How Transformers Are Revolutionizing Entity Matching

Matteo Paganelli[1], Donato Tiano[2], Francesco Del Buono[2], Andrea Baraldi[2], Riccardo Benassi[2], Giacomo Guiduzzi[2] and Francesco Guerra[2,*]

[1]Hasso Plattner Institute, Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany
[2]University of Modena and Reggio Emilia, Via P. Vivarelli 10, Modena, Italy

Abstract

State-of-the-art Entity Matching (EM) approaches rely on transformer architectures to capture hidden matching patterns in the data. Although their adoption has resulted in a breakthrough in EM performance, users have limited insight into the motivations behind their decisions. In this paper, we perform an extensive experimental evaluation to understand the internal mechanisms that allow the transformer architectures to obtain such outstanding results. The main findings resulting from this evaluation are: (1) off-the-shelf transformer-based EM models outperform previous (deep-learning-based) EM approaches; (2) different pre-training tasks result in different effectiveness performance, which is only partially motivated by a different learning of record representations, and (3) the fine-tuning process based on a binary classifier limits the generalization of the models to out-of-distribution data and prevents from learning entity-level representations.

Keywords

Entity Matching, Data integration, Transformers, Interpretability

1. Introduction

Data integration aims to combine heterogeneous data sources into a single, unified, duplicate-free data representation. This improves information organization and accessibility, facilitating more efficient decision-making processes and improving overall data quality. One of the main steps of the data integration pipeline is Entity Matching (EM) which aims to recognize records that refer to the same real-world entity.

Nowadays this task is mainly approached through supervised methods where deep learning models are trained with pairs of records labeled as *match* (or 1), if the two records refer to the same entity, or as *non-match* (or 0) in the opposite case [1]. Architecturally these models consist of two fundamental components: 1) an encoder whose goal is to generate meaningful record pair representations, and 2) a binary classifier that classifies the encoder output as *match* or

SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy
*Corresponding author.
matteo.paganelli@hpi.de (M. Paganelli); donato.tiano@unimore.it (D. Tiano); francesco.delbuono@unimore.it (F. D. Buono); andrea.baraldi96@unimore.it (A. Baraldi); riccardo.benassi@unimore.it (R. Benassi); giacomo.guiduzzi@unimore.it (G. Guiduzzi); francesco.guerra@unimore.it (F. Guerra)
0000-0001-8119-895X (M. Paganelli); 0000-0003-0605-4184 (D. Tiano); 0000-0003-0024-2563 (F. D. Buono); 0000-0002-1015-5490 (A. Baraldi); 0009-0007-4819-259X (R. Benassi); 0000-0003-0819-405X (G. Guiduzzi); 0000-0001-6864-568x (F. Guerra)

non-match. While the classifier typically coincides with a linear layer or a multi-layer perceptron (MLP), most of the complexity of the model resides in the encoder. Current state-of-the-art approaches, like Ditto [2] and R-SupCon [3], implement this component via the transformer architecture [4], or derived models (such as BERT[5], SBERT[6] and RoBERTa[7]), which are able to capture hidden matching patterns in the data after a fine-tuning process[2, 8, 9].

The adoption of transformer architectures has resulted in a breakthrough in the effectiveness of the EM approaches. However, they are black-box architectures and it is not easy to understand which are the internal mechanisms that allow them to obtain such outstanding results. Providing an answer to this question is crucial to increase their trustworthiness and promote their application in real-world scenarios [10].

This paper is an extended abstract of [11, 12], where we addressed this problem. More specifically, we analyzed how transformer-based architectures perform the EM task according to three perspectives[1]. They concern (1) how off-the-shelf transformer-based EM models perform compared to EM state-of-the-art approaches (Section 3); (2) the impact of the pre-training technique on the ability of the transformer to learn the EM task (Section 4), and (3) which is their performance in recognizing entities (Section 5) and how much they can generalize to out-of-distribution data (Section 6).

The three main findings that we obtained by answering the previous questions are:

1. *Off-the-shelf transformer-based EM models outperform previous deep-learning-based EM models (like DeepMatcher[13]) and perform well even on dirty data, where values are misplaced across attributes*;
2. *Different pre-training tasks result in different effectiveness performances, which is only partially motivated by a different learning of record representations.* Only R-SupCon can differentiate the knowledge encoded in the embeddings between matching and non-matching records;
3. *Models that are fine-tuned for EM via a binary classifier do not fully recognize cliques of entity descriptions and have limited generalization capacity to out-of-distribution data.*

2. The Experimental Analysis

This section describes the experimental setup adopted to answer the three research questions mentioned above.

Datasets. We performed the experiments against the datasets provided by the Magellan library[2] which is the reference benchmark for the evaluation of EM tasks. The datasets consist of pairs of entity descriptions sharing a common structure. Table 1 summarizes some statistical measures describing the datasets: the total number of record pairs (fourth column), the percentage of pairs associated with a match label (fifth column), and the number of attributes (last column). Each dataset is already split into train, validation, and test sets with a ratio of 3:1:1.

Models. The evaluation considers four EM models based on the transformer architecture ranging from simple baselines to more advanced and fully-fledged state-of-the-art approaches.

[1]Further analyzes are available in the original papers which are not reported here for reasons of limited space.
[2]https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md

Table 1
The datasets used for the evaluation.

Acronym	Type	Dataset	Size	% Match	# Attributes
S-FZ	Strucured	Fodors-Zagats	946	11.63	6
S-DG		DBLP-GoogleScholar	28,707	18.63	4
S-DA		DBLP-ACM	12,363	17.96	4
S-AG		Amazon-Google	11,460	10.18	3
S-WA		Walmart-Amazon	10,242	9.39	5
S-BR		BeerAdvo-RateBeer	450	15.11	4
S-IA		iTunes-Amazon	539	24.49	8
T-AB	Textual	Abt-Buy	9,575	10.74	3
D-IA	Dirty	iTunes-Amazon	539	24.49	8
D-DA		DBLP-ACM	12,363	17.96	4
D-DG		DBLP-GoogleScholar	28,707	18.63	4
D-WA		Walmart-Amazon	10,242	9.39	5

- **BERT[5].** This is a simple baseline where the BERT language model is used to encode pairs of records into meaningful pair representations and a subsequent binary classifier is asked to predict *match* or *non-match* based on these representations;
- **SBERT[6].** SBERT is a modification of BERT that uses a siamese architecture to generate meaningful sentence embeddings whose distance approximates the sentence similarity. The objective of this training is very close to the one adopted in EM and therefore provides an alternative form of training for EM models. Similar to the BERT baseline, we use SBERT to produce a pair representation which is provided as input to a binary classifier;
- **Ditto[2].** Ditto is a RoBERTa-based model customized for solving EM by means of the application of domain knowledge injection and data augmentation to the input data;
- **R-SupCon[3].** R-SupCon is a RoBERTa-based model for product matching that applies a pre-training procedure based on supervised contrastive learning [14]. The idea is to force the model to create embedding representations that are close for descriptions referring to the same real-world entities and are far for different entities.

While Ditto and R-SupCon represent state-of-the-art EM methods, BERT and SBERT provide some baselines to evaluate the performance of off-the-shelf transformer-based architectures on the EM task without relying on further optimizations. For these models, we considered both a *pre-trained (PT)* and a *fine-tuned (FT)* version. The architecture of the pre-trained model extends the original language model with two fully connected layers of 100 and 2 neurons respectively (the 2 output neurons represent the *match* and *non-match* classes). These additional layers have been trained on the EM task to predict whether pairs of input records are matching, while the original pre-trained model remains unaltered. The fine-tuned architecture instead consists of a single classification layer inserted on top of the embedding corresponding to the [CLS] token, which summarizes the contents of the entire pair of records[3]. The whole architecture is here trained on the EM task, thus modifying the weights of the original language model.

[3]This is the usual standard practice adopted for fine-tuning language models to downstream tasks [2, 8].

Table 2
The effectiveness of the tested models in the EM task.

	DM+	BERT (pt)	BERT (ft)	SBERT (pt)	SBERT (ft)	Ditto	R-SupCon
S-FZ	100.00	97.67	97.67	97.67	100.00	97.78	92.68
S-DG	94.70	92.40	94.78	92.47	94.24	94.97	80.54
S-DA	98.45	97.41	98.65	96.84	98.30	96.86	99.21
S-AG	70.70	63.26	68.52	64.88	60.48	75.31	79.23
S-WA	73.60	59.89	78.85	60.23	78.05	85.40	80.12
S-BR	78.80	82.76	84.85	82.76	84.85	90.32	96.55
S-IA	91.20	85.19	93.10	77.19	93.10	92.31	85.71
T-AB	62.80	59.50	83.51	57.79	84.18	87.04	93.43
D-IA	79.40	84.21	94.74	75.00	93.10	83.64	68.18
D-DA	98.10	96.10	98.42	95.98	98.42	96.65	99.44
D-DG	93.80	92.27	94.77	91.22	95.05	94.86	80.13
D-WA	53.80	50.76	77.33	55.26	76.68	87.05	77.06
AVG	82.95	80.12	88.77	78.94	88.04	90.18	86.02
STD	15.40	17.03	9.90	16.19	11.71	6.74	10.04

3. Entity Matching effectiveness

This experiment evaluates the effectiveness of off-the-shelf transformer-based EM models (like the proposed BERT and SBERT baselines) compared to EM state-of-the-art methods. In addition to Ditto and R-SupCon, we also consider DeepMatcher (DM+)[13], which is a reference deep-learning-based EM approach that does not rely on a transformer architecture. The results are shown in Table 2, which reports the F1 score for each model.

Discussion. Even if *DM+* obtains good results with most datasets, off-the-shelf transformer-based EM models outperform it. This is particularly evident for the fine-tuned versions compared to the original pre-trained versions. Regarding the BERT-based EM baseline, fine-tuning improves the performance by around 8%, and by more than 10% with large dirty datasets (i.e., with more than 10k records).

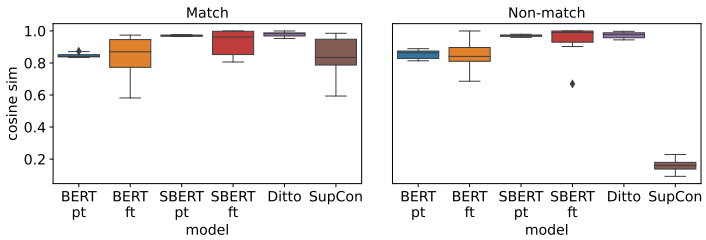
BERT and SBERT achieve similar accuracy levels in almost all datasets. Moreover, they both show better performance in dirty datasets than in structured datasets. This result is consistent with [8, 15], which show that transformer architectures are particularly robust to dirty data (e.g., where values are misplaced across attributes).

Ditto achieves the best effectiveness: it obtains an average F1 score of 90.18%, which is 2-4 points higher than the other tested models. This derives from the injection of domain knowledge and the application of a more advanced technique for encoding attribute values.

Finally, we observe that the average performance of R-SupCon is not as good as expected. It outperforms Ditto by about 4% in some datasets (e.g., T-AB, D-DA, S-DA, S-BR, and S-AG). However, it performs poorly with structured and dirty DBLP-GoogleScholar and iTunes-Amazon datasets (on average 12.5% lower). One of the reasons is that the approach was executed with the standard hyper-parameters, with no specific fine-tuning for the selected datasets.

	Match	Non-match
S-FZ	0.569	0.248
S-DG	0.538	0.171
S-DA	0.724	0.149
S-AG	0.423	0.220
S-WA	0.407	0.292
S-BR	0.553	0.264
S-IA	0.455	0.319
T-AB	0.192	0.133
D-IA	0.467	0.333
D-DA	0.691	0.164
D-DG	0.578	0.201
D-WA	0.438	0.324
AVG	0.503	0.235
STD	0.141	0.072

(a) Average Jaccard similarity between record pairs.



(b) Embedding similarity in BERT, SBERT, Ditto and R-SupCon.

Figure 1: Impact of different pre-training procedures in entity representation similarity.

4. The impact of the pre-training technique

This section investigates the importance of the technique adopted for pre-training transformer-based models in learning how to solve EM. The BERT model is pre-trained to perform two tasks: the prediction of masked words and the prediction of the next sentences. The effectiveness of these techniques has been largely demonstrated in many NLP problems [7]. However, we have limited knowledge of whether these pre-training techniques are the most effective for learning EM. Therefore, we wonder whether a different pre-training technique could improve the accuracy of EM tasks. We selected SBERT and R-SupCon because, as mentioned in section 2, they introduce alternative forms of pre-training based respectively on sentence similarity and on the knowledge of labels to produce similar representations for records referring to the same real-world entities.

More specifically, we analyze how entity representations change after different pre-training procedures. For each pair, we compute first the embeddings of both records[4] and then the similarity of the pair of embeddings. Table 1(a) shows the distribution of Jaccard similarities between records, divided by matching and non-matching pairs. These values provide a reference for evaluating the similarity of the embeddings. Matching pairs have a greater similarity than non-matching pairs, therefore we expect a model that can discriminate between matches and non-matches to encode this "distance" at the level of embeddings. The cosine similarity values for the embeddings computed by the tested models are shown in Figure 1(b).

Discussion. The pre-trained version of BERT and SBERT show a compact distribution of the cosine similarity of the embeddings. The fine-tuning step increases the variability of these results, but the median similarity remains approximately the same. We observe that BERT and SBERT generate very high cosine similarity (≥ 0.9) for both matching and non-matching records.

[4]We average the embeddings of the words in the record.

Table 3
BERT and R-SupCon accuracy in discovering entities.

	# cliques	Uncompleted cliques (%)		F1	
		BERT	R-SupCon	BERT	R-SupCon
Computers	83	13.25%	10.84%	92.82	88.78
Cameras	44	6.82%	4.55%	90.85	90.25
Shoes	44	20.45%	29.55%	89.04	74.25
Watches	70	17.14%	11.43%	94.47	80.95
S-DG (Valid)	50	18.00%	34.00%	94.78	80.54
D-DG (Valid)	50	24.00%	36.00%	94.77	80.13
AVG	56.83	16.61%	21.06%	92.79	82.48

Therefore, the embeddings similarity alone cannot tell whether the records refer to the same entity or not. This can probably be explained by the well-known *anisotropy phenomenon*: token embeddings occupy a narrow cone, resulting in a high similarity between any sentence pair [16]. Ditto shows a similar behavior: the median of the similarity does not significantly change in descriptions referring to matching and non-matching entities. This is expected since Ditto relies on the standard BERT architecture that does not train the model to learn this kind of knowledge. Conversely, R-SupCon is the only approach that learns a different behavior for matching and non-matching entity descriptions. The similarity of the generated embeddings is consistent with the Jaccard similarity shown in Table 1(a). This is the result of the contrastive learning technique, which requires that records referring to the same entity have closer embeddings than records of different entities.

5. Recognizing the entities

This experiment aims to evaluate the ability of transformer-based EM models to perform *Entity Resolution*, i.e., to identify groups of records that refer to the same real-world entity. Real-world entities are typically identified by computing the transitive closure of the matching decisions on pairs of records. This generates cliques, where the records included in each clique represent an entity [17]. The EM task is usually modeled in the literature as a binary classification problem. Therefore, the EM model cannot recognize multiple pairs of records referring to the same real entity. Nevertheless, evaluating if these models can preserve the cliques provides us insights into their understanding of the entity concept.

In this experiment, we examine how many cliques are recognized by the model with respect to the ground truth. Of all the datasets used in the previous experiments, only the S-DG and D-DG datasets generate cliques of size greater than 2. Therefore, we included the datasets describing laptops, cameras, shoes, and watches from the WDC benchmark[5]. We train an EM model on the training set from the benchmark, apply the model to the validation set, and calculate the cliques comprising descriptions of matching entities. In this experiment, we compare R-SupCon

[5]https://webdatacommons.org/largescaleproductcorpus/v2/index.html

Table 4
Robustness of BERT, SBERT, Ditto, and R-SupCon to out-of-distribution records.

Domain	Source	Target	BERT	SBERT	Ditto	R-SupCon
Same	S-WA	T-AB	0.50	0.48	0.53	0.33
	T-AB	S-WA	0.46	0.51	0.56	0.39
	S-DG	S-DA	0.95	0.96	0.92	0.96
	S-DA	S-DG	0.75	0.70	0.87	0.92
	D-DG	S-DA	0.96	0.95	0.94	0.96
	D-DG	D-DA	0.96	0.95	0.95	0.96
		AVG	0.76	0.76	0.80	0.75
Different	S-IA	S-DA	0.39	0.53	0.32	0.91
	S-IA	S-DG	0.37	0.46	0.31	0.78
	S-DA	S-IA	0.60	0.64	0.84	0.71
	S-DG	S-IA	0.83	0.79	0.45	0.75
	D-IA	D-DA	0.48	0.64	0.16	0.81
	D-DA	D-IA	0.55	0.67	0.72	0.61
		AVG	0.54	0.62	0.47	0.76
Total		**AVG**	0.65	0.69	0.63	0.76

with the BERT-based baseline. R-SupCon generates discriminative embeddings, that encode the similarity of the records; BERT presents similar behaviors compared to the other remaining models, as highlighted in the previous experiments.

Table 3 shows the results of the experiment. The first column reports the number of cliques in the ground truth. The other columns show the percentage of cliques not correctly recognized by the models and the accuracy obtained in terms of F1 score.

Discussion. Table 3 shows that an average of 16% of cliques are not recognized by the BERT model, even if the model reaches a high level of accuracy (more than 92% on average). The results of the experiment align with those reported in Section 4: since the model does not correctly recognize entities, it generates very similar embeddings for any pair of descriptions without distinguishing them based on the entity they belong to.

A similar result is achieved by R-SupCon, where the lower level of accuracy impacts the number of cliques found. However, R-SupCon finds more cliques in datasets on which the models have similar effectiveness.

6. Generalization to out-of-distribution records

In this experiment, we evaluate the robustness of EM models against out-of-distribution data, i.e., their behavior with data that differs from the training set. The experiment is inspired by [18], which explores domain adaptation techniques for deep EM models. Following a similar experimental evaluation, we evaluate the EM models against two scenarios. In the first scenario, we experiment with test sets from the same domain as the training data. For instance, we train

the EM models with S-WA and we evaluate them against T-AB, since both datasets describe products. The second scenario, on the other hand, evaluates the performance of models where the training sets and the test sets are from different domains. Table 4 shows the experiment results.

Discussion. In the first scenario, we observe that the EM models exhibit high performance reaching an average F1 score in the range of 0.75-0.80. For the datasets S-DA and D-DA, the scores are really close to the ones achieved with the training and testing set from the same dataset (see Table 2). The poorest results concern the experiments involving T-AB. This dataset is structurally different from S-WA even if it belongs to the same domain, because it includes large textual attributes. In the second scenario, where training and test datasets are from different domains, the performance decreases for all models apart from R-SupCon. This could be the result of the contrastive learning technique implemented in the model which makes the approach able to better generalize than the other learning techniques.

7. Conclusion

Summarizing the results obtained from the experiments, we observe that:

1. *Off-the-shelf transformer-based EM models outperform previous deep-learning-based EM models (like DeepMatcher[13]) and perform well even in dirty data, where values are misplaced across attributes;*
2. *Different pre-training tasks result in different effectiveness performance, which is only partially motivated by a different learning of record representations.* We compared four EM models, each pre-trained with a different method: the usual word-masking technique, the sentence-similarity-based task offered by SBERT, and R-SupCon based on contrastive learning. This showed that only R-SupCon can differentiate the knowledge encoded in the embeddings between matching and non-matching records (Section 4).
3. *Models that are fine-tuned for EM via a binary classifier do not fully recognize cliques of entity descriptions (Section 5) and have limited generalization capacity to out-of-distribution data* (Section 6).

We conclude that, even if transformer-based architectures represent a breakthrough in performing EM (Section 3), the reasons why they largely support the process can be only partially explained. Thus, we believe that there is still room to instill human rationales regarding the resolution of matching tasks within these architectures. In addition, exploring more advanced forms of fine-tuning and pre-training represents a concrete direction to make the behavior of such models more self-explanatory and promote their application in real-world scenarios.

References

[1] N. Barlaug, J. A. Gulla, Neural networks for entity matching: A survey, ACM Trans. Knowl. Discov. Data 15 (2021) 52:1–52:37.

[2] Y. Li, J. Li, Y. Suhara, A. Doan, W. Tan, Deep entity matching with pre-trained language models, Proc. VLDB Endow. 14 (2020) 50–60.

[3] R. Peeters, C. Bizer, Supervised contrastive learning for product matching, in: WWW (Companion Volume), ACM, 2022, pp. 248–251.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017, pp. 5998–6008.
[5] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT (1), Association for Computational Linguistics, 2019, pp. 4171–4186.
[6] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 3980–3990.
[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019).
[8] U. Brunner, K. Stockinger, Entity matching with transformer architectures - A step forward in data integration, in: EDBT, OpenProceedings.org, 2020, pp. 463–473.
[9] M. Paganelli, F. D. Buono, M. Pevarello, F. Guerra, M. Vincini, Automated machine learning for entity matching tasks, in: EDBT, OpenProceedings.org, 2021, pp. 325–330.
[10] A. Baraldi, F. D. Buono, F. Guerra, M. Paganelli, M. Vincini, An intrinsically interpretable entity matching system, in: EDBT, OpenProceedings.org, 2023.
[11] M. Paganelli, D. Tiano, F. Guerra, A multi-facet analysis of bert-based entity matching models, The VLDB Journal (2023) 1–26. doi:10.1007/s00778-023-00824-x.
[12] M. Paganelli, F. D. Buono, A. Baraldi, F. Guerra, Analyzing how BERT performs entity matching, Proc. VLDB Endow. 15 (2022) 1726–1738.
[13] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, Deep learning for entity matching: A design space exploration, in: SIGMOD Conference, ACM, 2018, pp. 19–34.
[14] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, in: NeurIPS, 2020.
[15] Y. Lin, Y. C. Tan, R. Frank, Open sesame: Getting inside bert's linguistic knowledge, CoRR abs/1906.01698 (2019).
[16] T. Jiang, S. Huang, Z. Zhang, D. Wang, F. Zhuang, F. Wei, H. Huang, L. Zhang, Q. Zhang, Promptbert: Improving BERT sentence embeddings with prompts, CoRR abs/2201.04337 (2022).
[17] D. Firmani, B. Saha, D. Srivastava, Online entity resolution using an oracle, Proc. VLDB Endow. 9 (2016) 384–395.
[18] J. Tu, J. Fan, N. Tang, P. Wang, C. Chai, G. Li, R. Fan, X. Du, Domain adaptation for deep entity resolution, in: SIGMOD Conference, ACM, 2022, pp. 443–457.