Probing the Robustness of Pre-trained Language Models for Entity Matching

Mehdi Akbarian Rastaghi
makbaria@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Ehsan Kamalloo
kamalloo@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

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

ABSTRACT

The paradigm of fine-tuning Pre-trained Language Models (PLMs) has been successful in Entity Matching (EM). Despite their remarkable performance, PLMs exhibit tendency to learn spurious correlations from training data. In this work, we aim at investigating whether PLM-based entity matching models can be trusted in real-world applications where data distribution is different from that of training. To this end, we design an evaluation benchmark to assess the robustness of EM models to facilitate their deployment in the real-world settings. Our assessments reveal that data imbalance in the training data is a key problem for robustness. We also find that data augmentation alone is not sufficient to make a model robust. As a remedy, we prescribe simple modifications that can improve the robustness of PLM-based EM models. Our experiments show that while yielding superior results for in-domain generalization, our proposed model significantly improves the model robustness, compared to state-of-the-art EM models.

CCS CONCEPTS

• **Information systems** → **Entity resolution**; **Data management systems**; • **Information integration** → *Entity resolution*; • **Entity Matching**;

KEYWORDS

Entity Matching, Entity Linking, Named Entity Disambiguation

ACM Reference Format:
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 and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557673

1 INTRODUCTION

In many real-world applications where data is integrated from multiple sources, matching mentions that refer to the same real-world entities is crucial. Entity Matching (EM) aims at automatically detecting such mentions or records that are likely derived from different schemas. With the recent success of transfer-learning from large pre-trained language models (PLMs) [4, 5, 18, 19] in many NLP tasks, EM models such as Ditto [12] have followed suit to leverage PLMs for EM. The paradigm of fine-tuning PLMs have achieved remarkable performance on several well-known EM benchmarks. Despite their success and popularity, PLMs are no panacea [2]. Numerous studies [14, 16] have found their tendency to learn the underlying spurious patterns in data. This essentially means that PLMs tend to acquire a superficial understanding of the task at hand and are more likely to fail under different circumstances such as distribution shift. Such shortcomings lead to inexplicable errors that inhibit their deployment in real-world applications.

EM can also be vulnerable to these problems. In this work, we investigate whether PLM-based models can be deployed for entity matching "in the wild," where the distribution of the test data often differs from that of the training data in a real-word setting. This is especially pivotal in EM as stored data from different sources are hardly homogeneous [1]. To this end, we study the robustness of these models to shed light on their pitfalls under various distribution shifts. Our focus in this paper is on "structured" data where the content of the records and the ordering of the fields vary from one domain to next. For this purpose, we first fine-tune a PLM-based model on a dataset, then evaluate it on several crafted test benchmarks in a zero-shot fashion. The benchmarks are created to evaluate EM models for two types of robustness that are prevalent in the real-world settings: domain shift and structural shifts. For domain shift, we conduct out-of-domain evaluations and for structural shifts, we devise perturbation strategies to modify the structure of tuples without altering the matching outcome.

In addition to the distribution shift and the change in structure between domains, the EM data is highly imbalanced. Table 1 shows this phenomenon for several well-known datasets. Data Augmentation (DA) is a common technique to circumvent this problem. In essence, DA makes a model invariant to changes that are less relevant to the task. Although shown effective, we find that DA alone is not sufficient for building a model that is robust to distribution shift. As a remedy, we propose a simple loss function to strengthen the models' ability to put more emphasis on the minority label. We also provide two other recommendations to make PLM-based EM models more robust. Our experiments corroborate that our proposed model is more robust than the state-of-the-art EM models[1].

Our main contributions can be summarized as follows: (1) We investigate the impact of common strategies in EM models from the robustness perspective. (2) We design an evaluation framework to test the robustness of EM models under various distribution shifts. (3) Based on our findings, we propose simple modifications

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557673

[1] Our code and models are released at https://github.com/makbn/robem

to Ditto, the state-of-the-art EM model, to build a robust model that surpasses Ditto on in-domain tests as well as out-of-domain tests.

2 RELATED WORK

EM prior to PLMs. EM has long been studied in the database community under different names such as data deduplication, record linkage, etc. [6]. Early EM models were rule-based [7, 20] or aimed at learning matching functions [3] using traditional machine learning. In the deep learning era, most EM models leverage deep neural networks. DeepMatcher [15], a prominent EM model, uses RNNs and the attention mechanism. Auto-EM [23] applies transfer learning, analogous to the popular trend in PLMs, to learn from a general-purpose model, trained on massive knowledge bases.

PLM-based EM. Recent EM models have shifted to the paradigm of fine-tuning PLMs to tackle the problem. Ditto [12], the state-of-the-art EM model, concatenates a pair of records to form a sequence and fine-tunes a PLM using a sequence classification objective. JointBERT [17] uses a dual-objective training method that combines binary matching and multi-class classification that forces the model to predict entity identifiers as well as matching decisions.

Domain Adaptation in EM. The goal of domain adaptation is to generalize to new domains through learning from unlabeled examples. In EM, recent works such as DAME [21] focus on transferring knowledge from multiple sources to a target domain using a Mixture-of-Experts model.

3 PRELIMINARIES

The goal of EM is to successfully match mentions, derived from presumably different data sources, that refer to the same real-world entity. In this work, our focus is on the structured data where mentions are stored as tuples [15]. In particular, suppose $A = \{A_0, A_1, ..., A_n\}$ and $B = \{B_0, B_1, ..., B_m\}$ denote two data sources where A_i and B_j represent records in each data source. Each record consists of several attributes — i.e., $A_i = (a_1, a_2, a_3, ...a_k)$. $A_i = B_j$ if and only if both A_i and B_j depict the same real-world entity. EM is characterized as a binary classification task to predict whether two records are identical or not. In a PLM-based EM model, two records are concatenated together, separated by a special token. The model is fine-tuned using a sequence classification objective.

4 ROBEM: A ROBUST ENTITY MATCHER

We build our model, namely RobEM, atop Ditto, the state-of-the-art EM model, that leverages PLMs for identifying identical pairs. Our main focus is to make a PLM-based model for EM more robust. To this end, we modify Ditto as discussed next.

Data Imbalance. In EM datasets, there is an extreme imbalance between examples labelled as negative and positive [1], rendering the negative examples as the majority class. However, Ditto and other prominent EM models use the standard cross-entropy loss, which does not take into account the data imbalance during training. We circumvent this problem by a common method, known as weighted cross-entropy, that is basically the standard cross-entropy, albeit with weights for each class [24]. The weights are typically proportional to the frequency of each class in the training data [11].

Dispensing with Attribute Names. In practice, structured records are collected from anywhere in the web. These records may lack attribute names due to a variety of reasons — e.g., parsing difficulties, or missing information. However, using attribute names is a common practice in EM [12, 15]. As a result, EM models are likely to become impaired when faced with circumstances where attribute headers are not given, curbing their robustness capabilities. To overcome this issue, we dispense with the assumption that attribute names are present in the data to account for such cases. This essentially denotes that our model purely relies on the content of each record for matching.

Classifier Head. In Ditto, the task-specific classification head that projects the output of a PLM to logits is a linear layer. However, due to the complexity of the task, adding a non-linearity in this layer can be helpful. Thus, we employ tanh with dropout, following the classifier head in RoBERTa [13], in the task-specific classifier head:

$$y = W_2 \cdot \tanh(W_1 \cdot E'_{[\texttt{CLS}]} + b_1) + b_2. \quad (1)$$

Moreover, we introduce a simple baseline, dubbed UnsupEM, that takes an off-the-shelf PLM to determine whether two tuples are equivalent based on their similarity in the embedding space. In particular, we first feed each tuple into a PLM and take the output of the `[CLS]` token as the representation of the tuple. We then compute cosine similarity between the two representations. A similarity that is above a certain threshold indicates equivalency.

5 EM ROBUSTNESS BENCHMARK

To assess the robustness capabilities of EM models, we devise a series of probing tests, simulating the distribution shifts that may arise in real-world scenarios. The first test is domain shift — or out-of-domain — where the domain of test data differs from that of training data. Our main goal here is to understand whether PLM-based EM models have actually mastered the task rather than relying on spurious patterns in the data. To this end, a model, trained on one dataset, is tested against the other datasets.

The next series of tests attempt to replicate schema discrepancies between the tuples from two different data sources. For this purpose, we check the invariance against structural shifts via applying the following perturbation operations on the original test data to produce 5 new test sets.

	Title	Manufacturer	Price	Label	
E_1	*microsoft visual studio team edition 2005*	*Microsoft*	5479	1	Original Sample
E_2	*microsoft visual studio 2005 professional*	*Microsoft*	757.75		
SFF	*Microsoft* *microsoft visual studio team edition 2005*		5479	1	Perturbed Samples
	microsoft visual studio 2005 professional	*Microsoft*	757.75		
DRP	*microsoft visual studio team edition 2005*	*Microsoft*		1	
	microsoft visual studio 2005 professional	*Microsoft*	757.75		
TYP	*microsoft visual studio team edition 2005*	*Microsoft*	5479	1	
	microsoft visual studio 2005 professional	*Microsoft*	$757.75		
MIS	*microsoft visual studio team edition 2005*	*Microsoft*	5479	1	
	microsoft visual studio 2005 professional	*Microsoft*	NULL		
EXT	*microsoft visual studio team edition 2005*	*Microsoft* IN-STORE-ONLY	5479	1	
	microsoft visual studio 2005 professional	*Microsoft*	757.75		

Figure 1: Perturbation operations for the schema discrepancy robustness benchmark. Key columns are shown in italic.

(1) **Robustness to column order (SFF)**: The ordering of columns does not affect the matching result between a pair of tuples. To ensure this condition, we shuffle columns of a tuple for each example in the test data.
(2) **Robustness to absence of non-key columns (DRP)**: *Non-key columns* are columns that do not contribute to the matching result between a pair of tuples — e.g., price in Figure 1. Matching should remain invariant to inclusion or exclusion of non-key columns. This condition is enforced by randomly dropping one or more non-key columns.
(3) **Robustness to missing values (MIS)**: The existence of missing values is prevalent when dealing with noisy data sources such as web tables. To imitate this, we randomly replace one or more non-key columns with `NULL`.
(4) **Robustness to extraneous columns (EXT)**: The presence of irrelevant non-key columns does not affect the matching result between a pair of tuples. We enforce this case by randomly adding columns from other datasets for each test example.
(5) **Robustness to different data types (TYP)**: Data entries can be expressed in a handful of ways without changing their semantics. This is especially the case for numerical values. For instance, "1k" is equivalent to "1,000" and "1e3". We curate several hand-crafted rules to randomly convert numbers to different formats to enforce this condition.

6 EXPERIMENTS

Setup. We implemented our models using the Huggingface Transformers library [22]. We select $\text{RoBERTa}_{\text{base}}$ [13], a well-known PLM, that is shown to be effective in EM [12]. We follow the hyperparameter configuration of Ditto for training our models. In particular, we set the maximum sequence length to 256, the batch size to 64, and the number of epochs to 40. The learning rate is set to $3e$-5 with a linear decay. All experiments were conducted on a single Nvidia V100 32GB GPU.

Datasets. We use 8 datasets, introduced in DeepMatcher [15]. The datasets are derived from an entity resolution benchmark [10] as well as the Magellan data repository [9]. The datasets are collected from a wide range of domains including products, publications, and businesses. For all datasets, each example consists of candidate pairs from two structured tables within the same schema. Table 1 presents the size of each dataset.

Table 1: Datasets size. P. refers to the tuple pairs marked as a match. And N. is the Number of non-matched instances.

Dataset	Train Set (N./P.)	Test Set (N./P.)
iTunes-Amazon	243 / 78	82 / 27
Amazon-Google	6175 / 699	2059 / 234
BeerAdvo-RateBeer	228 / 40	77/14
DBLP-ACM	6085 / 1332	2029 / 444
DBLP-Scholar	14016 / 3207	4672 / 1070
Fodors-Zagats	501 / 66	167 / 22
Walmart-Amazon	5568 / 576	1856 / 193
Abt-Buy	5127 / 616	1710 / 206

6.1 In-Domain Generalization

We first examine the generalization of EM models to unseen test data that are from the same domain as the training data. For in-domain experiments, we compare our results with two prominent neural entity matching models: Deepmatcher+ [8], an RNN-based model, and Ditto [12], a PLM-based model with the same number of parameters.

As presented in Table 2, RobEM consistently surpasses Ditto, on all datasets, except for two datasets, iTunes-Amazon (-1.65%), and DBLP-Scholar (-0.22%). Interestingly, the highest performance gain (+8.99%) is achieved on Abt-Buy. UnsupEM understandably trails all the baselines and RobEM on all datasets because it does not exploit any supervised signals from the data.

6.2 Out-of-Domain Generalization

We compare RobEM with UnsupEM and Ditto in out-of-domain experiments. UnsupEM offers a lower bound for supervised models. The vis-a-vis results — i.e., the difference between RobEM and the baselines — that consist of 56 runs are reported in Figure 2. Only on 15 cases in total, RobEM lags behind UnsupEM. Furthermore, when trained on BeerAdvo-RateBeer, RobEM struggles most with out-of-domain generalization. On the other hand, the models that are trained on iTunes-Amazon and BeerAdvo-RateBeer significantly outperform Ditto on all 7 datasets. In total, RobEM trails Ditto on 17 cases. Overall, the improvements of RobEM over UnsupEM and Ditto are statistically significant in 34 and 31 cases, respectively. Finally, we find that UnsupEM is a strong baseline in out-of-domain tests, leading both RobEM and Ditto on 13 tests.

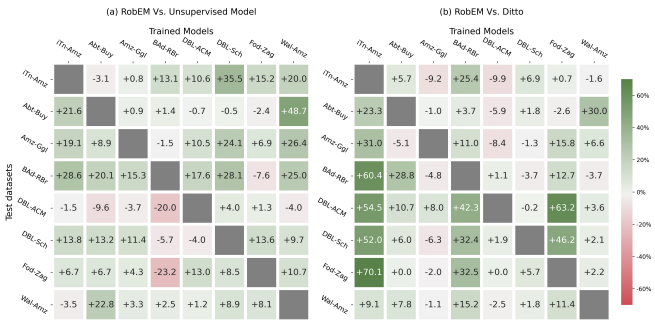

Figure 2: Difference between F1 scores of RobEM and two baselines, Ditto (right) and UnsupEM (left) in zero-shot out-of-domain experiments.

6.3 Data Augmentation

Data augmentation (DA) is a long known technique to counter data imbalance and to boost the generalization capabilities of models. In this section, we aim at evaluating RobEM and Ditto when trained on augmented data. We follow the DA method, presented in Ditto. In particular, Ditto DA involves generating augmented data online during training. Each example is augmented via a series of consecutive operations that randomly perturb attributes and tokens. Following Ditto, we generate one augmented sample for each training example. The in-domain results for DA are presented in the

Table 2: F1 scores for in-domain experiments. DeepMatcher+ and Ditto results are taken verbatim from [12].

Dataset	DeepMatcher+	UnsupEM	Ditto (DK)	RobEM		Ditto (All) + DA	RobEM + DA	
iTunes-Amazon	91.2	54.54	97.80	96.15	-1.65	97.06	**98.18**	+1.12
Abt-Buy	62.8	21.84	81.69	90.68	+8.99	89.33	**90.9**	+1.57
Amazon-Google	70.7	31.04	74.67	76.64	+1.97	75.58	**79.06**	+3.48
BeerAdvo-RateBeer	78.8	64.70	90.46	93.33	+2.87	94.37	**96.55**	+2.18
DBLP-ACM	98.45	92.77	99.10	**99.21**	+0.11	98.99	99.10	+0.11
DBLP-Scholar	94.7	69.77	95.80	95.62	-0.22	95.60	**95.86**	+0.26
Fodors-Zagats	**100.0**	86.95	**100.00**	**100.00**	0.00	**100.00**	**100.00**	0.00
Walmart-Amazon	73.6	29.85	83.73	86.68	+2.95	**86.76**	84.61	-2.15

Figure 3: Difference between F1 scores of RobEM+DA and Ditto+DA in zero-shot out-of-domain experiments.

right columns of Table 2. DA does not improve the in-domain performance of Ditto on three datasets, iTunes-Amazon, DBLP-ACM, and DBLP-Scholar. However, DA brings in-domain improvements for RobEM on all datasets but two cases. RobEM+DA consistently outperforms Ditto+DA on all datasets, except on Walmart-Amazon.

In the out-of-domain experiments, the results, presented in Figure 3, are consistent with our findings in Figure 2(b). Specifically, RobEM+DA, trained on iTunes-Amazon, BeerAdvo-RateBeer, and Walmart-Amazon, outperforms Ditto on all 7 datasets by a significant margin, except for one case. DA helps RobEM, trained on Amazon-Google, to achieve better results than Ditto on 5 datasets. However, when comparing RobEM with DA and without, we find using DA improves 3 out of 56 tests. This essentially highlights that DA is not necessarily useful for robustness under domain shift.

6.4 Schema Discrepancy Generalization

We conduct our robustness test only on the best performing models from the out-of-domain experiment. More precisely, we adopt the models, trained on iTunes-Amazon using DA throughout this section. We surmise that our findings can be extended to other datasets as well. To understand the impact of DA, we employ DA techniques, akin to the ones we adopted to generate the robustness benchmark in Section 5. The idea is to imitate the cases of potential distribution shift to teach the model during training. For brevity, we only use the methods for SFF in this experiment to generate augmented datasets offline:

Table 3: F1 scores on iTunes-Amazon dataset on our EM robustness benchmark. The perturbation operations are defined in Section 5.

	DA	in-domain	SFF	DRP	MIS	EXT	TYP	Δ_{avg}
RobEM	SF	96.15	91.6	93.35	96.15	95.74	97.13	-1.35
	SW	98.11	93.83	96.29	97.69	99.25	97.95	-1.10
	Ditto	98.18	96.8	98.11	98.11	97.74	98.11	-0.40
Ditto	SF	96.42	95.47	91.94	96.42	96.42	93.69	-1.63
	SW	94.54	90.29	91.78	92.85	92.96	94.54	-2.05
	Ditto	93.10	92.81	89.47	90.56	93.76	93.10	-1.16

- **Tuple Swap (SW)**, inspired by Ditto, refers to swapping the left tuple with the right one for each training example.
- **Attribute Shuffle (SF)**, inspired by Ditto, shuffles the order of attributes for each training example.

Table 3 shows the results, averaged over 20 runs, for RobEM and Ditto on our robustness benchmark. We report the average performance drop (Δ_{avg}), compared to in-domain results[2]. We observe that Ditto DA method is more robust, compared to SW and SF as it yields the lowest performance drop for both RobEM and Ditto. Also, RobEM is consistently more robust than Ditto across all three DA methods. Interestingly, SFF and DRP are the two most challenging perturbation tests for RobEM and Ditto, respectively.

7 CONCLUSION

In this work, we investigated the robustness capabilities of EM models under domain shifts and structrual shifts. We prescribe simple guidelines to build robust models that are suitable for deployment in the wild. Our proposed model outperforms the state-of-the-art PLM-based EM model under distribution shift. We hope that our findings spurs development of more robust EM models. Also, our robustness benchmark can be a basis for a thorough assessment of future EM models. We plan to explore unstructured data, complex data augmentation techniques and other forms of distribution shift as future directions.

ACKNOWLEDGMENTS

This research is supported by the Natural Sciences and Engineering Research Council and by a grant from Huawei.

[2]We attempted to replicate the Ditto results using the official codebase. Our results for Ditto+DA is 93.1%, whereas in the original paper, 97.1% is reported.

REFERENCES

[1] Nils Barlaug and Jon Atle Gulla. 2021. Neural Networks for Entity Matching: A Survey. *ACM Transactions on Knowledge Discovery from Data* (2021). https://doi.org/10.1145/3442200

[2] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. https://doi.org/10.1145/3442188.3445922

[3] Mikhail Bilenko and Raymond J. Mooney. 2003. Adaptive Duplicate Detection Using Learnable String Similarity Measures. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/956750.956759

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *Advances in Neural Information Processing Systems* 2020-December (2020). https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[5] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* (2019). https://doi.org/10.18653/v1/N19-1423

[6] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. 2007. Duplicate Record Detection: A Survey. *IEEE Trans. on Knowl. and Data Eng.* (2007). https://doi.org/10.1109/TKDE.1990.10000

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

[8] Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2020. Low-resource deep entity resolution with transfer and active learning. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference* (2020). https://doi.org/10.18653/v1/p19-1586

[9] Pradap Venkatramanan Konda. 2018. *Magellan: Toward building entity matching management systems*. http://www.vldb.org/pvldb/vol9/p1197-pkonda.pdf

[10] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment* (2010). https://doi.org/10.14778/1920841.1920904

[11] Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice Loss for Data-imbalanced NLP Tasks. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.45

[12] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, Jin Wang, Wataru Hirota, and Wang Chiew Tan. 2021. Deep Entity Matching: Challenges and Opportunities. *Journal of Data and Information Quality* (2021). https://doi.org/10.1145/3431816

[13] 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://doi.org/10.48550/arXiv.1907.11692

[14] Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/P19-1334

[15] 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. *Proceedings of the ACM SIGMOD International Conference on Management of Data*. https://doi.org/10.1145/3183713.3196926

[16] Timothy Niven and Hung Yu Kao. 2020. Probing neural network comprehension of natural language arguments. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. https://doi.org/10.18653/v1/p19-1459

[17] Ralph Peeters and Christian Bizer. 2021. Dual-Objective Fine-Tuning of BERT for Entity Matching. *Proc. VLDB Endow.* (2021). https://doi.org/10.14778/3467861.3467878

[18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. *OpenAI blog* (2018). https://www.gwern.net/docs/www/s3-us-west-2.amazonaws.com/d73fdc5ffa8627bce44dcda2fc012da638ffb158.pdf

[19] Ilya Sutskever Alec Radford, Jeffrey Wu, David Luan Rewon Child, and Dario Amodei. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* (2019). https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[20] 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. *Proceedings of the 2017 ACM International Conference on Management of Data*. https://doi.org/10.1145/3035918.3058739

[21] Mohamed Trabelsi, Jeff Heflin, and Jin Cao. 2022. DAME: Domain Adaptation for Matching Entities. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. https://doi.org/10.1145/3488560.3498486

[22] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[23] Chen Zhao and Yeye He. 2019. Auto-EM: End-to-End Fuzzy Entity-Matching Using Pre-Trained Deep Models and Transfer Learning. *The World Wide Web Conference*. https://doi.org/10.1145/3308558.3313578

[24] Ziyun Zhou, Hong Huang, and Binhao Fang. 2021. Application of Weighted Cross-Entropy Loss Function in Intrusion Detection. *Journal of Computer and Communications* (2021). https://doi.org/10.4236/jcc.2021.911001