Product Entity Matching via Tabular Data

Ali Naeim abadi
University of Alberta
Edmonton, Alberta, Canada
naeimaba@ualberta.ca

Mir Tafseer Nayeem
University of Alberta
Edmonton, Alberta, Canada
mnayeem@ualberta.ca

Davood Rafiei
University of Alberta
Edmonton, Alberta, Canada
drafiei@ualberta.ca

ABSTRACT

Product Entity Matching (PEM)–a subfield of record linkage that focuses on linking records that refer to the same product–is a challenging task for many entity matching models. For example, recent transformer models report a near-perfect performance score on many datasets while their performance is the lowest on PEM datasets. In this paper, we study PEM under the common setting where the information is spread over text and tables. We show that adding tables can enrich the existing PEM datasets and those tables can act as a bridge between the entities being matched. We also propose TATEM, an effective solution that leverages Pre-trained Language Models (PLMs) with a novel serialization technique to encode tabular product data and an attribute ranking module to make our model more data-efficient. Our experiments on both current benchmark datasets and our proposed datasets show significant improvements compared to state-of-the-art methods, including Large Language Models (LLMs) in zero-shot and few-shot settings.

CCS CONCEPTS

• **Information systems** → **Entity resolution**; **Deduplication**.

KEYWORDS

Entity matching, entity resolution, pretrained language models

ACM Reference Format:
Ali Naeim abadi, Mir Tafseer Nayeem, and Davood Rafiei. 2023. Product Entity Matching via Tabular Data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615172

1 INTRODUCTION

Entity Matching (EM) is the process of identifying and linking entity descriptions from different sources [11]. It involves recognizing records that refer to the same real-world entity, such as a person, organization, or product despite differences across databases [26]. Linking entities from both structured and unstructured sources is crucial in various domains such as e-commerce, HR hiring, advertising, and market research [18]. For example, a price comparison website may perform product matching for data from different vendors before finding a site that sells the same product for the lowest price. Or, a new vendor may want to create a comprehensive product catalog by collecting data from different sources and merging the entries that refer to the same product to avoid redundancy. However, product formatting across different sites may not be consistent, and multiple features of a product may be packed into a product title or description without any separation or indication of what those features are (see some examples in Table 1).

Amazon Product Title	Google Product Title
"mcafee total protection 2007 3 users"	"mtp07emb3rua mcafee total protection 2007 complete package 3 users cd mini-box"
"britannica deluxe"	"britannica deluxe 2008"
"nero 7 ultra edition enhanced"	"70009 nero ultra edition enhanced v.7 complete package 1 user cd win"

Table 1: A few hard negative examples [20]. Despite their highly similar titles, product pairs are not the same.

The problem of entity matching has been extensively studied using various datasets [15, 16, 27] and recent techniques such as injecting domain knowledge [10, 17], improving the serialization [1], and utilizing LLMs [21]. Here we focus on Product Entity Matching (PEM) for two important reasons: Firstly, the PEM datasets, such as Amazon-Google [20] and Walmart-Amazon [20], continue to pose a significant challenge, compared to many other datasets on which the existing EM models have already achieved a near-perfect performance [21]. Secondly, numerous real-world e-commerce applications can benefit from more effective PEM solutions. We have identified three major shortcomings of these popular PEM benchmarks. **Firstly**, some product titles are highly similar but are labeled as non-matching pairs (referred to as hard negative examples). Even for a human annotator, it is difficult to distinguish those non-matching examples based on the information given in the dataset. As a result, State-Of-The-Art (SOTA) models struggle to disambiguate them. Table 1 shows three hard negative examples from the Amazon-Google dataset [20]. **Secondly**, certain parts of a title are more useful for reaching a matching decision (e.g., model, year introduced, functionalities, etc.), but those pieces of information may be located in different places for different products, and the model may not have direct access to the encoded attributes. For instance, in the first example in Table 1, the manufacturer *"mcafee"* is written at the beginning of the Amazon product title and in the middle of the Google one. An EM model should recognize if it is an important piece of information and the attribute it describes. **Thirdly**, in all these datasets, a fixed number of attributes are given for all samples (e.g., title, manufacturer, and price for Amazon-Google dataset [20]). Different products can have different sets of attributes and limiting the attributes to a fixed set is likely to miss key product features.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615172

We enrich PEM benchmarks from [20] to offer a product detail table as a source of additional knowledge for every Amazon product, serving as a bridge to connect two entities of interest and potentially improving the accuracy of matching decisions. The datasets include a varying number of attributes for each product, and the introduction of the detail table allows all characteristic features to be captured. The supplementary data is anticipated to reveal distinguishing features that may not be present in the product title, which can help the model to disambiguate hard negative examples. Consider the example shown in Figure 1, where a model number is given for the Google product but this model number is not mentioned in the title of the Amazon product. The presence of the model number in the detail table provides this missing link between the two product descriptions.

We also present **TA**ble & **T**ext for **E**ntity **M**atching (**TATEM**), an entity matching approach based on PLMs. TATEM reaches new SOTA results for PEM benchmarks by incorporating the complementary tabular data. We further introduce an **A**ttribute **R**anking **M**odule (**ARM**) to rank the attributes of one entity based on their relevance to other entity of interest. Our evaluation shows that ARM is capable of finding the most effective attributes for EM. Our contributions are summarized as follows: **(1)** We enrich popular and challenging PEM benchmarks [20] with complementary product tables. **(2)** We propose a new serialization technique to encode semi-structured tables in our PEM datasets for PLM-based models. **(3)** We develop TATEM, a model which employs both tabular and textual information for EM, reaching a new SOTA for challenging PEM benchmarks. **(4)** We design ARM to select important product-specific attributes and to make the model data-efficient.

2 RELATED WORK

Before the advent of Deep Learning (DL), early EM models were using rule-based models [9, 29] or traditional ML to learn matching functions [3]. In the era of DL, many EM models used RNN architectures and attention mechanisms: MPM [12], DeepMatcher [20], Hi-EM [33], Seq2SeqMatcher [22], and DeepER [8] models. Recent high-performance models benefit from a fine-tuned PLM to tackle the problem. DITTO [17], a prominent EM model, concatenates a pair of records to form a sequence, and fine-tunes a PLM to solve a sequence-pair classification problem. Brunner and Stockinger [5] introduce a similar transformer-based solution. ROBEM [1], inspired by DITTO, achieves promising results with an improved serialization technique, a loss function designed for the imbalanced dataset, and a higher degree of non-linearity in the classification head. Peeters and Bizer [24] deploy Joint BERT [6] for EM. SupCon [25] extends this approach using supervised contrastive learning [14], achieving SOTA results for Amazon-Google dataset. More recently, Narayan et al. [21] utilize LLMs such as **GPT3** [4] to push the SOTA results for EM benchmarks.

There are some studies on using additional resources to improve the performance. KAER [10] resorts to Wikidata as a knowledge graph to inject external knowledge at both schema and entity levels. However, this method is less likeley to be effective for hard negative examples, as they typically belong to the same category at the schema level, and new and less popular products are less likely to be found in the knowledge graph. DITTO [17] highlights the importance of extra domain knowledge for EM by adding Named Entity Recognition (NER) tags from spaCy [30] and rewriting text spans with developer-specified rules (e.g., replacing 5 % and 5.00 % with 5.0%). However, the rewriting rules do not add any extra domain knowledge. Although DITTO [17] uses its domain knowledge to identify the important pieces of information in the title, it cannot identify the specific attribute a text span describes, and the model has no direct access to the text span as a separate attribute. KAER [10] argues that external knowledge can help with the heterogenity of data sources, but the reported results are only on the EM benchmarks that share the same schema. These recent studies have not focused on or addressed the aforementioned issues due to the lack of a relevant dataset. Therefore, in this paper, we introduce Amazon-Google-Tab and Walmart-Amazon-Tab datasets (§3) and a model called TATEM (§4) to fill these gaps.

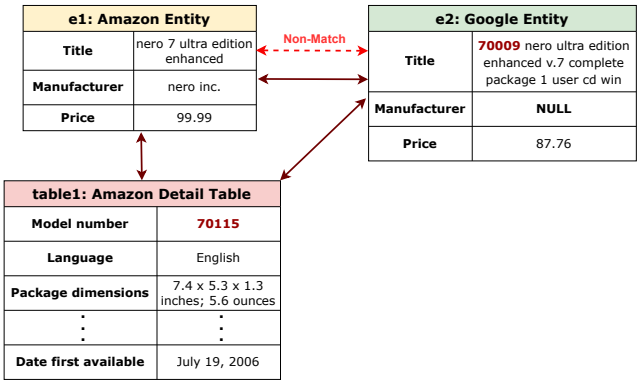

Figure 1: A hard negative example disambiguated using an Amazon product detail table, showing that relying on the information given in titles alone is hard to vote against a match because of the large number of overlapping tokens. Our model TATEM disambiguates this by establishing a relationship between $e2$ and $table1$ (*if exists*). Here, the `Model Number` field helps TATEM to reach a `Non-Match` decision.

3 DATASET

Many structured EM datasets have a fixed schema for their records, with a predetermined number of attributes for each sample. For instance, the original Amazon-Google dataset introduced by Köpcke et al. [16] had four attributes: title, manufacturer, price, and description. Based on the original Amazon-Google [16] and Walmart-Amazon datasets [13], Mudgal et al. [20] released structured Amazon-Google and Walmart-Amazon datasets with only three and five attributes, respectively. Currently, high-performance EM models utilize the later version of the datasets. Our proposed enriched Amazon-Google-Tab and Walmart-Amazon-Tab datasets are based on the original datasets. To improve the effectiveness of EM models in disambiguating hard negative examples (see Figure 1) and to provide them with new challenges, we have added product-specific tables with varying numbers of attributes and many distinct schemas. Those tables are obtained by retrieving Amazon product pages using `ASIN` [2] and extracting the relevant tabular data.

Our enriched Amazon-Google-Tab and Walmart-Amazon-Tab datasets capture all the key features of Amazon products, such as

	Amazon-Google [20]	Walmart-Amazon [20]
#Train samples (N./P.)	6175/699	5568/579
#Test samples (N./P.)	2059/234	1856/193
#Attrs (*fixed*)	3	5
	Amazon-Google-Tab (*ours*)	**Walmart-Amazon-Tab** (*ours*)
#Tables (Amazon)	909	16264
Table coverage	66%	73%
#Unique attrs	84	695
Avg. #attrs	10.2	19.97
Max #attrs	28	81

Table 2: Statistics of the datasets.

model number, language, compatible OS, genre, and more. However, detail tables are product specific, and the schema and attributes vary between products. Table 2 provides some statistics of the structured Amazon-Google [20], structured Walmart-Amazon [20], our enriched datasets. The Amazon-Google-Tab dataset provides product detail tables for 909 Amazon products through 84 unique attributes. Interestingly, our Walmart-Amazon-Tab dataset includes more intricate product tables for a total of 16,264 Amazon items using 695 different unique attributes. Consequently, the EM task becomes challenging in **(1)** serialization of tables using PLMs (§4.1) and **(2)** data-efficient solutions for long tables (§4.2).

4 TATEM MODEL

4.1 TATEM Serialization

TATEM employs a serialization technique for semi-structured, product-specific data. In contrast, DITTO [17] and ROBEM [1] use serialization techniques for structured datasets that have a fixed number of attributes. Although ROBEM and DITTO serialization techniques have the potential to be applied to semi-structured data, they are not as effective as TATEM serialization as it is shown in Table 3. Here TATEM serialization is explained for Amazon-Google-Tab although the same procedure is applied to Walmart-Amazon-Tab. For every example, there exist three fixed attributes: title, manufacturer, and price; and k additional attributes from the product detail table, with k varied for each product:
$e = (title,\ val_{title}), (manufac,\ val_{manufac}), (price,\ val_{price}),$
$\{(attr_i,\ val_i)\}_{1 \le i \le k}$.

To serialize an entity e, for the first three attributes, only the attribute value is considered because they always appear in the same position, but for the other k attributes, the attribute name is concatenated with the attribute value because the attributes are different for every product. Similar to ROBEM, a special token appears between the attributes:
$serialize(e) ::= val_{title}$ [ATTR] $val_{manufac}$ [ATTR] val_{price} [ATTR] $(attr_i,\ val_i) \ldots$ [ATTR] $(attr_k,\ val_k))$.

For example, the Amazon entity given in Figure 1 is serialized as: nero 7 ultra edition enhanced [ATTR] nero inc. [ATTR] 99.99 [ATTR] model number 70115 [ATTR] ... [ATTR] date first available July 19, 2006.

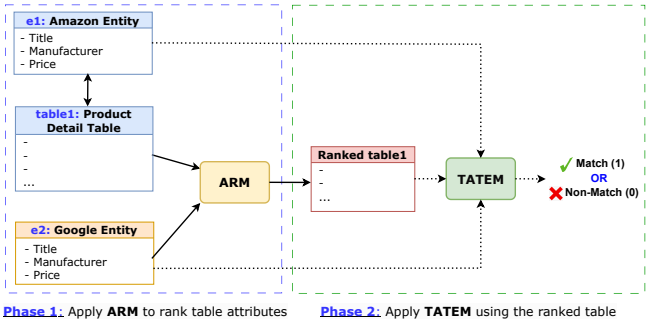

Figure 2: Our TATEM model coupled with ARM for PEM.

4.2 Attribute Ranking Module (ARM)

In both of our datasets, each product entity is associated with a large number of attributes, making it challenging to determine which attributes are the most indicative of a true match or non-match. The goal of ARM is to generate the top n attribute-value pairs for a given product entity (e.g., from Amazon) in response to a pair of entities (e.g., Amazon-Google) for EM. This is important for two main reasons. Firstly, transformer-based PLMs [7, 19] have a limitation on the maximum length of input sequences they can handle [31]. The most common solution to this problem is to trim the input sequences to a certain length (*e.g.,* 512). However, trimming a long input sequence is tricky for EM in general because important information for matching decisions may be located towards the bottom of the tables. Secondly, employing TATEM equipped with ARM can improve the overall efficiency and effectiveness of the EM task. For instance, reducing the number of input tokens can save computational resources, quicken the inference time, and save financial resources in particular when using pay-as-you-go services such as GPT PLMs [23].

Here ARM is described for Amazon-Google-Tab although the same procedure is applied to Walmart-Amazon-Tab dataset to select the most influential attributes. Walmart*-Amazon-Tab Inspired by unsupervised text ranking [32], ARM calculates the relevance of Amazon detail table attributes, $attr_i$, with respect to a Google product context, and it returns the top n attributes based on these estimates of relevance.

$$P\,(Relevence = 1 \mid attr_i, cntx) \triangleq \phi(\eta_{attr}(attr_i), \eta_{cntx}(cntx))$$

where ϕ is a comparison function and η_{cntx} and η_{attr} give encodings of a context from one source and a table attribute from another source, respectively. Our design choice for both encoders is Sentence-BERT (SBERT) [28], and we utilize cosine similarity as the comparison function and title as the Google product context. Important Amazon attributes for EM should contain information about features that are mentioned in a Google record, but not in an Amazon record. Consequently, an effective ARM should make matching records closer and non-matching records farther by adding just a few high-ranking attributes without a big drop in performance.

Figure 2 illustrates the operations of TATEM on an Amazon entity (t1) with 3 attributes (`title`, `manufacturer`, `price`) and a product detail table, compared to a Google product (t2). In phase 1, ARM outputs a ranked list of attribute-value pairs. Interestingly, the ranked table attributes are different for each pair of records. In phase

2, TATEM is applied to the Amazon-Google pair enriched with the ranked Amazon attributes and it outputs a matching decision (0, 1).

5 EXPERIMENTAL EVALUATION

We utilized RoBERTa$_{base}$ [19] as our PLM because it has been found effective for the EM tasks [17, 18]. For training the models, we followed the parameter configuration of DITTO [17].

5.1 Results

Here, we aim to evaluate the importance of auxiliary tabular data about product key features, the effectiveness of TATEM serialization methodology, and the effect of ARM on the model's data efficiency. As a base for comparison, Table 3 shows the performance, in terms of F1-score, of DeepMatcher+ (DM+), DITTO, ROBEM, KAER, SupCon, and GPT3 on structured PEM datasets [20]. To demonstrate how competitive models (DITTO, ROBEM, and SupCon) perform if they have access to our enriched datasets, we design two sets of experiments: **(1)** We convert our semi-structured Amazon-Google-Tab and Walmart-Amazon-Tab datasets into a structured format with a column allocated to each unique attribute. The results are denoted as Amazon-Google-Tab and Walmart-Amazon-Tab (*ours*) [structured] in Table 3. **(2)** We modify the implementation of DITTO and ROBEM to be compatible with our semi-structured data using their respective serializations described in Section 4.1. For the modified models, denoted as DITTO-m and ROBEM-m, we take the attribute name and attribute value directly from our Amazon-Google-Tab and Walmart-Amazon-Tab datasets to serialize each entry.

As reported in Table 3, once DITTO has access to the structured format of Amazon-Goole-Tab dataset, it outperforms the current SOTA model (SupCon) and presents a better performance than ROBEM, underscoring the importance of having additional product knowledge for EM. On the contrary, when DITTO-m and ROBEM-m are utilized for Amazon-Goole-Tab dataset, ROBEM-m outperforms DITTO-m. This disparity in performance is rooted in the serialization techniques employed by each model, highlighting the need for a serialization technique that is tailored to the structure of data. Adding complementary information to Walmart-Amazon in a structured format doesn't help and even leads to a decrease in F1 score. This unintentional behavior is rooted in the difference between our two enriched datasets. In fact, Walmart-Amazon-Tab includes far more unique features (695 compared to 84 unique features) as noted in Table 2, and this leads to wider, more sparse tables in a structured format, which confuses the PLM-based EM. In contrast, employing Walmart-Amazon-Tab in a semi-structured format for DITTO-m and ROBEM-m outperforms the last SOTA results. It emphasizes the advantages of directly serializing semi-structured data, particularly for lengthy, complex product tables.

Our proposed model, TATEM, reaches new SOTA results (F1 score of 82.2 and 90.56 for Amazon-Google-Tab and Walmart-Amazon-Tab, respectively) as it benefits from a serialization technique that is specially designed for the product-specific tabular structure of our enriched datasets. Based on our findings, the best serialization for the three *fixed* attributes (i.e., title, manufacturer, price) is to exclude the attribute names. On the other hand, for k varying attributes from the product detail table, including both the attribute name and value is the best strategy as it provides the EM model with information about the attribute type. The KAER model [10], despite having access to additional entity information from WikiData, fails to outperform ROBEM and reaches an F1 score of 76.25 on Amazon-Google dataset [20], demonstrating that accessing extra information does not essentially guarantee an increase in performance.

Models	F1 Score	
	Amazon-Google [20]	**Walmart-Amazon** [20]
DM+ (2018)	70.7	73.6
DITTO (2020)	75.58	86.76
KAER (2023)	76.25	-
ROBEM (2022)	79.06	86.68
SupCon (2022)	79.28	-
GPT3(k=0) (2022)	54.3	60.6
GPT3(k=10) (2022)	63.5	87.0
	Amazon-Google-Tab (*ours*) [structured]	**Walmart-Amazon-Tab** (*ours*) [structured]
DITTO	80.56	86.85
ROBEM	78.50	85.74
SupCon	78.58	-
	Amazon-Google-Tab (*ours*)	**Walmart-Amazon-Tab** (*ours*)
DITTO-m	79.35	86.42
ROBEM-m	80.92	88.31
TATEM (*ours*)		
+ w/ *all tuples*	**82.2**	**90.56**
+ w/ ARM (n=1)	80.12	88.52
+ w/ ARM (n=3)	81.28	89.24
+ w/ ARM (n=5)	81.83	89.77

Table 3: Performance of TATEM compared to different baselines. All reported results for TATEM (*ours*) are statistically significant in paired t-test by taking DITTO (2020) as a reference with the confidence of 95% (p-value < 0.05).

In our datasets, a typical entity can have up to 81 attributes. However, ARM can significantly reduce the number of tokens fed to a PLM. An effective ARM should identify the most important attributes for EM to make matching pairs closer and non-matching pairs further apart with just a few attributes. To evaluate the effects, ARM is applied to our datasets, and from the ranked list, only the top n attributes are selected. Just adding the top one attribute can beat the current SOTA results (see Table 3), and the top five attributes are responsible for most of the performance gain.

6 CONCLUSION AND FUTURE WORK

We have introduced two new datasets and solution that uses PLMs with a novel serialization technique to encode semi-structured tables. The experiments conducted on both existing benchmark datasets and the proposed datasets show significant improvements. Additionally, we have designed an unsupervised attribute ranking module that enhances the model's data-efficiency and cost-effectiveness. For future work, we will study the robustness of our model against distribution shift and input perturbation.

ACKNOWLEDGMENTS

This research has been supported by the NSERC and a grant from Huawei. Also, Ali Naeim abadi was supported by the AI GSS, while Mir Tafseer Nayeem was supported by the Huawei PhD Fellowship.

REFERENCES

[1] Mehdi Akbarian Rastaghi, Ehsan Kamalloo, and Davood Rafiei. 2022. Probing the Robustness of Pre-Trained Language Models for Entity Matching. In *Proceedings of the 31st ACM International Conference on Information &; Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 3786–3790. https://doi.org/10.1145/3511808.3557673

[2] Rebecca Allen. 1996. Amazon Standard Identification Number (ASIN). https://en.wikipedia.org/wiki/Amazon_Standard_Identification_Number Accessed: February 20, 2023.

[3] Mikhail Bilenko and Raymond J. Mooney. 2003. Adaptive Duplicate Detection Using Learnable String Similarity Measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, D.C.) *(KDD '03)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/956750.956759

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[5] Ursin Brunner and Kurt Stockinger. 2020. Entity matching with transformer architectures - a step forward in data integration. In *Proceedings of EDBT 2020*. OpenProceedings. https://doi.org/10.21256/zhaw-19637 23rd International Conference on Extending Database Technology, Copenhagen, 30 March - 2 April 2020.

[6] Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for Joint Intent Classification and Slot Filling. *CoRR* abs/1902.10909 (2019). arXiv:1902.10909 http://arxiv.org/abs/1902.10909

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. *Proc. VLDB Endow.* 11, 11 (oct 2018), 1454–1467. https://doi.org/10.14778/3236187.3236198

[9] Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. 2009. Reasoning about Record Matching Rules. *Proc. VLDB Endow.* 2, 1 (aug 2009), 407–418. https://doi.org/10.14778/1687627.1687674

[10] Liri Fang, Lan Li, Yiren Liu, Vetle I. Torvik, and Bertram Ludäscher. 2023. KAER: A Knowledge Augmented Pre-Trained Language Model for Entity Resolution. https://doi.org/10.48550/ARXIV.2301.04770

[11] Ivan P. Fellegi and Alan B. Sunter. 1969. A Theory for Record Linkage. *J. Amer. Statist. Assoc.* 64, 328 (1969), 1183–1210. https://doi.org/10.1080/01621459.1969.10501049

[12] Cheng Fu, Xianpei Han, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. End-to-End Multi-Perspective Matching for Entity Resolution. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4961–4967. https://doi.org/10.24963/ijcai.2019/689

[13] Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F. Naughton, Narasimhan Rampalli, Jude Shavlik, and Xiaojin Zhu. 2014. Corleone: Hands-off Crowdsourcing for Entity Matching. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (Snowbird, Utah, USA) *(SIGMOD '14)*. Association for Computing Machinery, New York, NY, USA, 601–612. https://doi.org/10.1145/2588555.2588576

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 18661–18673. https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf

[15] 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 over Data Science Stacks. *Proc. VLDB Endow.* 9, 13 (sep 2016), 1581–1584. https://doi.org/10.14778/3007263.3007314

[16] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of Entity Resolution Approaches on Real-World Match Problems. *Proc. VLDB Endow.* 3, 1–2 (sep 2010), 484–493. https://doi.org/10.14778/1920841.1920904

[17] 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 (oct 2020), 50–60. https://doi.org/10.14778/3421424.3421431

[18] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, Jin Wang, Wataru Hirota, and Wang-Chiew Tan. 2021. Deep Entity Matching: Challenges and Opportunities. *J. Data and Information Quality* 13, 1, Article 1 (jan 2021), 17 pages. https://doi.org/10.1145/3431816

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019). https://arxiv.org/abs/1907.11692

[20] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 19–34. https://doi.org/10.1145/3183713.3196926

[21] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? *Proc. VLDB Endow.* 16, 4 (dec 2022), 738–746. https://doi.org/10.14778/3574245.3574258

[22] 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 *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 629–638. https://doi.org/10.1145/3357384.3358018

[23] OpenAI. 2023. Pricing: Simple and flexible. Only pay for what you use. https://openai.com/pricing Accessed: May 20, 2023.

[24] Ralph Peeters and Christian Bizer. 2021. Dual-Objective Fine-Tuning of BERT for Entity Matching. *Proc. VLDB Endow.* 14, 10 (oct 2021), 1913–1921. https://doi.org/10.14778/3467861.3467878

[25] Ralph Peeters and Christian Bizer. 2022. Supervised Contrastive Learning for Product Matching. In *Companion Proceedings of the Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 248–251. https://doi.org/10.1145/3487553.3524254

[26] Ralph Peeters, Reng Chiz Der, and Christian Bizer. 2023. WDC Products: A Multi-Dimensional Entity Matching Benchmark. https://doi.org/10.48550/ARXIV.2301.09521

[27] Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In *Companion Proceedings of The 2019 World Wide Web Conference* (San Francisco, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 381–386. https://doi.org/10.1145/3308560.3316609

[28] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[29] Rohit Singh, Vamsi Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. 2017. Generating Concise Entity Matching Rules. In *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, Illinois, USA) *(SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1635–1638. https://doi.org/10.1145/3035918.3058739

[30] spaCy. 2017. Entity Recognizer. https://spacy.io/api/entityrecognizer Accessed: February 20, 2023.

[31] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification?. In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings* (Kunming, China). Springer-Verlag, Berlin, Heidelberg, 194–206. https://doi.org/10.1007/978-3-030-32381-3_16

[32] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (Virtual Event, Israel) *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 1154–1156. https://doi.org/10.1145/3437963.3441667

[33] Chen Zhao and Yeye He. 2019. Auto-EM: End-to-End Fuzzy Entity-Matching Using Pre-Trained Deep Models and Transfer Learning. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2413–2424. https://doi.org/10.1145/3308558.3313578