

Interpretable Entity Matching with WYM

Andrea Baraldi, Francesco Del Buono, Francesco Guerra, Giacomo Guiduzzi, Matteo Paganelli and Maurizio Vincini

University of Modena and Reggio Emilia, Via P. Vivarelli 10, Modena (MO), Italy

Abstract

This paper introduces WYM (Why do You Match?), an intrinsically explainable model designed for Entity Matching (EM). WYM is built upon decision units, which are basic information units formed by either pairs of similar terms belonging to different entity descriptions, or unique terms present in only one of the descriptions. Decision units enable the definition of a new feature space that can compactly and meaningfully represent pairs of entity descriptions. By training an explainable binary classification model on these features, WYM generates customized and effective explanations for EM datasets.

Keywords

XAI, Entity Matching, Machine Learning, Deep Learning

1. Introduction

The task of Entity Matching (EM), which involves determining if entries in a dataset refer to the same real-world entity, is a difficult challenge even for human experts. While Machine Learning (ML) and Deep Learning (DL) models are highly accurate, they suffer from low interpretability, creating possible critical problems in operational scenarios where the accuracy of the model is as important as the ability to understand its behavior.

Issues related to interpretability are mainly addressed in the literature in two ways: 1) by exploiting post-hoc analysis [1] or 2) by designing models that incorporate interpretability into their data structures. The main difference between these kinds of techniques lies in the trade-off between model accuracy and explanation fidelity. Inherently interpretable models could provide accurate and undistorted explanations but may sacrifice prediction performance to some extent [2]. This happens also for BERT-based EM systems [3]. Most of the tools proposed in the literature for explaining EM models (i.e., *LIME* [4], *SHAP* [5], *Explainer* [6], *Mojito* [7], *LEMON* [8], *Landmark Explanation* [9], and *CERTA* [10]) are post-hoc approaches. WYM [11] is the only intrinsically explainable EM system available among the deep learning based EM models.

The explanation of a model consists of an impact value associated with each input feature, which represents its weight in the decision-making process of the model [12]. However, in the context of EM where records describe pairs of entities, a feature-based representation of the


SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy

✉ andrea.baraldi96@unimore.it (A. Baraldi); francesco.delbuono@unimore.it (F. D. Buono); francesco.guerra@unimore.it (F. Guerra); giacomo.guiduzzi@unimore.it (G. Guiduzzi); pagamatteo@gmail.com (M. Paganelli); maurizio.vincini@unimore.it (M. Vincini)

ORCID 0000-0002-1015-5490 (A. Baraldi); 0000-0003-0024-2563 (F. D. Buono); 0000-0001-6864-568x (F. Guerra); 0000-0003-0819-405X (G. Guiduzzi); 0000-0001-8119-895X (M. Paganelli); 0000-0001-9262-2939 (M. Vincini)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

explanations can result in usability issues. Specifically, records may contain a significant number of features, leading to complex explanations that are challenging to manage and comprehend for users [13, 8]. Moreover, records representing matching entities are prone to a high level of duplicated terms, making the explanations difficult to read and interpret. To resolve this issue, it is necessary to specify which entity description the duplicated features belong to, i.e., either the left entity or the right entity, to avoid confusion and uncertainty in feature weights.

To address the challenges of EM model interpretability, we proposed WYM [11] (Why do You Match?), an intrinsically explainable EM model based on “*decision units*”. Decision units are intuitive information units that acknowledge records as pairs of entity descriptions. They can be either *paired* or *unpaired*. Paired decision units consist of semantically similar features (i.e., tokenized words in the case of textual datasets) found in the descriptions of different entities, while unpaired decision units represent isolated features from an entity description that lack a corresponding feature in the other description. By using decision units as the feature space for training an intrinsically explainable EM model, WYM offers a more intuitive and interpretable explanation for its decisions. The core of WYM consists of three main components: the *Decision unit generator*, which computes the decision units from the dataset records; the *Decision unit relevance scorer*, which assigns weights (relevance scores) to each decision unit based on its importance in the matching decision; and the *Explainable matcher*, which computes the match prediction and generates the explanations by associating a contribution score to each decision unit. An additional component, the *Explanation analysis tool*, is in charge of analyzing the results of the Explainable matcher for generating counterfactual (i.e., explanations where the smallest change to the feature values flips the prediction to the opposite output [12]) and exemplary (i.e., a subset of representative explanations from the ones from the entire dataset) explanations.

This paper is an extended abstract of paper [11] where the effectiveness of decision units in providing an explanation for the results of an EM model is introduced.

2. The WYM Explainable Matcher

The WYM functional architecture for generating intrinsically interpretable predictions for entity descriptions is shown in Figure 1. It comprises four main components: the *Decision unit generator*, the *Relevance scorer*, the *Explainable matcher*, and the *Explanation analysis tool*. In the following, we suppose that entity descriptions share the same schema.

Decision unit generator.

This component implements three main functionalities. Firstly, a word-piece tokenization with stop word removal is applied. Then we apply the BERT language model (fine-tuned for the task at hand) to encode the meaning of the entity descriptions into contextualized embeddings. Finally, embeddings from an entity description are possibly paired with the ones of the second description, thus forming paired decision units. For this last operation, we leverage the schema of the dataset, if any, to reduce the alignment space where an adaptation of the Gale–Shapley implementation of the Stable Marriage algorithm is applied.

Example. Let us consider the dataset iTunes-Amazon dataset from the Magellan benchmark.

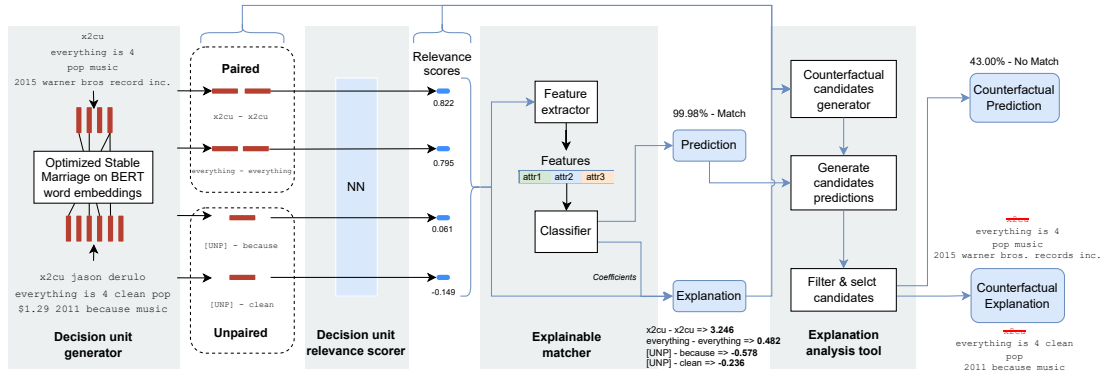


Figure 1: The process for generating a prediction and an explanation with WYM.

The goal is to understand when descriptions refer to the same song and the impact of the decision units in the prediction. The Decision unit generator takes the descriptions in the dataset as input as shown in the left part of Figure 1 and generates paired and unpaired decision units. The Figure shows two examples of paired and unpaired units. Among them, [x2cu, x2cu], the name of the song, is an example of paired decision unit, [clean], a word that is part of the album title in the second description, is an example of an unpaired decision unit.

Decision unit relevance scorer.

To assign relevance scores to decision units in the matching process, WYM utilizes a supervised regression model trained using a dataset where each entry represents a decision unit, and the target class is estimated by applying a heuristic rule, which relies on the class to which the decision unit belongs and the frequency of co-occurrence of the decision unit to the target class computed on the entire dataset.

Example. The relevance score assigned to the paired decision unit [x2cu, x2cu] is 0.822, thus pushing WYM to consider the descriptions as matching. The score of the unpaired decision unit [clean] is -0.149, thus pushing the classifier to consider the descriptions as non-matching.

Explainable matcher

The relevance scores assigned to the decision units provide an estimate of their contribution to the overall prediction. We enhanced this estimation by incorporating contextual and structural knowledge. We introduced three types of knowledge by aggregating features and scores based on attribute, entity description, and record. To accomplish this, we employed various statistical operators such as max, min, and count, which were applied to the decision units. The new dataset is used to train a binary classifier (a logistic regression classifier) that infers if the pairs of entity descriptions refer to the same real-world entity. Finally, we leverage the interpretability of the selected binary classifier to estimate the effect that each decision unit has on the prediction. We begin by extracting the learned coefficients from the classifiers, which indicate the importance of each generated feature. Next, we employ an inverse feature engineering transformation to identify the units that contribute to each feature and associate them with the impact score.

non-matching. Counterfactual explanations may lack meaning when applied to non-matching entities, particularly when the entities are vastly dissimilar, and the inclusion of additional features to establish a match would result in a significant alteration of the original description.

WYM implements the following heuristics to generate counterfactual explanations for matching entity descriptions: *MoRF*: the Most Relevant Features are removed first, thus allowing the users to identify the features *needed* for the explanation. A probabilistic variation of the MoRF heuristic is also implemented to generate many counterfactual explanations for the same record; *LeRF*: the Least Relevant Features are removed first, thus allowing the users to identify the features which are *sufficient* for the explanation. A probabilistic variation of the LeRF heuristic, generating many explanations for the same record, is also implemented; *random*: the features are randomly removed, thus generating reference baselines; *manual*: the user selects the decision units to remove.

Example. Figure 2c shows a *counterfactual* example generated by WYM with the MoRF strategy from the same prediction introduced in Scenario 1. The left part of the Figure shows the original pair of entity descriptions. On the right, we show the counterfactual explanation where the title and the duration of the song are removed. This means that selected units in the title and duration assume paramount importance in the prediction. The user can select other heuristics and the results are shown with the same tabular representation.

To compute counterfactual explanations for a pair of non-matching entity descriptions, WYM: 1) extracts from it the *positive explanation*, which only includes the “positive” decision units making the entity descriptions refer to the same real-world entity; 2) injects “negative” decision units into the positive explanation until the prediction changes again to non-matching. Since the goal is to identify the features that maximize the diversity between the descriptions, the most negative decision units are injected firstly.

Example 3. Figure 2d shows a *counterfactual* example for a non-matching entity prediction. The removal of unpaired decision units from the song title of both descriptions and from the genre of the first entity description makes WYM change the prediction class. This counterfactual explanation is consistent with the previous (and it is something somewhat expected from our domain knowledge): the terms in the song title are the ones that lead the model to understand if descriptions refer to the same real song.

Finally, the WYM system offers a feature that automatically identifies *exemplary explanations* from the ones generated for the entire dataset. These explanations are evaluated based on their ability to jointly satisfy three key metrics: explanation entropy, prediction-relevant units, and explanation overlap. Firstly, the explanation entropy metric is used to gauge the balance of token impacts in an explanation. An explanation with a low entropy (i.e., an unbalanced distribution of token impacts) is more intriguing than one with a high entropy (i.e., a near-uniform distribution), as it helps users identify the most significant units for prediction more easily. Secondly, the prediction-relevant units metric is used to evaluate the relevance of the decision units for the prediction, with a preference for a lower percentage of units that are relevant for prediction, as this provides a more concise and usable interpretation of the model’s behavior. Lastly, the explanation overlap metric is used to examine sets of explanations with complementary characteristics, such as explanations with different decision unit impact distributions at the attribute level, to achieve a more comprehensive interpretation of the EM model’s behavior.

Table 1

The Magellan Benchmark used in the experiments.

Dataset	Type	Datasets	Size	% Match
<i>S-DG</i>		DBLP-GoogleScholar	28,707	18.63
<i>S-DA</i>		DBLP-ACM	12,363	17.96
<i>S-AG</i>		Amazon-Google	11,460	10.18
<i>S-WA</i>	Structured	Walmart-Amazon	10,242	9.39
<i>S-BR</i>		BeerAdvo-RateBeer	450	15.11
<i>S-IA</i>		iTunes-Amazon	539	24.49
<i>S-FZ</i>		Fodors-Zagats	946	11.63
<i>T-AB</i>	Textual	Abt-Buy	9,575	10.74
<i>D-IA</i>		iTunes-Amazon	539	24.49
<i>D-DA</i>	Dirty	DBLP-ACM	12,363	17.96
<i>D-DG</i>		DBLP-GoogleScholar	28,707	18.63
<i>D-WA</i>		Walmart-Amazon	10,242	9.39

The user can specify the weight of each metric in the selection of emblematic explanations, and the Smooth Local Search technique [14] is applied to optimize the selection process.

3. Experimental evaluation

In this reduced extended abstract, the experimental evaluation focuses on demonstrating 1) how effective is WYM in solving EM tasks (Section 3.1) and 2) if the impact scores provide a reliable interpretation of the EM predictions (Section 3.2). Interested readers can refer to the paper [11] and the GitHub project at <https://github.com/softlab-unimore/WYM> for a complete experimental evaluation of the approach.

Datasets. The experiments are performed against 12 datasets provided by the Magellan library¹ which are usually considered the reference benchmark for the evaluation of EM tasks. In Table 1, we show some of their descriptive statistics: the total number of records representing matching entities in the fourth column and the percentage of records associated with a matching label in the last column. For the purposes of the experimental evaluation, each dataset is divided into training, validation, and test set which were created with 60-20-20 proportions.

3.1. Effectiveness of the EM Model

The effectiveness of WYM against the datasets in the benchmark in terms of F1 score is computed. The results are compared with the results achieved by DeepMatcher+² (DM+), one of the pioneering EM systems based on Deep Learning, AutoML [16], an approach that provides the automatic application of ML models to the EM problem by pipelining AutoML systems with transformer-based encoders, CorDEL [17] and *DITTO* [15], a contrastive DL approach and a BERT-based approach currently representing the state-of-the-art systems for solving the EM tasks. The results are shown in Table 2.

¹<https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md>

²DM+ is the combination of experiments / implementations as defined in [15]

Table 2

Effectiveness measured with the F1 score, and, in brackets, the rank of each model for each dataset. The comparison between WYM and the other approaches is shown in the right part of the table. Values are in bold (underlined) when they differ from WYM more than 3% (less than -3%).

Dataset	WYM	DM+	AutoML	CorDEL	DITTO	Δ	Δ	Δ	Δ
						DM+	AutoML	CorDEL	DITTO
						(%)	(%)	(%)	(%)
<i>S-DG</i>	0.936 (5)	0.947 (2)	0.940 (3)	0.940 (3)	0.956 (1)	-0.8	-0.1	-0.1	-1.7
<i>S-DA</i>	0.990 (3)	0.985 (4)	0.970 (5)	0.992 (2)	0.99 (1)	0.5	2	-0.02	0
<i>S-AG</i>	0.625 (5)	0.707 (3)	0.673 (4)	0.700 (2)	0.756 (1)	<u>-8.2</u>	<u>-4.8</u>	<u>-7.5</u>	<u>-13.</u>
<i>S-WA</i>	0.726 (4)	0.736 (3)	0.649 (5)	0.940 (1)	0.857 (2)	-0.1	7.7	<u>-21.4</u>	<u>-12.0</u>
<i>S-BR</i>	0.839 (3)	0.788 (5)	0.805 (4)	0.889 (2)	0.944 (1)	5.1	3.4	<u>-5.0</u>	<u>-10.5</u>
<i>S-IA</i>	1 (1)	0.912 (5)	0.922 (4)	1 (1)	0.971 (3)	8.8	7.8	0.0	2.9
<i>S-FZ</i>	1 (1)	1 (1)	0.969 (5)	1 (1)	1 (1)	0.0	3.1	0.0	0.0
<i>T-AB</i>	0.645 (4)	0.628 (5)	0.769 (2)	0.649 (3)	0.893 (1)	1.7	<u>-12.4</u>	-0.4	<u>-24.8</u>
<i>D-IA</i>	0.963 (1)	0.794 (5)	0.870 (3)	0.824 (4)	0.957 (2)	16.9	9.3	13.9	0.6
<i>D-DA</i>	0.972 (3)	0.981 (2)	0.969 (5)	0.970 (4)	0.99 (1)	-0.9	0.3	0.2	-1.8
<i>D-DG</i>	0.923 (4)	0.938 (2)	0.934 (3)	0.915 (5)	0.958 (1)	-1.5	-1.1	0.8	<u>-3.5</u>
<i>D-WA</i>	0.603 (3)	0.538 (4)	0.652 (2)	0.512 (5)	0.857 (1)	6.5	<u>-4.9</u>	9.1	<u>-25.4</u>
AVG	0.852 (3.1)	0.830 (3.4)	0.843 (3.8)	0.861 (2.8)	0.927 (1.1)				

Discussion. The overall WYM performance is slightly better than DM+, similar to AutoML and CorDEL, and worse than DITTO. The average F1 score measured on the overall benchmark is 0.852 (WYM), 0.83 (DM+), 0.843 (AutoML), 0.861 (CorDEL) and 0.927 (DITTO). If we consider a threshold of $\pm 3\%$ from the result achieved by our approach, where we consider the results to be similar, we observe that WYM performs better than DM+ in 4 datasets, worst in 1 dataset, and within the threshold in the remaining 7 datasets; it performs better than AutoML in 4 datasets, worst in 3 datasets, and within the threshold in the remaining 5 datasets; better than CorDEL in 2 datasets, worst in 3 and within the threshold in the remaining 7 datasets; finally, it performs worse than DITTO in 7 datasets, and within the threshold in the remaining 5 datasets. The detailed error analysis showed that WYM makes a large number of errors in recognizing product codes in the entity descriptions. In many cases, they form a decision unit even if they are not the same. This is mainly due to the tokenization mechanism introduced by BERT. Heuristics can be applied to address the problem. In particular, we verified an improvement of the F1 score in the T-AB dataset (from 0.645 to 0.754) after the insertion of domain knowledge that allows only equal product codes to belong to the same paired decision units.

3.2. Reliability of explanations

To evaluate the contribution of the impact scores assigned to the decision units to the overall accuracy of WYM, this experiment perturbs the dataset records by removing selected decision units and analyzing the performance variations on these synthetic datasets. We experiment with three techniques for the removal of the decision units applied to the datasets: 1) *MoRF*, where we eliminate for each record the k decision units that contribute most to the prediction (i.e. units with high positive impact in records describing matching entities and units with high negative impact for non-matching), 2) *LeRF*, where the k decision units that contribute less

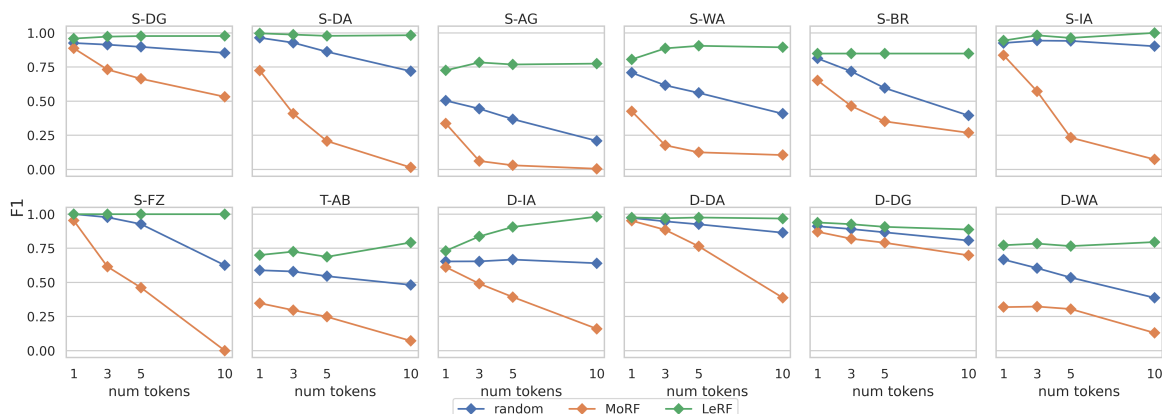


Figure 3: F1 score obtained in EM Models after the removal of the most relevant decision units (MoRF), less relevant decision units (LeRF) and random units.

to the prediction are removed (i.e. high negative impact in case of entity matches and high positive impact / in case of non-matches), and 3) *Random*, where k random decision units are removed. We expect that when we remove the most relevant decision units (MoRF) from records describing matching entities, the effectiveness (F1 score) will decrease; on the other hand, the model should not be affected by the removal of the least relevant units first (LeRF). The results of the experiment are shown in Figure 3, where, for each dataset, the F1 score generated by WYM as the removal technique varies, is reported.

Discussion. Analyzing the results we observe how impact scores assigned by WYM reflect the real importance of each token on the prediction. By perturbing the data with the *MoRF* strategy, WYM performance drops drastically (up to 60% in some datasets). The phenomenon is mostly marked as the number of removed units increases, however, in some datasets (such as Abt-Buy, Amazon-Google, and the two versions of Walmart-Amazon) the performance drops after the removal of a single unit. Moreover, the *LeRF* perturbation does not produce substantial variations in performance, which in most of the datasets slightly improves.

4. Conclusion

We presented WYM, i.e. an approach for performing interpretable entity matching that predicts if a pair of entity descriptions refer to the same real-world entity, and provides the terms (i.e., the decision units) that mainly led to the decision. As already pointed out in the literature [2], providing interpretability to the predictions comes with the price of decreasing the effectiveness of the approach. We consider WYM as a good compromise between the quality of the predictions and the capability of interpreting them. Other approaches for explainable EM (e.g., *DITTO*) definitely achieves the best performance, but acts for the users as an oracle that does not provide any support for understanding the reasons for its decisions. WYM obtains high quality results and provides decision units with the impact scores that can easily explain the predictions.

References

- [1] Z. Zhang, J. Singh, U. Gadiraju, A. Anand, Dissonance between human and machine understanding, *Proc. ACM Hum. Comput. Interact.* 3 (2019) 56:1–56:23.
- [2] M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, *Commun. ACM* 63 (2020) 68–77.
- [3] M. Paganelli, F. D. Buono, A. Baraldi, F. Guerra, Analyzing how BERT performs entity matching, *Proc. VLDB Endow.* 15 (2022) 1726–1738.
- [4] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?" explaining the predictions of any classifier, in: *Proceedings of the 22nd ACM SIGKDD*, 2016, pp. 1135–1144.
- [5] A. Ghorbani, J. Y. Zou, Data shapley: Equitable valuation of data for machine learning, in: *ICML*, volume 97, PMLR, 2019, pp. 2242–2251.
- [6] A. Ebaid, S. Thirumuruganathan, W. G. Aref, A. Elmagarmid, M. Ouzzani, Explainer: Entity resolution explanations, in: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, 2019, pp. 2000–2003.
- [7] V. D. Cicco, D. Firmani, N. Koudas, P. Merialdo, D. Srivastava, Interpreting deep learning models for entity resolution: an experience report using LIME, in: *aiDM@SIGMOD*, ACM, 2019, pp. 8:1–8:4.
- [8] N. Barlaug, Lemon: Explainable entity matching, *IEEE Transactions on Knowledge and Data Engineering* (2022) 1–16.
- [9] A. Baraldi, F. D. Buono, M. Paganelli, F. Guerra, Using Landmarks for Explaining Entity Matching Models, in: *EDBT*, 2021.
- [10] T. Teofili, D. Firmani, N. Koudas, V. Martello, P. Merialdo, D. Srivastava, Effective explanations for entity resolution models, in: *ICDE*, IEEE, 2022, pp. 2709–2721.
- [11] A. Baraldi, F. D. Buono, F. Guerra, M. Paganelli, M. Vincini, An intrinsically interpretable entity matching system, in: *EDBT*, OpenProceedings.org, 2023.
- [12] C. Molnar, *Interpretable Machine Learning*, 2019. <https://christophm.github.io/interpretable-ml-book/>.
- [13] S. Thirumuruganathan, M. Ouzzani, N. Tang, Explaining entity resolution predictions: Where are we and what needs to be done?, in: *HILDA@SIGMOD*, ACM, 2019, pp. 10:1–10:6.
- [14] U. Feige, V. S. Mirrokni, J. Vondrák, Maximizing non-monotone submodular functions, *SIAM J. Comput.* 40 (2011) 1133–1153.
- [15] 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.
- [16] 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.
- [17] Z. Wang, B. Sisman, H. Wei, X. L. Dong, S. Ji, Cordel: A contrastive deep learning approach for entity linkage, in: *ICDM*, IEEE, 2020, pp. 1322–1327.