Technical Report

Appendix for the Paper: An Intrinsically Interpretable Entity Matching System

Andrea Baraldi
University of Modena and Reggio Emilia
Modena, Italy
andrea.baraldi96@unimore.it

Francesco Del Buono
University of Modena and Reggio Emilia
Modena, Italy
francesco.delbuono@unimore.it

Francesco Guerra
University of Modena and Reggio Emilia
Modena, Italy
francesco.guerra@unimore.it

Matteo Paganelli
University of Modena and Reggio Emilia
Modena, Italy
matteo.paganelli@unimore.it

Maurizio Vincini
University of Modena and Reggio Emilia
Modena, Italy
maurizio.vincini@unimore.it

This appendix provides additional material on the experimental study in the paper "An Intrinsically Interpretable Entity Matching System" (Baraldi, Del Buono, Guerra, Paganelli, Vincini). This extra material has been omitted for space reasons in the submitted version. It consists of the following parts:

- adding Section 1.3 with experiments to evaluate the time performance of our approach;
- adding Section 1.4 with a small scale evaluation of the decision units with users.

The description of the experiments already included in the paper is also reported for the sake of completeness.

1 EXPERIMENTS

The experimental evaluation aims at answering four main questions: 1) How effective is WYM in solving EM tasks (Section 1.1); 2) If the impact scores provide a reliable interpretation of the EM predictions (Section 1.2); 3) If the time required to train the system and to compute the explanations make it usable in real world scenarios (Section 1.3); 4) If the decision-based explanations are effective for the users (Section **??**).

Datasets. The experiments are performed against 12 datasets provided by the Magellan library[1] 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. Figure 1 shows the average distribution of paired and unpaired decision units in the datasets. As expected, the overall number of units associated with non-matching descriptions is greater than the one of matching descriptions. Among the former, we see more unpaired than paired decision units. The *T-AB* dataset shows a different distribution, with a large number of unpaired units. The reason is that this is a database of large textual descriptions where the presence of periphrasis makes difficult the creation of paired units. 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. The

[1] https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md

Table 1: The Magellan Benchmark used in the experiments.

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

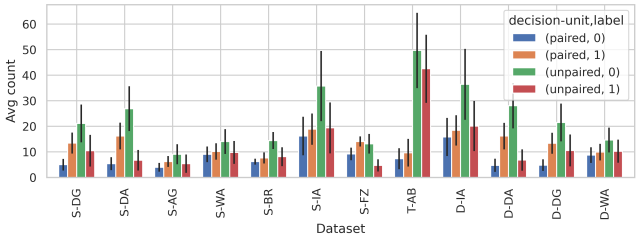


Figure 1: Average distribution of the decision units in the datasets.

implementation of WYM used in the experiments is available in the project github at https://github.com/softlab-unimore/WYM.

Settings. We run the experiments on a VM deployed on Google Cloud with 12 GB of RAM, GPU K80, and Intel(R) Xeon(R) CPU @ 2.30GHz. In all experiments, we adopted the following thresholds for the generation of the decision units: $\theta = 0.6$, $\eta = 0.65$ and $\epsilon = 0.7$.

1.1 Effectiveness of the EM Model

The effectiveness of WYM is evaluated according to three main perspectives: in Section 1.1.1, the performance is compared with competing systems; in Section 1.1.2, we evaluate how WYM behaves by varying the size of training sets, and in Section 1.1.3 we introduce a study of the components of the WYM architecture, evaluating different settings and implementation choices.

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 DM+ (%)	Δ AutoML (%)	Δ CorDEL (%)	Δ *DITTO* (%)
S-DG	0.936 (5)	0.947 (2)	0.940 (3)	0.940 (3)	0.956 (1)	-0.8	-0.1	-0.1	-1.7
S-DA	0.990 (3)	0.985 (4)	0.970 (5)	0.992 (2)	0.99 (1)	0.5	2	-0.02	0
S-AG	0.625 (5)	0.707 (3)	0.673 (4)	0.700 (2)	0.756 (1)	-8.2	-4.8	-7.5	-13.
S-WA	0.726 (4)	0.736 (3)	0.649 (5)	0.940 (1)	0.857 (2)	-0.1	**7.7**	-21.4	-12.0
S-BR	0.839 (3)	0.788 (5)	0.805 (4)	0.889 (2)	0.944 (1)	**5.1**	**3.4**	-5.0	-10 .5
S-IA	1 (1)	0.912 (5)	0.922 (4)	1 (1)	0.971 (3)	**8.8**	**7.8**	0. 0	2.9
S-FZ	1 (1)	1 (1)	0.969 (5)	1 (1)	1 (1)	0.0	**3.1**	0.0	0.0
T-AB	0.645 (4)	0.628 (5)	0.769 (2)	0.649 (3)	0.893 (1)	1.7	-12.4	-0.4	-24.8
D-IA	0.963 (1)	0.794 (5)	0.870 (3)	0.824 (4)	0.957 (2)	**16.9**	**9.3**	**13.9**	0.6
D-DA	0.972 (3)	0.981 (2)	0.969 (5)	0.970 (4)	0.99 (1)	-0.9	0.3	0.2	-1.8
D-DG	0.923 (4)	0.938 (2)	0.934 (3)	0.915 (5)	0.958 (1)	-1.5	-1.1	0.8	-3.5
D-WA	0.603 (3)	0.538 (4)	0.652 (2)	0.512 (5)	0.857 (1)	**6.5**	-4.9	**9.1**	-25.4
AVG	**0.852 (3.1)**	**0.830 (3.4)**	**0.843 (3.8)**	**0.861 (2.8)**	**0.927 (1.1)**				

1.1.1 Comparison with competing approaches. 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+ (DM+) [2], one of the pioneering EM systems based on deep learning, AutoML [8] [3], an approach that provides the automatic application of ML models to the EM problem by pipelining AutoML systems with transformer-based encoders, CorDEL [10] and *DITTO* [7], 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.

1.1.2 Learning Curves. The learning curves provide insight into the dependence of the EM model on the size of the training set. We select samples of increasing size from the training set and we evaluate the F1 score of the predictions on the test set. The dimensions of the experimented samples are 500, 1K, 2K records, and the entire dataset. Figure 2 shows the learning curves obtained. For sake of simplicity, the experiments were performed by using the encodings obtained with a pre-trained version of the BERT model, but the shape of the curve does not change by using fine-tuning. The curves for the datasets *S-BR*, *S-IA*, *S-FZ*, *D-IA* are not shown: the size of the training (270 records in *S-BR*) and test sets (90 elements in *S-BR*) are too small for a reliable evaluation.

1.1.3 Analysis of the WYM *components.* With the experiments in this Section, we aim to understand the contribution of each WYM component to the overall performance of the system.

The Decision Unit Generator. The experiment shows the contribution of the word embeddings on the creation of the decision units measured through the overall performance of the approach. The current implementation relies on the SBERT embeddings [9]. We show the performance obtained with two other kinds of embeddings: the pre-trained BERT model, and the BERT model fine-tuned on the EM task. Finally, the performance obtained with decision units computed on the basis of the Jaro-Winkler distance [11], i.e., an edit distance-based similarity measure, offers a simple baseline for the problem. The results of the experiment are shown in Section "Decision Unit Generator" of Table 3.

[2]DM+ is the combination of experiments / implementations as defined in [7]

[3]We average the results related to the best configuration (Hybrid-EM-Adapter) for AutoSklearn, AutoGluon and H20AutoML

Table 3: Effectiveness (F1 score) varying the component implementations. In brackets the rank of the model for each datataset.

	WYM	Decision Unit Generator			Scorer			Matcher
		j-w dist.	BERT-pt	BERT-ft	bin. scr.	cos. sim.	bin j-w	smp. feat.
S-DG	0.936 (1)	0.923 (6)	0.930 (4)	0.930 (4)	0.932 (3)	0.936 (1)	0.834 (8)	0.904 (7)
S-DA	0.990 (1)	0.980 (5)	0.989 (2)	0.986 (3)	0.976 (6)	0.983 (4)	0.965 (7)	0.952 (8)
S-AG	0.625 (2)	0.542 (6)	0.647 (1)	0.624 (3)	0.532 (7)	0.550 (5)	0.312 (8)	0.607 (4)
S-WA	0.726 (2)	0.710 (3)	0.670 (4)	0.592 (6)	0.458 (7)	0.748 (1)	0.281 (8)	0.611 (5)
S-BR	0.839 (8)	0.848 (7)	0.903 (3)	0.963 (1)	0.933 (2)	0.903 (3)	0.875 (6)	0.848 (5)
S-IA	1.000 (1)	1.000 (1)	1.000 (1)	1.000 (1)	0.981 (7)	0.982 (6)	0.898 (5)	0.836 (8)
S-FZ	1.000 (1)	1.000 (1)	0.977 (4)	0.955 (8)	0.977 (4)	1.000 (1)	0.957 (3)	0.977 (4)
T-AB	0.645 (1)	0.546 (4)	0.599 (2)	0.527 (5)	0.396 (7)	0.515 (6)	0.272 (8)	0.581 (3)
D-IA	0.963 (2)	0.653 (7)	0.982 (1)	0.929 (3)	0.828 (4)	0.821 (5)	0.513 (7)	0.755 (8)
D-DA	0.972 (1)	0.913 (7)	0.960 (2)	0.940 (6)	0.958 (5)	0.957 (4)	0.752 (8)	0.953 (5)
D-DG	0.923 (3)	0.850 (7)	0.925 (1)	0.924 (2)	0.897 (6)	0.911 (4)	0.585 (8)	0.907 (5)
D-WA	0.603 (1)	0.505 (6)	0.554 (4)	0.557 (3)	0.356 (7)	0.545 (5)	0.269 (8)	0.578 (2)
AVG	0.85 (2)	0.79 (5)	0.85 (2.4)	0.82 (3.8)	0.77 (5.4)	0.82 (3.8)	0.63 (7)	0.79 (5.3)

Table 4: Classifiers used as Explainable Matchers (F1 score). In bold, the best performance per dataset.

Dataset	LR	LDA	KNN	DT	NB	SVM	AB	GBM	RF	ET	Avg.	S.D.
S-DG	0.925	0.927	0.927	0.894	0.890	**0.936**	0.912	0.921	0.918	0.936	0.919	0.015
S-DA	0.982	0.971	0.984	0.949	0.963	**0.990**	0.977	0.968	0.977	0.985	0.975	0.012
S-AG	0.578	0.622	**0.625**	0.554	0.597	0.583	0.595	0.604	0.602	0.621	0.598	0.021
S-WA	0.721	0.709	0.632	0.639	0.572	0.720	0.686	**0.726**	0.716	0.719	0.684	0.050
S-BR	0.788	0.690	0.788	0.828	0.718	0.788	**0.839**	0.765	0.765	0.765	0.773	0.041
S-IA	**1.000**	0.852	0.906	0.912	0.787	**1.000**	0.947	0.947	**1.000**	0.982	0.933	0.067
S-FZ	0.977	0.977	0.977	0.977	0.977	0.977	**1.000**	0.955	**1.000**	**1.000**	0.982	0.014
T-AB	0.631	**0.645**	0.560	0.564	0.533	0.627	0.609	0.622	0.580	0.607	0.598	0.035
D-IA	0.909	0.760	0.760	0.871	0.696	0.830	**0.963**	0.931	0.881	0.877	0.848	0.077
D-DA	**0.972**	0.963	0.962	0.955	0.948	**0.972**	0.955	0.955	0.961	0.968	0.961	0.008
D-DG	**0.923**	0.922	0.902	0.893	0.887	0.918	0.898	0.913	0.919	0.921	0.910	0.013
D-WA	0.570	**0.603**	0.424	0.497	0.524	0.556	0.566	0.567	0.559	0.584	0.545	0.049
Avg.	0.831	0.803	0.787	0.794	0.758	0.825	0.829	0.823	0.823	0.830	–	–
S.D.	0.159	0.141	0.180	0.170	0.166	0.160	0.159	0.149	0.163	0.155	–	–

The Decision Unit Scorer. This component is based on a fully connected neural network to provide a relevance score to the decision units. We calculate the effectiveness of the overall model after the substitution of the neural network with (1) a binary score that assigns 1 to the paired decision units and 0 to the unpaired; and (2) a simple scorer based on the cosine similarity of the embeddings of the tokens. The binary approach is also applied to

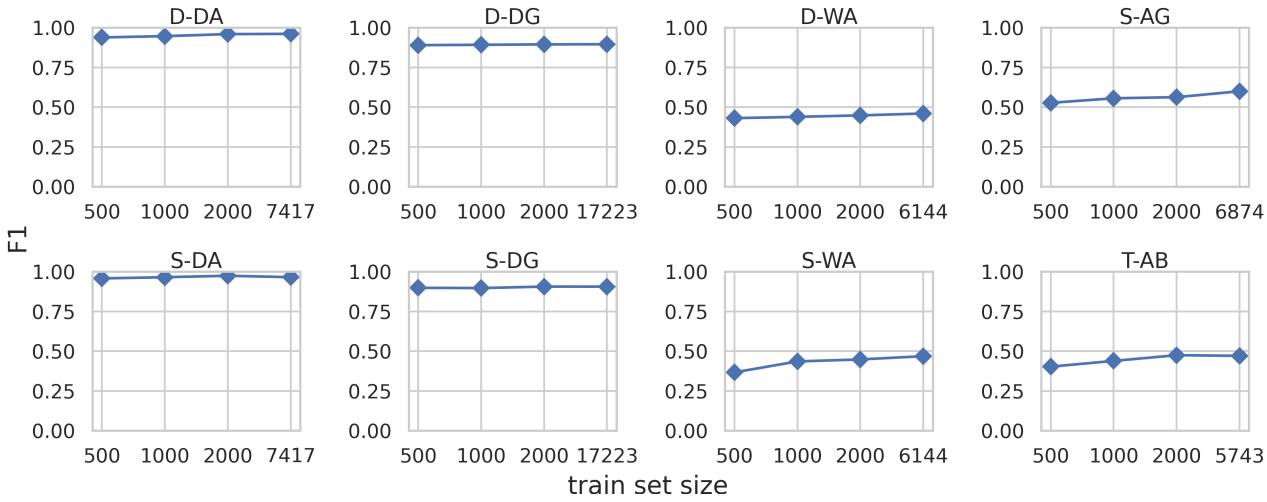


Figure 2: Learning Curves. The training set size is shown in the x-axes, the accuracy of the model (F1 score) in the y-axes.

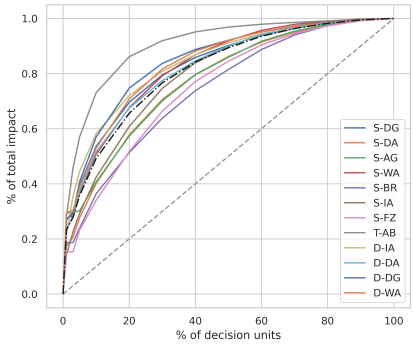


Figure 3: The conciseness of the generated explanation via a Pareto chart.

the decision units created with the Jaro-Winkler distance, providing a baseline for the approach. The results are published in the "Scorer" Section of Table 3.

The Explainable Matcher. This component generates the datasets to which the interpretable binary classifiers are applied. The datasets are created via feature engineering processes on the relevance scores. To evaluate the effectiveness of feature engineering, we perform an experiment with datasets composed of a limited number of features. In particular, the datasets contain 6 features, obtained by applying the count and the average operators on all the relevance scores, on the positive (the one pushing the decision toward the match), and on the negative scores (the one pushing towards the non-match). The F1 score achieved is shown in the "simplified feature" column of the "Matcher" Section of Table 3. Moreover, we perform a second experiment to compare the F1 scores achieved by all classifiers evaluated. The results of this experiment are reported in Table 4 (the last two rows and columns represent the average and the standard deviations on the datasets and classifier, respectively).

1.2 Interpretability of the EM Model

In this second set of experiments we investigate the interpretability of WYM by analyzing whether the impact scores can reveal the rationale behind the decisions. We evaluate this aspect in quantitative terms by analyzing the conciseness (Section 1.2.1), the faithfulless (Section 1.2.2), the contribution (Section 1.2.3) of the explanations and by comparing the WYM impacts with the explanations generated by a different tool (the Landmark EM explanation system – Section 1.2.4).

1.2.1 Conciseness. The conciseness can make the explanations consumable for humans, who cannot analyze the dozens of decision units part of a dataset record. Therefore, an explanation is usable if it can describe the prediction with few elements. In Figure 3, we show the results of the Pareto analysis performed for each record in every dataset by ordering the decision units per impact in descending order and plotting the cumulative values.

1.2.2 Faithfulness. We evaluate the faithfulness of the explanation to the EM Model via the notion of sufficiency, i.e., the ability of the top-impact elements to model a prediction [1, 4–6]. We adopt the post-hoc accuracy [3] to measure the sufficiency of the WYM explanations. For each test data, we select the top v important units based on the impact attributions for the model to make a prediction and compare it with the original prediction made on the whole input text. We compute the post-hoc accuracy on M examples,

$$post - hoc - acc(v) = \frac{1}{M}\sum_{m=1}^{M} 1[y_v^{(m)} = y^{(m)}] \quad (1)$$

where $y^{(m)}$ is the predicted label on the m-th test data, and $y_v^{(m)}$ is the predicted label based on the top v important words. Higher post-hoc accuracy indicates better explanations. Figure 4 shows the results of the experiments by using up to the top 5 decision units. We compared the values obtained in 3 settings: we consider WYM evaluated as EM model and explainer, WYM evaluated as EM model and explained with LIME, and *DITTO* explained with LIME.

1.2.3 Contribution of the decision units to the overall accuracy. 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

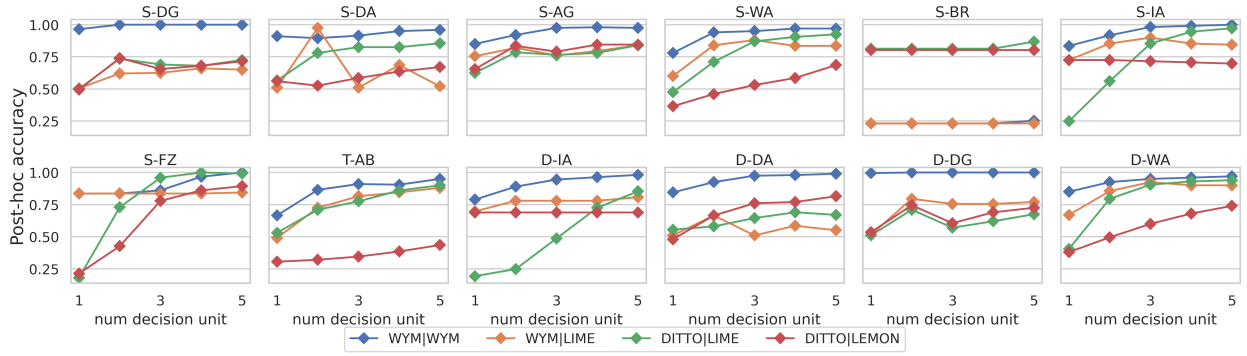


Figure 4: Sufficiency evaluated with the post-hoc accuracy. To high values correspond accurate explanations.

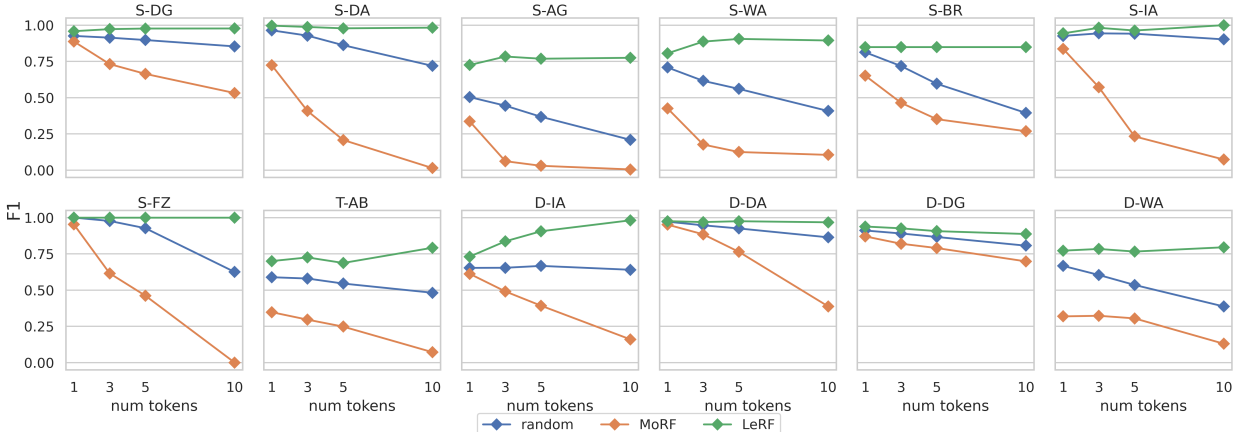
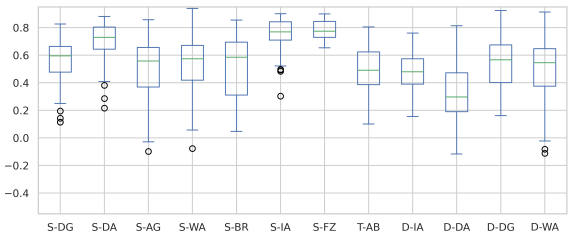


Figure 5: F1 score obtained in EM Models after the removal of the most relevant decision units (MoRF), less relevant decision units (LeRF) and random units.

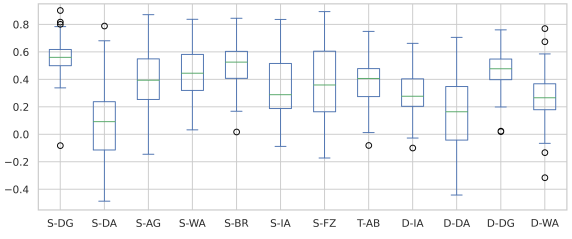
datasets. This is an extension of the post-hoc evaluation presented in the previous Section.

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 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 5, where, for each dataset, the F1 score generated by WYM as the removal technique varies, is reported.

1.2.4 Correlation with Landmark. The WYM explanations are compared with the ones generated by *Landmark Explanation*[2], a framework that extends the capabilities of a post-hoc perturbation-based explainer to the EM scenario. The experiment is performed by selecting a balanced sample of 100 elements from



(a) Impact score match records



(b) Impact score non-match records

Figure 6: Pearson correlation between the explanations generated by WYM and Landmark.

each benchmark dataset and using *Landmark* to compute the explanations[4]. Since it provides scores to tokens (and not to decision units), the explanations are post-processed by merging semantically similar tokens and averaging their scores. The outputs are then compared with the ones of WYM through the Pearson correlation measure. Figure 6 shows the results of the experiment, where the distribution of the correlation scores across all the records of each dataset is reported.

1.3 Time performance

We evaluate WYM time performance in terms of efficiency (Section 1.3.1). Moreover, we analyze the time breakdown of the pipeline in its components (Section 1.3.2).

1.3.1 Efficiency. To evaluate the efficiency of WYM, we computed the time required to train (with 40 epochs, learning rate $3e^{-5}$, and batch size 256) the EM system and to generate the explanations. Table 5 shows the time required to train WYM (we consider the entire process, from the entity descriptions to the trained model) in terms of time required per record and throughput (number of records processed per second); Table 6 shows the time required to compute the explanations (there is no difference between the time required to generate explanations for records in the test and validation sets).

Discussion. The Tables show that the time required for training is double the time for testing / validating the approach. The throughput in training is between 4.6 and 15.11 record per second (with an average of 9 record per second). The throughput in testing is between 9.1 and 27.86 record per second (with an average of 19.76 record per second). The training time is similar as the one shown by DITTO[5]. Concerning the average prediction (with no explanation) DITTO takes between 6.78 and 8.01 ms (similar times are performed by Deep Matcher). Our approach is one order of magnitude slower than DITTO, as shown in Table 6, but we compute the explanation.

Take away. The time required to train WYM and to generate the explanations are comparable with the ones of other EM systems. The time performance shows that WYM can be used for computing the explanations in typical real world scenarios (on average it can generate 70k+ explanations per hour).

1.3.2 Pipeline breakdown. We computed the time required for each component of the pipeline (generation of the embeddings, computation of the decision units, relevance score generation, training of the EM system, generation of the predictions, generation of the explanations), to evaluate if and where are the possible bottlenecks. Figure 7 show the percentage of time required by each component.

Discussion. The component that takes the large part of time in the training phase is the one in charge of computing the relevance scores (between 20 and 50% of time is spent in this phase). The generation of the embeddings is the most expensive phase in the testing/validating phases. We observe that WYM spend between 25% and 50% of time in the generation of the explanations (37% on average) in the testing phase.

[4]We configure Landmark Explanation to generate 100 perturbations for each entity of an EM record

[5]See Appendix B of the report in arxiv (https://arxiv.org/pdf/2004.00584.pdf) – the training time is shown in a small Table with a log scale. Nevertheless, for datasets around 10k records the time is close to 10^3 seconds, similar to the one shown by WYM (762 seconds for training S-DA with 7k records).

Table 5: Time required to process a record and throughput (train set).

TRAIN	Number of records	Time per record	Records per second
S-DG	17223	0.124	8.078
S-DA	7417	0.103	9.728
S-AG	6874	0.066	15.106
S-WA	6144	0.102	9.852
S-BR	268	0.156	6.418
S-IA	321	0.218	4.596
S-FZ	567	0.118	8.482
T-AB	5743	0.137	7.310
D-IA	321	0.205	4.883
D-DA	7417	0.106	9.470
D-DG	17223	0.119	8.382
D-WA	6144	0.092	10.929
AVG	6305	0.111	9.001

Table 6: Time required to process a record and throughput (test set – the validation set shows similar performance).

TEST	Number of records	Time per record	Records per second
S-DG	5742	0.0544	18.382
S-DA	2473	0.0555	18.018
S-AG	2293	0.0359	27.855
S-WA	2049	0.0546	18.315
S-BR	91	0.0662	15.106
S-IA	109	0.1101	9.083
S-FZ	189	0.0656	15.244
T-AB	1916	0.0634	15.773
D-IA	109	0.0955	10.471
D-DA	2473	0.0504	19.841
D-DG	5742	0.0487	20.534
D-WA	2049	0.0462	21.645
AVG	2102	0.0506	19.763

Take away. WYM was not conceived to be efficient. The analysis of the pipeline time breakdown shows where to focus the attention to optimize the process.

1.4 Users' evaluation

A small scale users' evaluation was performed to understand the quality of the decision unit-based explanations and to assess if this type of explanation is useful to understand the behaviour of the EM system. 15 users have been interviewed (colleagues and students from the ICT doctorate course at the University of Modena and Reggio Emilia) and the questionnaire reported in Appendix A has been administered. Three pairs of entity descriptions have been showed and the participants required to evaluate the feature-based explanations generated with DITTO and LIME and the decision unit-based explanations generated with WYM. The first pair of entity descriptions are referring to the same real entity; the second pair refers to different entities; the third pair is composed of the same description copied twice. The participants shown an overall good agreement in the answers (Fleiss' kappa =0.787). We report the answers achieved for the most significant questions:

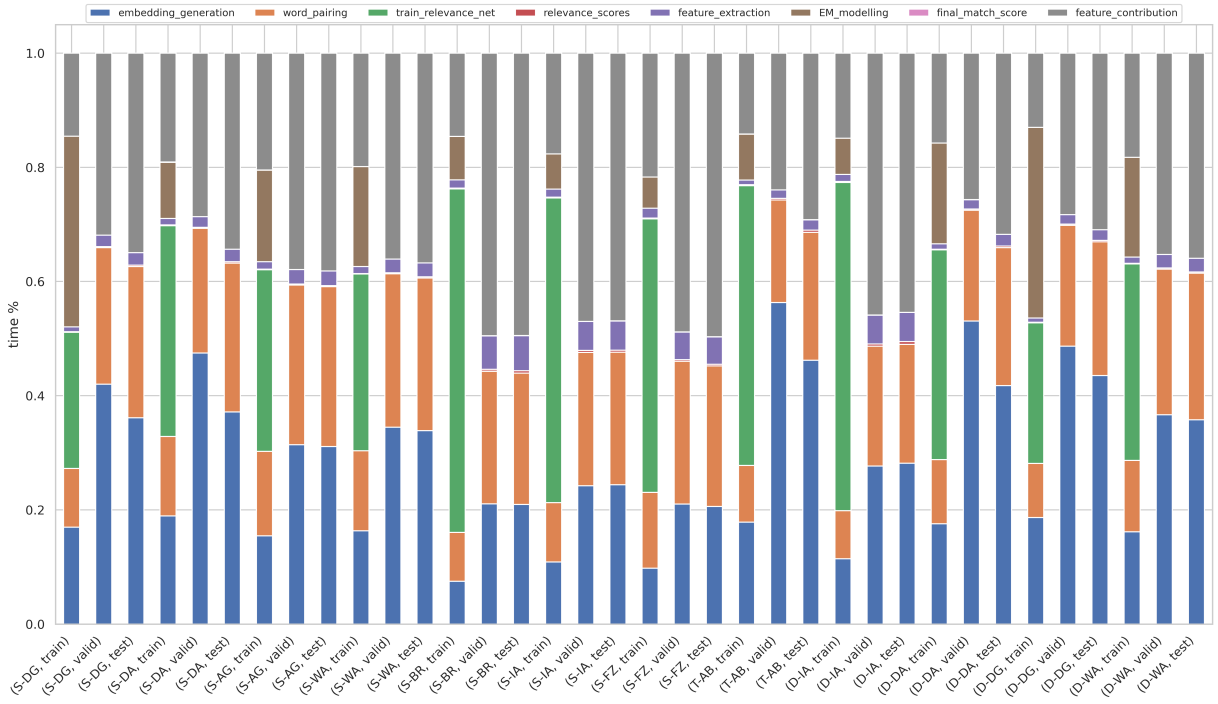


Figure 7: Pipeline time breakdown.

- **Question 1.2.** Only 2 participants (13%) reported that the feature-based explanation can explain that the first pair of entity descriptions are matching.
- **Question 2.2.** Only 1 participant (7%) thought that the feature-based explanation can explain that the second pairs of entity descriptions are not matching.
- **Question 3.2.** 13 participants (87%) thought that the feature-based explanation can explain the third pair of entity descriptions.
- **Questions 1.3, 2.3, 3.3.** All participants thought that decision unit-based descriptions can explain the decisions on the entity descriptions.
- **Questions 1.5, 2.5.** All participants thought that decision unit-based descriptions can better explain the prediction than the feature-based.
- **Questions 3.5.** 13 participants (87%) think thought that the decision unit-based descriptions can better explain the third prediction than the feature-based.

Discussion. The answers shows the limited capability of feature-based explanations to support users in understanding the EM predictions. Only when the entity descriptions are really close (as in the third pair) the participants appreciated the explanations.

Take away. The study involved a small number of users, but showed that users prefer decision unit-based explanations.

REFERENCES

[1] Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* 58 (2020), 82–115.

[2] Andrea Baraldi, Francesco Del Buono, Matteo Paganelli, and Francesco Guerra. 2021. Using Landmarks for Explaining Entity Matching Models. https://edbt2021proceedings.github.io/docs/p259.pdf. In *EDBT*.

[3] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *ICML (Proceedings of Machine Learning Research)*, Vol. 80. PMLR, 882–891.

[4] Mengnan Du, Ninghao Liu, and Xia Hu. 2020. Techniques for interpretable machine learning. *Commun. ACM* 63, 1 (2020), 68–77.

[5] Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan L. Boyd-Graber. 2018. Pathologies of Neural Models Make Interpretation Difficult. In *EMNLP*. Association for Computational Linguistics, 3719–3728.

[6] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding Neural Networks through Representation Erasure. *CoRR* abs/1612.08220 (2016).

[7] 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 (2020), 50–60.

[8] Matteo Paganelli, Francesco Del Buono, Marco Pevarello, Francesco Guerra, and Maurizio Vincini. 2021. Automated Machine Learning for Entity Matching Tasks. In *EDBT*. OpenProceedings.org, 325–330.

[9] 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

[10] Zhengyang Wang, Bunyamin Sisman, Hao Wei, Xin Luna Dong, and Shuiwang Ji. 2020. CorDEL: A Contrastive Deep Learning Approach for Entity Linkage. In *ICDM*. IEEE, 1322–1327.

[11] William Winkler. 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. *Proceedings of the Section on Survey Research Methods* (01 1990).

A QUESTIONNAIRE

The questionnaire delivered to the users is composed of 3 entity descriptions to analyze, as shown in Figures 8, 9, and 10.

1 Descriptions referring to the same entity.

Consider the following descriptions of product L and R:

Entity	Name	Description	Price
L	sirius stiletto 2 home docking kit slh2	sirius stiletto 2 home docking kit slh2 stereo audio output connects with home audio system or speakers headphone jack pc sync	49
R	sirius slh2 sirius ? stiletto 2 portable radio home kit	-	-

Notation:
In the following we use a sort of dot notation to refer to the entity descriptions:
`L.Name, sirius` is the term sirius in the content of the Name attribute of the Entity L.
`R.Name, sirius` is the term sirius in the content of the Name attribute of the Entity R.
`L.Description, sirius` is the term sirius in the content of the Description attribute of the Entity L.

1.1 Are in your opinion L and R describing the same product? (Yes/No)

1.2 Let us suppose that an Entity Matching system predicts that the entries describe the same elements and provides the impact of each term in the records for making the decision.
Do you think that the following terms (ordered by importance) can explain why the descriptions refer to the same entity? (Yes/No)

1. `L.Name, slh2`
2. `L.Description, slh2`
3. `R.Name, slh2`
4. `R.Name, 2`
5. `L.Description, output`
6. `L.Description, speakers`

1.3 Let us suppose that an Entity Matching system predicts the impact of term pairs (when possible) for making a decision. Do you think that the following (pairs of) terms (ordered by importance) can explain why the descriptions refer to the same entity? (Yes/No)

1. `(L.Name, R.name) stiletto, stiletto`
2. `(L.Name, R.name) slh2, slh2`
3. `(L.Name, R.name) sirius, sirius`

1.4 Is it useful to provide terms/pairs with “negative” impact (they push towards the opposite decision). (Yes/No) For example
`(L.Name, -) docking`
`(L.Name, -) system`
(Since the term docking (system) is only part of one description, the referred entities cannot be the same).

1.5 Which is the more useful explanation between 1.2 and 1.3?

1.6 Are there other pairs of terms/single terms that can provide a more powerful explanation other than the ones in 1.3?

Figure 8: Questionnaire, first page.

2 Descriptions referring to different entities.

Consider the following descriptions of product L and R:

Entity	Name	Description	Price
L	onkyo black 7.1-channel home theater system hts5100b	onkyo hts5100 black 7.1-channel home theater system hts5100b connect up to 3 hdmi-enabled sources audio playback from your ipo...	-
R	denon dht-589ba home theater system	a/v receiver , 5.1 speakers dolby pro logic ii , dts	448.8

2.1 Are in your opinion L and R describing the same product? (Yes/No)

2.2 Let us suppose that an Entity Matching system predicts that the entries describe the same elements and provides the impact of each term in the records for making the decision.
Do you think that the following terms (ordered by importance) can explain why the descriptions refer to the same entity? (Yes/No)

1. `R.Description, pro`
2. `R.Description, dts`
3. `L.Description, function`
4. `L.Description, chip`
5. `L.Description, room`
6. `L.Description, radio`

2.3 Let us suppose that an Entity Matching system predicts the impact of term pairs (when possible) for making a decision. Do you think that the following (pairs of) terms (ordered by importance) can explain why the descriptions refer to the same entity? (yes/no)

1. `(L.Name, R.name) hts5100b, dht-589ba`
2. `(L.Description, R.Description) dsp, dts`
3. `(L.Name, - ) onkyo`
4. `(-, R.Description) -, speakers`

2.4 Is it useful to provide terms/pairs with “negative” impact (they push towards the opposite decision). For example
`(L.Name, R.name) home, home`

2.5 Which is the more useful explanation between 2.2 and 2.3?

2.6 Are there other pairs of terms /single terms that can provide a more powerful explanation other than the ones in 2.3?

Figure 9: Questionnaire, second page.

3 Using the same description.

Consider the following descriptions of product L and R:

Entity	Name	Description	Price
L	nikon coolpix s610 digital camera midnight black 26125	10 megapixel 16:9 4x optical zoom 4x digital zoom 3 ' active matrix tft color lcd	-
R	nikon coolpix s610 digital camera midnight black 26125	10 megapixel 16:9 4x optical zoom 4x digital zoom 3 ' active matrix tft color lcd	-

3.1 Are in your opinion L and R describing the same product? (Yes/No)

3.2 Let us suppose that an Entity Matching system predicts that the entries describe the same elements and provides the impact of each term in the records for making the decision.
Do you think that the following terms (ordered by importance) can explain why the descriptions refer to the same entity? (Yes/No)

1. L.Name, s610
2. R.Name, s610
3. R.Name, 26125
4. L.Name, 26125
5. L.Name, black
6. R.Name, midnight

3.3 Let us suppose that an Entity Matching system predicts the impact of term pairs (when possible) for making a decision. Do you think that the following (pairs of) terms (ordered by importance) can explain why the descriptions refer to the same entity? (yes/no)

1. (L.Name, R.name) 26125,26125
2. (L.Name, R.name) s610, s610
3. (L.Description, R.Description) tft,tft

3.4
-

3.5 Which is the more useful explanation between 3.2 and 3.3?

3.6 Are there other pairs of terms/single terms that can provide a more powerful explanation other than the ones in 3.3?

Figure 10: Questionnaire, third page.