In *NeurIPS 2022 First Table Representation Workshop*.

## APPENDIX

**A. Hyperparameter Settings.** Baseline and KAER models share the same hyperparameter setting, in which the batch size equals 64, the max length equals 512, the learning rate is 3e-5, and the number of training epochs is 20.

**B. Baseline models.** This paper includes two baseline models, i.e., RoBERTa without any knowledge injection and Ditto with its default knowledge injection methods.

| Dataset | Injection |
|---|---|
| dirty/DBLP-Google | Ditto + General |
| dirty/iTunes-Amazon | Ditto + Product |
| structured/iTunes-Amazon | Ditto + Product |
| structured/Amazon-Google | Ditto + Product |
| structured/DBLP-ACM | Ditto + General |
| structured/DBLP-Google | Ditto + General |
| textual/Abt-Buy | Ditto + Product |

Table 3: Datasets and Corresponding Ditto default knowledge injection methods

In detail, Table 3 shows the datasets and corresponding injection methods utilized by Ditto. ”+ General” indicates that Ditto injects seven entity types into the corresponding dataset. The seven types include ’PERSON’, ’ORG’, ’LOC’, ’PRODUCT’, ’DATE’, ’QUANTITY’, and ’TIME’. ”+ Product” indicates that Ditto annotates as ’PORODUCT’, any entity mentions in the following types, i.e., ’NORP’, ’GPE’, ’LOC’, ’PERSON’, and ’PRODUCT’.

**C. Two-sided Paired T-test.** The two-sided paired T-test is implemented to test whether the predictions of different models are significantly different. We first compare whether the predicted label is equal to the ground-truth label. The value is 1 if the prediction is equal to the ground truth and 0 otherwise. And then, the paired T-test is conducted.