Effective Explanations for Entity Resolution Models

TOMMASO TEOFILI, Roma Tre University
DONATELLA FIRMANI, Sapienza University
NICK KOUDAS, University of Toronto
VINCENZO MARTELLO, Roma Tre University
PAOLO MERIALDO, Roma Tre University
DIVESH SRIVASTAVA, AT&T Chief Data Office

Entity resolution (ER) aims at matching records that refer to the same real-world entity. Although widely studied for the last 50 years, ER still represents a challenging data management problem, and several recent works have started to investigate the opportunity of applying deep learning (DL) techniques to solve this problem. In this paper, we study the fundamental problem of explainability of the DL solution for ER. Understanding the matching predictions of an ER solution is indeed crucial to assess the trustworthiness of the DL model and to discover its biases. We treat the DL model as a black box classifier and – while previous approaches to provide explanations for DL predictions are agnostic to the classification task – we propose the CERTA approach that is aware of the semantics of the ER problem. Our approach produces both saliency explanations, which associate each attribute with a saliency score, and counterfactual explanations, which provide examples of values that can flip the prediction. CERTA builds on a probabilistic framework that aims at computing the explanations evaluating the outcomes produced by using perturbed copies of the input records. We experimentally evaluate CERTA's explanations of state-of-the-art ER solutions based on DL models using publicly available datasets, and demonstrate the effectiveness of CERTA over recently proposed methods for this problem.

1 INTRODUCTION

Recent developments in Machine Learning (ML) and Deep Learning (DL) [22] have had a profound impact on several research communities, especially computer vision [34] and natural language understanding [7]. ML/DL has also had considerable impact on data management research, yielding alternate proposals for, among other topics, query optimization, selectivity estimation, approximate query processing, and entity resolution [5, 10, 16, 21]. Although DL models have demonstrated unparalleled prediction accuracy for very specific tasks, they are often criticized as offering predictions without any intuition or rationale [15].

Entity Resolution (ER) is the task that aims at matching records that refer to the same real-world entity. Although widely studied for the last 50 years [11], ER still represents a challenging data management problem. Recent works have investigated the application of DL techniques to solve the ER problem [5, 10, 16, 21].

A typical application of an ML model to the ER problem involves the training of a classifier, possibly a deep neural network, for this problem [5, 10, 16, 21]. Given a set of training data and associated labels (match or non-match), a classifier is trained to solve a binary classification problem. Subsequently given a pair of records, the records are suitably encoded and the classifier yields a binary prediction for the pair. As with any classification problem, it is assumed that future data follow the same distribution as that of the training data set. The ML classification models applied to this problem typically apply either traditional SVM [6], LSTM architectures [10, 21] or deep transformer architectures like BERT [5, 16]. Several recent approaches have demonstrated impressive prediction accuracy for the ER problem [4, 23].

Authors' addresses: Tommaso Teofili, Roma Tre University, tommaso.teofili@uniroma3.it; Donatella Firmani, Sapienza University, donatella.firmani@uniroma2.it; Nick Koudas, University of Toronto, koudas@cs.toronto.edu; Vincenzo Martello, Roma Tre University, v.martello@inf.uniroma3.it; Paolo Merialdo, Roma Tre University, paolo.merialdo@uniroma3.it; Divesh Srivastava, AT&T Chief Data Office, divesh@research.att.com.

	Name$_{Abt}$	**Description$_{Abt}$**	**Price$_{Abt}$**
u_1	sony bravia theater black micro system davis50b	sony bravia theater black micro...	NaN
u_2	altec lansing inmotion portable audio system ...	altec lansing inmotion ipod portable audio system im600usb...	NaN
u_3	sony 19 ’ bravia m-series silver lcd flat panel hdtv ...	sony 19 ’ bravia m-series silver lcd flat panel hdtv ...	NaN

(a) Abt

	Name$_{Buy}$	**Description$_{Buy}$**	**Price$_{Buy}$**
v_1	sony bravia dav-is50 / b home theater system	dvd player , 5.1 speakers 1 disc (s) progressive ...	NaN
v_2	altec lansing inmotion im600 portable audio ...		NaN
v_3	sony bravia m series ...	19 ’ atsc , ntsc 16:9 1440 x 900 ...	379.72

(b) Buy

Fig. 1. Sample records from the Abt-Buy dataset.

Input	**Ground-Truth**	**Ditto**	**DeepMatcher**	**DeepER**
$\langle u_1, v_1 \rangle$	Match	Match (0.98)	Match (0.71)	**Non-Match** **(0.01)**
$\langle u_2, v_2 \rangle$	Match	Match (0.93)	**Non-Match** **(0.16)**	Match (0.69)
$\langle u_3, v_3 \rangle$	Match	**Non-Match** **(0.002)**	Match (0.73)	Match (0.89)

Fig. 2. ER predictions performed by different DL systems on three pairs of the records from Figure 1. In brackets, the matching *score* of the system: for all the systems, $score \in [0, 1]$, and $score > 0.5$ corresponds to Match.

	Explanation (Saliency)			
ER System on tuple	**CERTA**	**Mojito**	**LandMark**	**SHAP**
DeepER on $\langle u_1, v_1 \rangle$	Description$_{Abt}$, Name$_{Buy}$	Name$_{Buy}$, Name$_{Abt}$	Name$_{Abt}$, Description$_{Buy}$	Description$_{Abt}$, Price$_{Buy}$
DeepMatcher on $\langle u_2, v_2 \rangle$	Description$_{Buy}$, Name$_{Buy}$	Description$_{Abt}$	Description$_{Abt}$, Price$_{Abt}$	Description$_{Abt}$, Name$_{Abt}$
Ditto on $\langle u_3, v_3 \rangle$	Description$_{Buy}$, Name$_{Buy}$	Description$_{Buy}$, Price$_{Buy}$	Description$_{Abt}$, Name$_{Abt}$	Price$_{Abt}$, Name$_{Buy}$

Fig. 3. Saliency explanations generated with different techniques for the wrong predictions of Figure 2.

Since DL models typically do not come with any explanations providing reasons for their predictions, an active research area has been the exploration of techniques to offer *explainable* predictions revealing the process the DL network followed to reach its decision [12].

Explanations represent an effective way to debug the system and are fundamental to trust its decisions, as they aim to provide the rationale behind a classifier's predicted outcome. For example, explanations are useful in situations where an ML classifier for ER makes wrong *predictions* (either classifies a match as non-match or vice-versa), as well as they can assist to check whether a classifier is making correct predictions for sound reasons.

Figure 1 reports some records from the *Abt-Buy* dataset, a popular benchmark for ER [21]. Figure 2 shows the predictions obtained for three such record pairs by three ER systems based on DL, namely DeepER [16], DeepMatcher [21], and Ditto [10]. The three pairs are in match, but all the three systems make mistakes on one of them (even Ditto, which performs very well, with $F1 \simeq 0.91$, on that dataset). Observe that the pairs in fact are rather similar: having explanations about the wrong predictions could help understand the roots of the misclassifications and improve the performance of the DL systems for ER.

Popular approaches to provide an explanation for an ML classifier output are based on *saliency* and *counterfactual* explanation methods [1, 19].

	Matching Score				
ER System on tuple	**Original**	**CERTA**	**Mojito**	**LandMark**	**SHAP**
DeepER on $\langle u_1, v_1 \rangle$	0.01	0.35	0.03	0.15	0.02
DeepMatcher $\langle u_2, v_2 \rangle$	0.16	0.97	0.17	0.24	0.16
Ditto on $\langle u_3, v_3 \rangle$	0.002	0.99	0.15	0.008	0.002

Fig. 4. Inspecting the faithfulness of saliency explanations generated with different techniques.

Saliency methods. These methods explain the prediction of the classifier by assigning a *saliency* score to each feature in the specific prediction input. This way, the features that influence the predicted outcome the most can be identified. In the context of explaining the results of a classifier for ER, saliency methods aim at identifying the most influential attributes in an input pair, with respect to the predicted outcome. In the example of Figure 2, a saliency method should identify which attributes in the pair $\langle u_3, v_3 \rangle$ are influencing Ditto predict it as a non-match the most. Notable examples of saliency methods are LIME [26] and SHAP [18], which were conceived for generic classification tasks on textual data and images, ignoring the semantics of the problem the classifier is used to solve. Mojito [8] and LandMark [3] represent adaptations of these methods specifically tailored for the ER task. Saliency explanation methods are sometimes also referred as feature attribution methods in literature.

Counterfactual explanations. These methods help understanding the behavior of the system by providing modified copies of the original input that lead to a different predicted outcome than the original prediction. In our example, a counterfactual explanation can help answering the question *"how the pair $\langle u_3, v_3 \rangle$ should be (minimally) changed in order to make Ditto predict it as a match?"*. Counterfactual explanations for ER systems, to the best of our knowledge, have not been explored at all in the literature, while there are several task agnostic methods, including DiCE [20], and the counterfactual versions of LIME and SHAP, *LIME-C* and *SHAP-C* [25].

It has been observed that saliency and counterfactual explanation methods are different but complimentary methods to be used to best evaluate causality aspects of a classifier prediction [14]. Saliency methods align well with the notion of *necessity*, while counterfactual explanation methods align with the notion of *sufficiency* [36].

This paper presents CERTA, an original method that provides both saliency and counterfactual explanations for ER systems. CERTA considers specific characteristics of the ER task, and builds on the sound theoretical framework developed by Watson *et al.* [36], which frames the concepts of probability of necessity and probability of sufficiency in the context of explanations.

While previous proposals [3, 8] represent interesting attempts to provide explanations to ER systems, they lack a theoretical foundation and the effectiveness of their explanations is limited.

Figure 3 shows the saliency explanations generated by CERTA, Mojito, LandMark and SHAP for the wrong predictions of Figure 2. Observe that the four approaches produce different explanations. For example, CERTA indicates that the most influential attributes for the DeepER results are Description from the Abt table and Name from the Buy table (denoted as Description$_{Abt}$ and Name$_{Buy}$, respectively), while Mojito identifies Name$_{Buy}$ and Name$_{Abt}$.

Similarly, Figure 5 shows counterfactual explanations generated by CERTA and by DiCE for the prediction of DeepER on the pair $\langle u_1, v_1 \rangle$. For each method, we report in boldface the values of the generated explanation that should flip the prediction (from non-match to match). Note that the different explanations provide contrasting results.

Given such a diversity of results, one may wonder which explanation is the most faithful to the actual behavior of the ER system. For saliency methods, one way to evaluate the effectiveness of an explanation consists of computing a new prediction using as input an altered pair, where the values of the attributes indicated by the saliency method are copied into the other tuple. For example, in evaluating the faithfulness of LandMark, copying the value of Name$_{Abt}$ into Name$_{Buy}$, and the value of Description$_{Buy}$ into Description$_{Abt}$. As the tuples have been made more similar by the attributes that most influenced the decision, it is expected that the matching score of the classifier increases. Similarly, for a counterfactual explanation it is possible to check how the values suggested by the explanation method change the prediction.

Figure 4 shows the original matching on the original input pairs and those obtained by modifying the input pairs according to the explanations of Figure 3. For all the methods but CERTA, the matching scores do not change significantly, even if the tuples have been made more similar by following the insights of the explanations. Apparently, the saliency computed by these explanation methods does not reflect the importance of the attributes for the decisions of the ER systems. In contrast, the explanation generated by CERTA changes the matching score a lot. Similarly, Figure 5 reports the matching score of DeepER on the pair modified as suggested by the explanation. Also in this case, it is easy to observe that CERTA produces a more effective explanation, which actually forces the system to flip the prediction (since the resulting matching score is greater than 0.5).

In Section 5 we provide results of an extensive evaluation that demonstrates the superiority of CERTA in a wider experimental setting.

Contributions. We make the following contributions in the context of providing explanations for ER models:

- We present the CERTA algorithm, which can exploit the semantics of the ER problem to provide saliency and counterfactual explanations that are quantitatively effective with respect to previous approaches.
- We introduce the first counterfactual explanation technique for ER classifiers.
- We present a principled framework based on the notions of probability of necessity and sufficiency and lattice structures.
- We experimentally evaluate CERTA's explanations of state-of-the-art ER solutions based on DL models using publicly available datasets, and demonstrate the effectiveness of CERTA over recently proposed methods for this problem.

Paper outline. Section 2 discusses related work. Section 3 introduces the problem statement. Section 4 describes our approach to efficiently compute saliency and counterfactual explanations. Section 5 presents the experimental evaluation that we have conducted. Section 6 discusses concluding remarks and future work.

		Counterfactual explanation					
	Matching Score	**Name**$_{Abt}$	**Description**$_{Abt}$	**Price**$_{Abt}$	**Name**$_{Buy}$	**Description**$_{Buy}$	**Price**$_{Buy}$
CERTA	0.54	sony bravia theater black micro system davis50b	**denon 5-disc cd auto changer dcm290 cd-r/rw playback advanced ...**	NaN	sony bravia dav-is50 / b home theater system	"dvd player , 5.1 speakers 1 disc (s) progressive scan...	NaN
DiCE	0.34	**lg 14 ' washer and dryerred pedestal ...**	sony bravia theater black micro system davis50b 5.1-channel surround	NaN	**canon pixma mx700 multifunction photo ...**	**lithium ion (li-ion) 8.4 v dc photo battery**	NaN

Fig. 5. Counterfactual explanations by CERTA and *DiCE* for the DeepER prediction on $\langle u_1, v_1 \rangle$: the values of the attributes identified by each counterfactual explanation method are highlighted in **bold**. The matching score is computed modifying the original pair using the values suggested by the counterfactual explanation. The original matching score is equal to 0.01 (Non-Match), as reported in Figure 2.

2 RELATED WORKS

Much recent research has been conducted in the context of *explainable AI* [12]. Explanation systems can be divided into different categories, in particular we focus on saliency and counterfactual explanation systems. Saliency explanation systems describe the relationship between input features and the output of a model, for example providing a relevance score for each feature. One of the best known systems is LIME [26], which aims at explaining the prediction of any classifier for text, images or tabular data. Another explanation system, called SHAP [18], develops a saliency explanation scheme based on game theoretic concept of Shapley values. All such methods can be applied in principle to any classification task, including ER. However, in the case of ER the classification task takes as input pairs of records rather than a single record (e.g., as in image classification tasks) and using the mentioned general purpose explanation methods may not be desirable. We refer the reader to [31, 35] for further discussion on the problem of providing explanation methods for the ER task and for data integration in general.

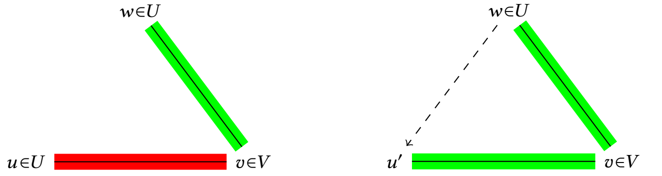
More recently, new explanation systems have been proposed for the ER task, namely, Mojito [8], ExplainER [9] and LandMark [3]. Mojito [8] provides an adaptation of a general purpose explanation method – that is, LIME [26] – on ER models. Mojito introduces two specific operations: "Mojito pre-processing", which transforms a record pair to a string representation, and "LIME COPY", which generates new record pairs in conjunction with the standard "DROP" operator provided by LIME. LandMark [3] provides a further adaptation of LIME to the specific setting of Entity Resolution. It internally generates two explanations for each record pair, each one explaining the classifier (with LIME) when the other record is kept unchanged. ExplainER [9] provides a unified graphical user interface to identify representative pairs to understand the model's behaviour and identify attributes that are overall more influential. In the back-end, ExplainER can plug-in different general purpose explanation systems (including, LIME [26] and Anchors [27]) by modelling the ER task as a binary text classification task. We note that Mojito, ExplainER, LandMark consist of more or less advanced adaptations of general purpose methods to the ER task, and do not provide any new explanation method. A complementary approach to explainable ER was recently proposed by SystemER [24]. Even though SystemER is not an explanation system, it enables the user to learn an inherently explainable ER model, with human-comprehensible rules and the desired level of quality, by involving expert humans in the loop.

Several counterfactual explanation approaches have been developed [29, 32]. For the sake of this work, we consider counterfactual explanation methods that can treat the ER classifier as a black box function. In this context, model agnostic counterfactual explanation approaches that can be adapted to the ER task include DiCE [20], LIME-C and SHAP-C [25], which we adopt as baselines. Other interesting counterfactual frameworks that need access to the inner workings of the classifier include [17, 33]. To the best of our knowledge no counterfactual explanation methods specifically designed for the ER setting exist yet.

3 FOUNDATIONS AND PROBLEM STATEMENT

We refer to real-world objects (e.g., products, persons, organizations) as *entities* and to structured entity descriptions as *records*. Given two sets of records, U and V, ER consists of identifying all the record pairs $u, v \in U \times V$ that refer to the same entity. We say that record pairs referring to the same entity are *matching*, and denote as $E^+ \subseteq U \times V$ the set of matching record pairs in the ground truth. Analogously, we refer as $E^- = (U \times V) \setminus E^+$ to the set of non-matching record pairs. We assume that records $u \in U$ have attributes $A_U = \{a_{U_1}, a_{U_2} \dots, a_{U_h}\}$ and, similarly, records $v \in V$ have attributes $A_V = \{a_{V_1}, a_{V_2} \dots, a_{V_k}\}$, therefore U and V may have different schemas. We refer to the value of the i-th attribute of a record $r \in U$ (resp. V) as $r[a_{U_i}]$ (resp. $r[a_{V_i}]$), with $a_{U_i} \in A_U$ (resp. $a_{V_i} \in A_V$).

ER Explanations. We are interested in providing explanations for a model M solving ER as a binary classification problem. We refer as $M(\langle u, v \rangle)$ to the function learned by the model M. Such a function ought to be $\mathbb{T}$ (true) if $(u, v) \in E^+$,



(a) $M(\langle u, v\rangle) = \mathbb{F}, M(\langle w, v\rangle) = \mathbb{T}$

(b) Making the perturbed version of u, denoted u', more similar to w by copying values from w to u triggers $M(\langle u', v\rangle) = \mathbb{T}$.

Fig. 6. Perturbation on a non-matching pair $\langle u, v\rangle$.

and $\mathbb{F}$ (false) otherwise, but can make mistakes if the model is not perfect. The model M can be trained with a subset $T^+ \cup T^-$ of the ground truth $E^+ \cup E^-$ (with $T^+ \subseteq E^+$ and $T^- \subseteq E^-$), or can be unsupervised.

A *local* explanation aims at describing the behavior of M for a single prediction $M(\langle u, v\rangle) = y$. A post-hoc explanation method involves an auxiliary method to explain M after it has been trained. We distinguish two types of post-hoc local explanations, *saliency* explanations and *counterfactual* explanations, as follows:

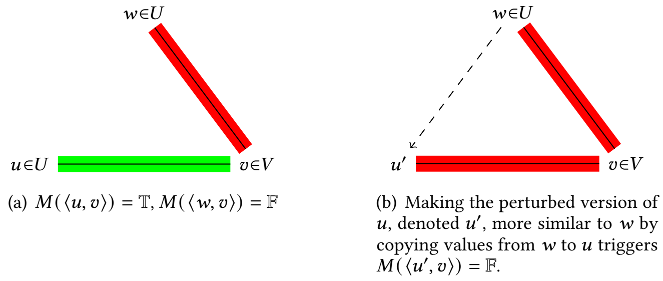
- A *saliency* explanation for ER assigns an importance score to each attribute $a \in A_U \cup A_V$, for a prediction $M(\langle u, v\rangle) = y$. The saliency score aims at capturing the contribution of the attribute to the predicted value.
- A *counterfactual* explanation provides input samples that change a prediction to a desired outcome. Same as for the saliency explanations, we focus on providing attribute based counterfactual explanations. A counterfactual explanation for $M(\langle u, v\rangle) = y$ consists of a pair $\langle u', v'\rangle$, that is equal to $\langle u, v\rangle$ except for one or more attribute values and results in $M(\langle u', v'\rangle) = \overline{y}$.

Similarly to other popular explanation techniques, in order to generate saliency and counterfactual explanations for a prediction $M(\langle u, v\rangle) = y$, we resort to the notion of *perturbation*, which consists of assessing how altering (perturbing) portions an input sample affects the corresponding prediction yielded by the model. In particular, we evaluate the influence that attributes of the input pair have on the prediction by verifying if perturbing their values yields a *flip* in the prediction outcome.

Our approach to generate the perturbations is based on the following intuitions. Consider the prediction $M(\langle u, v\rangle) = \mathbb{F}$, with $u \in U$ and $v \in V$, for which we want to generate an explanation. Let $w \in U$ be a record such that $M(\langle w, v\rangle) = \mathbb{T}$, that is, w and v are a match according to M. As depicted in Figure 6, if we progressively copy attribute values from w to u, deriving a u', increasingly making u' more similar to w based on their content, at some point the prediction of the model will flip, declaring u' and v to be a match. Repeating the same procedure for many records $w \in U$ produces evidence of the influence that attributes and set of attributes have on the input prediction. A similar argument can be formulated for the case of two records u and v that are predicted as a match by the model, i.e., $M(\langle u, v\rangle) = \mathbb{T}$, as depicted in Figure 7. Analogously, we can derive the sets of attributes that if their corresponding attribute values are altered the pair becomes a non-match in a consistent manner.

The above intuitions are formalized by the concepts of *open triangle* and *open triangle perturbations*, which are the building blocks for our probabilistic definition of saliency explanation and counterfactual explanation.

Open triangles. A *left open triangle* for $M(\langle u, v\rangle) = y$ is a triple $t = \langle u, v, w\rangle$ with $w \in U$ and $M(\langle w, v\rangle) = \overline{y}$. In such a *left open triangle*, u v, and w are dubbed the *free* record, *pivot* record, and the *support* record, respectively. Analogously,



(a) $M(\langle u, v \rangle) = \mathbb{T}, M(\langle w, v \rangle) = \mathbb{F}$

(b) Making the perturbed version of u, denoted u', more similar to w by copying values from w to u triggers $M(\langle u', v \rangle) = \mathbb{F}$.

Fig. 7. Perturbation on a matching a pair $\langle u, v \rangle$.

we can define a *right open triangle*, with the support record from the V table. For the sake of simplicity, going forward we mostly refer to *left open triangle* cases. All definitions and methods apply to *right open triangles* analogously.

Open triangle perturbations. Given a left open triangle, we generate a *perturbed copy* u' of the free record u from the support record w by means of a *perturbing record function* $\psi(u, w, A)$, with $A \subseteq A_U$. The perturbing function generates u' by replacing values of all the attributes in A in the free record u with their corresponding values from the support record w, i.e., $u'[a] \longleftarrow w[a], \forall a \in A$.

In right open triangles, where $v \in V$ is the free record and $u \in U$ is the pivot record, we select $w \in V$ and then build perturbed copies of v by replacing values of attributes in A_V.

Perturbed copies are used to compute saliency and counterfactual explanations according to the probabilistic framework developed in [36], which associates the former to the probability of necessity, and the latter to the probability of sufficiency.

3.1 Saliency Explanations

We define the *saliency* of an attribute $a \in A_U$ (resp. in A_V) in the prediction outcome $M(\langle u, v \rangle) = y$ as the probability that changing the value of a in u (resp. v) is a *necessary* factor for flipping the outcome of the prediction.

To compute such a probability, if $a \in A_U$, we rely on a set W of support records for the free node u: $W = \{w | w \in U, M(\langle w,\ v \rangle) = \overline{y}\}$, each record corresponding to a left open triangle $\langle u, v, w \rangle$. Otherwise, if $a \in A_V$, we rely on right open triangles analogously. In the following, for sake of simplicity, we focus on the former case.

Let $\mathcal{U}_{w,a}$ denote the set of perturbed copies of u generated by a support record w by changing all the possible sets of attributes $A \subseteq A_U$ that includes a given attribute a.

$$\mathcal{U}_{w,a} = \{\psi(u, w, A) | A \in \mathcal{P}(A_U), a \in A\}$$

where $\mathcal{P}(A_U)$ is the powerset of A_U. Let $\mathcal{U}_a = \bigcup_{w \in W} \mathcal{U}_{w,a}$.

Example 1. *Consider the records in Figure 1. Suppose we want to produce an explanation of the Ditto prediction* $M(u_1, v_1) = \mathbb{T}$. *A left triangle that uses* u_2 *as a support record (assuming* $M(u_2, v_1) = \mathbb{F}$*) creates* 4 *perturbed copies of* u_1*:*

$$\begin{aligned}\mathcal{U}'_{u_2,Name_{Abt}} = \{&\psi(u_1, u_2, \{Name_{Abt}\}),\\ &\psi(u_1, u_2, \{Name_{Abt}, Description_{Abt}\}),\\ &\psi(u_1, u_2, \{Name_{Abt}, Price_{Abt}\}),\\ &\psi(u_1, u_2, \{Name_{Abt}, Description_{Abt}, Price_{Abt}\})\}\end{aligned}$$

For the sake of simplicity, we show here only 2 *of such perturbed copies (copied values are in boldface):*

- $\psi(u_1, u_2, \{Name_{Abt}\}) =$
  ⟨***"altec lansing inmotion portable audio system ..."***,
  "sony bravia theater blackmicro...", *NaN*⟩
- $\psi(u_1, u_2, \{Name_{Abt}, Description_{Abt}\}) =$
  ⟨***"altec lansing inmotion portable audio system ..."***,
  "altec lansing inmotion ipod portable audio system im600usb...", *NaN*⟩

Saliency score. Given a prediction to explain $M(\langle u, v\rangle) = y$ the *saliency score* of an attribute $a \in A_U$, denoted as ϕ_a, corresponds to the probability that the value of a is changed with values coming from any $w \in W$, conditioned on the fact that $M(\langle w, v\rangle)$ flips the prediction, formally:

$$\phi_a = P(u' \in \mathcal{U}_a | M(\langle u', v\rangle) = \overline{y}) \tag{1}$$

The saliency score for the attributes belonging to the schema of A_U is $\Phi_{A_U} = \{\phi_{a_{U_1}}, \ldots, \phi_{a_{U_h}}\}$. The saliency score for the attributes belonging to A_V (i.e., for the schema of the right attribute v of the input pair of the prediction) are computed accordingly. Finally, a saliency explanation for an ER prediction $M(\langle u, v\rangle) = y$ is composed by the saliency scores for all the attributes in $A_U \cup A_V$, $\Phi = \Phi_{A_U} \cup \Phi_{A_V}$.

3.2 Counterfactual explanations

Counterfactual explanations are associated with the concept of sufficiency. That is, the probability that changing the value of a certain set of attributes is a *sufficient* factor for flipping the outcome of a prediction.

Let $\mathcal{U}_A$ be the set of perturbed copies u' altered by changing all the attributes in $A \subset A_U$, using a set of support records W from left open triangles.

$$\mathcal{U}_A = \{\psi(u, w, A) | w \in W\}$$

The probability of sufficiency that changing a given set of attributes $A \subset A_U$ in the original pair $\langle u, v\rangle$ results in flipping the prediction from y to $\overline{y}$ corresponds to the probability that $M(\langle u, v\rangle)$ is flipped conditioned on the fact that the attributes A have been changed in record u.

$$\chi_A = P(M(\langle u', v\rangle) = \overline{y} | u' \in \mathcal{U}_A) \tag{2}$$

For each A such that $\chi_A > 0$ we can generate a counterfactual explanation as we have at least one $\langle u', v\rangle$ such that $M(\langle u', v\rangle) = \overline{y}$ and $u' = \psi(u, w, A)$ for a given w.

We define a counterfactual explanation for $M(\langle u, v\rangle) = y$ as a pair of records $\langle u', v\rangle$ whose changed attributes $A \subset A_U$ have the highest probability of sufficiency that changing them yields a prediction flip, with A being as small as possible.

$$A^\star = \underset{A}{\operatorname{argmin}}(|\underset{A \subset \mathcal{P}(A_U) \setminus A_U}{\operatorname{argmax}} \chi_A|) \tag{3}$$

Symmetrically we can find counterfactual explanations on the attributes in A_V using right open triangles.

Note that, while providing a counterfactual explanation in terms of a proper example, CERTA also provides a human interpretable measure of the importance of the example. The value $\chi_{A^\star}$ associated with the set of attributes $A^\star$ reveals that by changing all the attributes in $A^\star$ the original predicted outcome flips with a probability of $\chi_{A^\star}$.

3.3 Obtaining triangles

Support records from open triangles are used to change the values of attributes in the free record of a prediction to be explained. Computing the scores ϕ and χ defined in Equations 1 and 2 require calculating how frequently such attribute modifications co-occur with a flipped outcome. Therefore CERTA needs an equal number of left and right open triangles to be generated to explain each prediction.

Left open triangles for a prediction $\langle u, v\rangle$ are obtained by calling the classifier M on all the records $w \in U \setminus \{u\}$ such that $M(\langle w, v\rangle) = \overline{y}$. Symmetrically, right open triangles for a prediction $\langle u, v\rangle$ are obtained by calling the classifier M on all the records $q \in V \setminus \{v\}$ such that $M(\langle u, q\rangle) = \overline{y}$.

In case the number of open triangles generated this way is smaller than expected, CERTA adopts a simple data augmentation scheme to generate more record pairs to evaluate, defined as follows. The value of an attribute a_{U_i} in a record w is a sequence of tokens (strings separated by white space) $w[a_{U_i}] = \{s_1, s_2, ..., s_n\}$. For each record w in U, we generate a new set of records W_w, by changing each possible combination of attributes in w by dropping the first-k or the last-k tokens, with k varying between 1 and $n - 1$.

Intuitively larger numbers of triangles are desirable, in order to more accurately approximate the probability values for necessity and sufficiency. An experimental evaluation of the impact of the number of triangles used to generate explanations is provided in Section 5.5.

4 COMPUTING NECESSITY AND SUFFICIENCY PROBABILITIES

In order to calculate the probability of necessity of an attribute (ϕ_a with $a \in A_U$ or $a \in A_V$), which provides us its saliency score, and the probability of sufficiency of a set of attributes (χ_A with $A \subset A_U$ or $A \subset A_V$), which allows us to obtain a counterfactual explanation, we use a frequentist approach. Namely we count:

- the number of times an attribute is changed with respect to the number of actual flips (eq. 1);
- the number of times changing a set of attributes results in a flip, with respect the number of times that the set of attributes is changed (eq. 2).

Computing the above numbers exactly would require to process multiple open triangles and test all the corresponding perturbed copies (namely $|W| \cdot ((|\mathcal{P}(A_U)| - 2) + (|\mathcal{P}(A_V)| - 2))$ copies)[1] of the free record and, for each of them, computing the prediction. We can, however, be more efficient by *inferring* which attributes result in a flip, as described in the following.

[1] We do not need to compute the empty set and the entire set of attributes A_U and A_V.

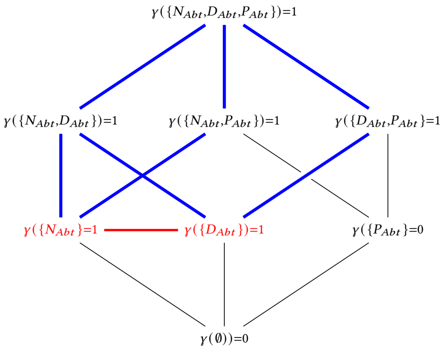

$\gamma(\{N_{Abt},D_{Abt},P_{Abt}\})$=1
$\gamma(\{N_{Abt},D_{Abt}\})$=1
$\gamma(\{N_{Abt},P_{Abt}\})$=1
$\gamma(\{D_{Abt},P_{Abt}\}$=1
$\gamma(\{N_{Abt}\}$=1
$\gamma(\{D_{Abt}\})$=1
$\gamma(\{P_{Abt}\}$=0
$\gamma(\emptyset))$=0

Fig. 8. A lattice structure for a left open triangle on a pair $\langle u_i, v_i \rangle$ from *Abt-Buy* dataset. Nodes are tagged by *flipping* operator $\gamma(\cdot)$. For the sake of readability, we have abbreviated the attribute names with their initials. A minimal flipping antichain $\{\{N_U\}, \{T_U\}\}$ is highlighted in red.

Given a prediction $M(\langle u, v \rangle) = y$, for each left (resp. right) triangle $t = \langle u, v, w \rangle$, with $w \in W$, we build a lattice on the partial order between the elements of the power set $\mathcal{P}(A_U)$ (resp. $\mathcal{P}(A_V)$) and the subset inclusion relation. Figure 8 shows a lattice structure for the power set of the attributes of the *Abt* schema of Figure 1 (for now, ignore the colors of the edges and the $\gamma()$ function).

Then, we tag each node A of the lattice with $\gamma(A)$, where:

$$\gamma(A) = \mathbb{1}(M(\langle u', v \rangle) = \overline{y})$$

with $u' = \psi(u, w, A)$.

Essentially, each node is tagged 1 if copying the values of attributes in A from the support record w into the corresponding attributes of the free record u leads to flipping the original output y, 0 otherwise.

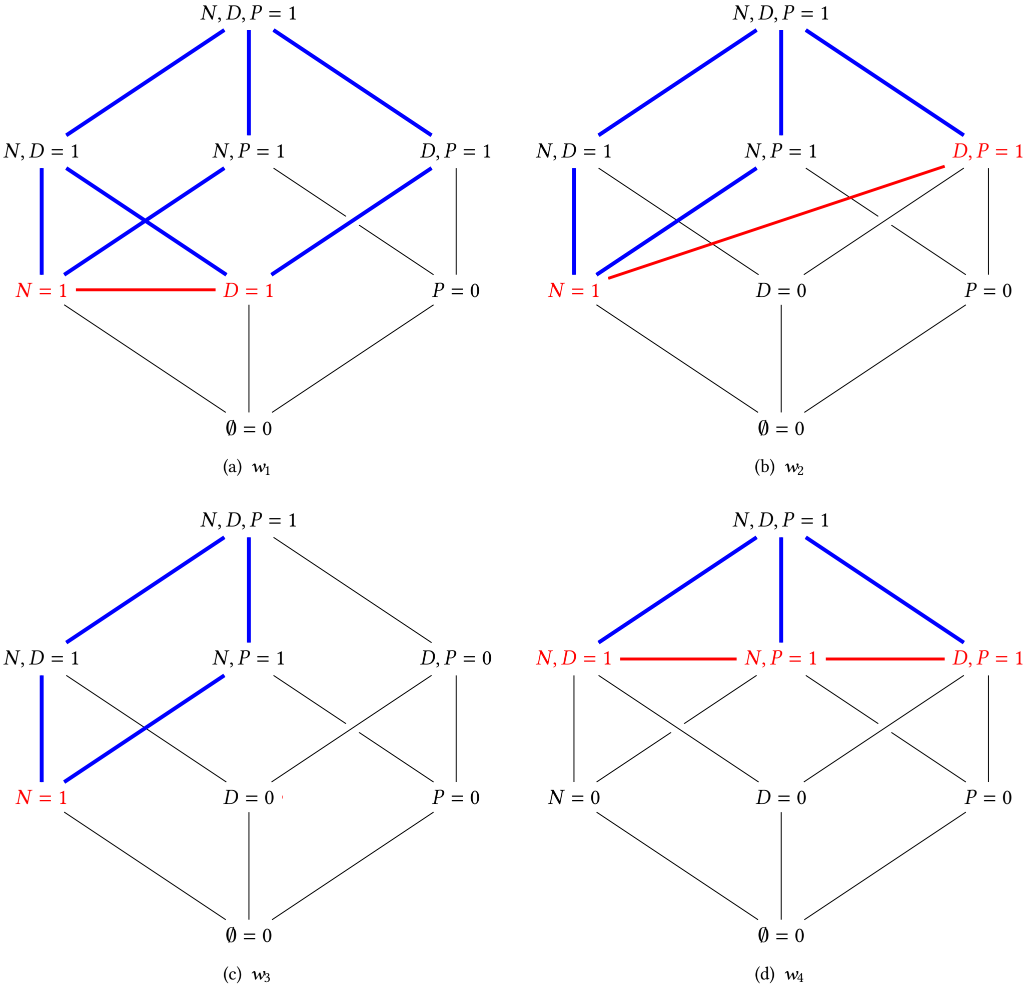
Continuing our example, let us suppose that any subset of $\mathcal{P}(\{Name_{Abt}, Description_{Abt}, Price_{Abt}\})$ except $\{Price_{Abt}\}$ flips the prediction $M(\langle u, v \rangle)$, Figure 8 shows the lattice structure of our running examples with the nodes tagged accordingly.

Inspired by the work in [30], we can make the simplifying assumption that the classifier M is *monotone*: if copying the values of the attributes in A from the support record w to the free record u yields a flipped outcome, then we expect that copying values from a superset $A' \supset A$ the same way will also flip the prediction. Formally, $\gamma(A) = 1 \implies \gamma(A') = 1, \forall A' \supset A$. An empirical evaluation of the veracity of this property is provided in Section 5.6.

Consider Figure 8: assuming M is monotone, if perturbing u copying only $\{Name_{Abt}\}$ flips the prediction, then also all the perturbations built using supersets of $\{Name_{Abt}\}$[2] will flip the predictions, and thus we do not need to compute them.

Given a lattice L, an *antichain* is a set of nodes in L that are not pairwise comparable according to the partial order relations of the lattice. We define the concept of *flipping antichain* as a lattice antichain formed by nodes tagged with 1 (that is, nodes for which the prediction flipped).

[2] Namely: $\{Name_{Abt}, Description_{Abt}\}$, $\{Name_{Abt}, Price_{Abt}\}$, $\{Name_{Abt}, Description_{Abt}, Price_{Abt}\}$.



(a) w_1

(b) w_2

(c) w_3

(d) w_4

Fig. 9. Example lattice structures.

Given a set of flipping antichains Γ, a flipping antichain $\eta \in \Gamma$ is *minimal* (Minimal Flipping Antichain, or MFA in short) if any other flipping antichain in Γ only contains elements that are supersets of elements of η (i.e., any subset of the attribute sets in η do not cause a flip).

For this reason, when the monotone classification property is satisfied, identifying an MFA η saves us from calculating all the predictions corresponding to the perturbations involving supersets of elements in η.

Assuming monotone classification, performing as few predictions as possible on a lattice L corresponds to finding the largest MFAs in L. To this end, we visit the lattice bottom-up with a breadth-first strategy until all the lattice nodes are tagged. For each visited node, we compute the prediction associated to the perturbation corresponding to the attributes of the node. Whenever the prediction flips with respect to the input prediction, we propagate the predicted outcome to all the upward chains leading to the supremum of the lattice.

Example. Consider the pair of records $\langle u_1, v_1 \rangle$ in Figure 1 with M = Ditto and let us focus on explanations for attributes of u_1. As $M(\langle u_1, v_1 \rangle) = \mathbb{T}$, we need to identify records $w \in U$ s.t. $M(\langle w, v_1 \rangle) = \mathbb{F}$. Such records, let them be the fictitious records $W = \{w_1, w_2, w_3, w_4\}$, are used as support records for building four left open triangles $\langle u_1, v_1, w \rangle$, $w \in W$, with u_1 as the free record and v_1 as the pivot. Let the lattices corresponding to the four triangles be those shown in Figure 9. Note that all the triangles are left and thus all the lattices' nodes represent subsets of attributes in A_U. For sake of brevity, we show only each attribute's initial (i.e., N for $Name_{Abt}$, D for $Description_{Abt}$ and P for $Price_{Abt}$) and omit the γ notation. The nodes included in the largest MFA and the edges representing upward paths with flip propagation are highlighted respectively in red and blue.

When processing w_1 (i.e., the open triangle $\langle u_1, v_1, w_1 \rangle$), we get a flip for $\{N\}$ and $\{D\}$ and a non-flip for $\{P\}$. That is, $M(\langle \psi(u_1, w, A), v_1 \rangle) = \mathbb{F}$, for $A = \{N\}$ and $A = \{D\}$, while $M(\langle \psi(u_1, w, A), v_1 \rangle) = \mathbb{T}$ for $A = \{P\}$. Assuming that M is monotone, we can infer the flip/non-flip results for all the upward nodes in the lattice in Figure 9(a) and identify $\{\{N\}, \{D\}\}$ as the largest MFA without further testing.

When processing w_2 and $w3$, we get a flip for $\{N\}$ and a non-flip for the other singleton nodes. In those cases, we can infer only $\{N, D\}$ and $\{N, P\}$ while we need to test $\{D, P\}$ explicitly. That is, we need to collect the result of $M(\langle \psi(u_1, w, \{D, P\}), v_1 \rangle)$. In the case of w_2, the collected result is negative, yielding a flip, and thus we identify $\{\{N\}, \{D, P\}\}$ as the largest MFA (Figure 9(b)). In the case of w_3, the collected result is positive, yielding a non-flip, and thus the largest MFA consists solely of $\{N\}$ (Figure 9(c)).

Finally, when processing w_4, we get all non-flips at the first level, meaning that copying only one attribute from w_4 is not enough for flipping the prediction. In such a case, we need to test all the attribute pairs explicitly, by collecting the result of $M(\langle \psi(u_1, w, A), v_1 \rangle)$, for all $|A| = 2, A \in \mathcal{P}(A_U)$. As shown in Figure 9(d), we get all flips, and thus identify $\{\{N, D\}, \{N, P\}, \{D, P\}\}$ as the largest MFA.

In order to compute explanation scores ϕ and χ as in Equations 1 and 2 respectively, we need to consider all the nodes corresponding to flips, either tested or inferred. Specifically, in Figures 9(a)–9(d) we have a total of 19 flips. As for the saliency explanations, we obtain $\phi_N = \frac{15}{19}$, $\phi_D = \frac{13}{19}$ and $\phi_P = \frac{11}{19}$. As for the counterfactual explanations, we get $\chi_{\{N\}} = \frac{3}{4}$ (4 is the size of W), $\chi_{\{D\}} = \frac{1}{4}$, $\chi_{\{P\}} = 0$, $\chi_{\{N,D\}} = 1$, $\chi_{\{N,P\}} = 1$ and $\chi_{\{D,P\}} = \frac{3}{4}$. Since for this example we have $\max_{A \subset A_U} \chi_A = 1$ and $A^* = \{N, D\}$ or $A^* = \{N, P\}$ (note that A^* cannot be $\{N, D, P\}$ in Equation 3). The resulting counterfactual explanations are all the pairs $\langle u', v_1 \rangle$ such that $u' \in \{\psi(u, w, \{N, D\} | w \in W\} \cup \{\psi(u, w, \{N, P\} | w \in W\}$, as they all yield a flip.

The CERTA algorithm. Overall, the CERTA approach is summarized in Algorithm 1. CERTA keeps counters for sufficiency of sets of attributes (S), necessity of an attribute (N), and number of flips (f). First, it fetches τ open triangles (line 8); the method *get_triangles()* generates $\frac{\tau}{2}$ left open triangles from records $w \in U$ and $\frac{\tau}{2}$ right open triangles using records $q \in V$. Then, for each triangle CERTA builds the corresponding lattice (line 10) and finds the largest minimal flipping antichain (line 11). From the antichain η_{min} it derives all the inputs c that flip the prediction, associated to their corresponding set of changed attributes A (line 12) and updates candidate counterfactuals set C with c (line 13), flip counts for A (line 14) and aggregate flip counts f (line 15). Then, for each attribute $a \in A$ it updates the necessity counts (line 17). CERTA generates saliency scores Φ by dividing the necessity counts by the aggregate flip counts (line 19). For counterfactuals, it generates the sufficiency for attribute sets (line 14) and checks whether it is bigger than current maximum sufficiency (line 24) or equal but involving fewer attributes (line 27). This way the golden set of attributes is identified. Finally, it generates the list of counterfactual explanations whose changed attributes correspond to such a golden set (lines 30-33).

Algorithm 1: The CERTA algorithm.

```
input  : $M(\langle u, v \rangle) = y$, number of triangles $\tau$, $U$, $V$
output : attributes saliency $\Phi$, set of counterfactual examples $E$
procedure certa($u, v, \tau, M, U, V$):
    foreach $A \in \mathcal{P}(A_U) \setminus A_U \cup \mathcal{P}(A_V) \setminus A_V$ do
        $S[A] \leftarrow 0$;
    foreach $a \in A_U \cup A_V$ do
        $N[a] \leftarrow 0$;
    $f = 0$;
    $C \leftarrow \emptyset$;
    $T \leftarrow get\_triangles(M, u, v, y, U, V, \tau)$;
    foreach $t \in T$ do
        $L_t = build\_lattice(t)$;
        $\eta_{min} = get\_lmfa(L_t, M, u, v, y)$;
        foreach $(c, A) \in get\_flipped(\eta_{min})$ do
            $C \leftarrow C \cup \{(c, A)\}$;
            $S[A] \leftarrow S[A] + 1$;
            $f \leftarrow f + 1$;
            foreach $A \in a$ do
                $N[a] \leftarrow 1$;
    foreach $a \in A_U \cup A_V$ do
        $\phi_a \leftarrow \frac{N[a]}{flips}$;
        $\Phi \leftarrow \Phi \cup \{\phi_a\}$;
    $A^\star \leftarrow \emptyset$;
    $\chi^\star \leftarrow 0$;
    foreach $A \in \mathcal{P}(A_U) \setminus A_U \cup \mathcal{P}(A_V) \setminus A_V$ do
        if $\frac{S[A]}{|T|} > \chi^\star)$ then
            $\chi^\star \leftarrow \frac{S[A]}{|T|}$;
            $A^\star \leftarrow A$;
        else if $\frac{S[A]}{|T|} == \chi^\star \textit{and} |A| < |A^\star|$ then
            $\chi^\star \leftarrow \frac{S[A]}{|T|}$;
            $A^\star \leftarrow A$;
    $E \leftarrow \emptyset$;
    foreach $(c, A) \in C$ do
        if $A^\star == A$ then
            $E \leftarrow E \cup \{c\}$;
    return $\Phi, E$;
```

5 EXPERIMENTS

5.1 Experimental setup

We aim to quantitatively measure how explanations generated by CERTA and baselines are effective. Different quantitative measures of effectiveness exist, depending on the specific type of explanation to evaluate (see Section 5.3). We seek not to evaluate plausibility via any user study though, as any possible correlation between plausibility and model performance would increase user performance too and thus invalidate any subsequent result [13].

Dataset	Matches	Attr.s	Records	Values
AB (Abt-Buy)	5743	3	1081 - 1092	776 - 721
AG (Amazon-Google)	1167	3	1363 - 3226	650 - 1511
BA (beerAdvo-RateBeer)	68	4	4345 - 3000	1807 - 1323
DA (DBLP-ACM)	2220	4	2614 - 2292	1209 - 1060
DS (DBLP-Scholar)	5547	4	2614 - 64263	1152 - 32664
FZ (Fodors-Zagats)	110	6	533 - 331	360 - 236
IA (iTunes-Amazon)	132	8	6907 - 55923	903 - 6444
WA (Walmart-Amazon)	962	5	2554 - 22074	1370 - 9504
DDA (Dirty DBLP-ACM)	7418	4	2614 - 2292	938 - 840
DDS (Dirty DBLP-Scholar)	17223	4	2614 - 64263	909 - 25096
DIA (Dirty iTunes-Amazon)	321	8	6907 - 55923	1244 - 6364
DWA (Dirty Walmart-Amazon)	6144	5	2554 - 22074	1001 - 7347

Table 1. Datasets for experimental evaluation.

We perform separate experiments for saliency and counterfactual explanations, considering appropriate baseline methods respectively.

Affected models. We evaluate CERTA using three recent state-of-the-art ER systems based on deep learning (DL), namely:

- the LSTM version of DeepER [10], a DL architecture for ER based on distributed representation of records;
- the Hybrid model in DeepMatcher [21], a DL framework based on distributed representation of attributes
- the DistilBERT [28] based model of Ditto; [16], a DL solution based on the Transformers architecture, with data augmentation and injection of domain knowledge.

Datasets. We use the datasets of the DeepMatcher dataset repository,[3] which have been adopted by the above systems for their experimental evaluation.[4] Table 1 summarizes the main characteristics of each dataset: column "Matches" reports the number of matching pairs of the ground truth; "Records" and "Values" lists the number of records and the number of distinct values in the two sources, respectively. Each dataset comes with its own test and training set, which we use for training the DL models.

5.2 Baseline methods

For conducting quantitative evaluations of the effectiveness of CERTA, we identify two sets of baselines, one of saliency explanations, and one for counterfactual explanations.

Saliency method baselines. We compare the saliency explanations generated by CERTA both with methods that are aware of semantics of the ER task, and with methods that agnostic with respect to the semantics of the classification task. For ER semantics aware saliency explanation methods, we compare against Mojito [8] (which is based on LIME [26]) and LandMark [3]. For Mojito we use the *mojito-drop* technique for explaining *Match* predictions and the *mojito-copy* technique for explaining *Non-Match* predictions, in line with the semantics of the method.

For task agnostic methods, we use SHAP [18] within our evaluation as it is one of the most popular black box explanation methods.

Counterfactual method baselines. Also for the counterfactual explanations, we compare the results generated by CERTA with both semantics aware and semantics agnostic counterfactual methods. As semantics agnostic baseline, we compare against DiCE [20], a black box counterfactual explanation generation method. To the best of our knowledge,

[3] https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md
[4] We have excluded the Company dataset as it has only one attribute.

no ER specific counterfactual framemwork exists yet, therefore we adapt the *LIME-C* and *SHAP-C* counterfactual expanation methods [25] to work within the ER setting, as follows:

- we treat input record pairs as text;
- for LIME-C we adopt *Mojito* instead of plain LIME, to have a better fit with the ER setting.

5.3 Evaluation methodology

We consider different metrics for evaluating different kinds of explanations. Note that for each dataset, all the evaluated metrics are computed on all the examples contained in the corresponding test set. For CERTA we use $\tau = 100$ triangles in all our experiments, unless specified. In Section 5.5 we present experiments that show the robustness of CERTA with respect to this parameter.

For saliency explanations we use the quantitative explanation evaluation metrics of *Faithfulness* and *Confidence indication* [2].

- *Faithfulness* aims to measure whether the most salient attributes are also the most important ones for the predicted output. Faithfulness measures the area under the threshold-performance curve (AUC). Thresholds indicate the fraction of attributes that have to be masked. The attributes to be masked are taken from the saliency explanation, in descending saliency score order. The set of thresholds used is $\{0.1, 0.2, 0.33, 0.5, 0.7, 0.9\}$ and the performance measure is the F1 of the model M. Faithful explanations are expected to induce a higher F1 drop as more salient attributes are incrementally masked. Low AUC values indicate high faithfulness.
- *Confidence indication* aims to measure whether a saliency explanation is a good indicator of the score of the classifier to explain. Confidence indication is calculated as the *mean absolute error* (MAE) of a logistic regression classifier trained with saliency explanation scores for match/nomatch (input) and the actual score of the model (label). A low MAE value indicates that the model's score can be easily identified by looking at the produced explanations.

For both faithfulness and confidence indication, lower values are better.

Quality of counterfactual explanations are evaluated by means of the *Proximity*, *Sparsity* and *Diversity* metrics [20].[5]

- *Proximity* is the mean of attribute-wise distances between a counterfactual example and the original input pair. Proximity for a set of examples is simply the average proximity over all the examples.
- *Sparsity* captures the number of changed attributes between the original input and a generated counterfactual.
- *Diversity* measures attributes-wise distances between each pair of counterfactual examples.

For all the above metrics, i.e., diversity, sparsity and proximity, higher values are better. To conclude the evaluation of counterfactual explanations, we also report the average number of generated counterfactual explanations by each considered method.

5.4 Results

Saliency explanations. In Table 2 we report an evaluation of the faithfulness of the saliency explanations generated using CERTA versus the all identified baselines. For the DeepER model CERTA reports the best faithfulness measure, but for the DS and DDA datasets, where Mojito is the most faithful (CERTA being the second most faithful). For DeepMatcher CERTA reports the best faithfulness measure, but for the DS dataset where SHAP results in being more faithful; there is

[5] Another metric defined in [20] is *Validity*, which measures the fraction of examples returned by a method that are actually counterfactuals, that is, that flip the prediction. However, CERTA produces by construction counterfactual explanation, while DiCE also returns examples that do not. Then, for a fair comparison, we do not report experimental results based on Validity.

	DeepER				DeepMatcher				Ditto			
Dataset	CERTA	LandMark	Mojito	SHAP	CERTA	LandMark	Mojito	SHAP	CERTA	LandMark	Mojito	SHAP
AB	**0.006**	0.12	0.03	21.49	**17.51**	17.56	19.59	18.21	**0.25**	0.31	0.3	0.32
AG	**0.03**	0.13	0.06	0.16	**1.42**	5.17	4.71	**1.42**	**0.31**	0.33	**0.31**	0.35
BA	**0.003**	0.23	0.17	0.21	**8.18**	25.17	27.71	9.13	**0.24**	0.39	0.37	0.36
DA	**0.04**	0.33	0.09	0.17	**20.23**	34.46	35.58	34.99	**0.14**	0.15	**0.14**	0.41
DS	0.42	0.50	**0.32**	0.44	34.9	26.4	52.7	**21.59**	**0.04**	0.10	0.12	0.11
FZ	**0.336**	0.338	0.42	0.34	**4.46**	9.75	4.71	4.71	0.23	0.39	0.41	**0.22**
IA	**0.03**	0.23	0.11	0.16	**25.72**	41.32	46.23	41.08	**0.67**	0.69	0.68	0.68
WA	**0.02**	0.25	0.38	0.09	**10.49**	10.99	38.6	29.53	**0.57**	0.64	0.59	0.59
DDA	0.28	0.52	**0.26**	0.44	**17.51**	29.3	30.97	61.41	**0.34**	0.41	0.41	0.44
DDS	**0.45**	0.48	0.46	0.49	**5.85**	6.12	8.84	8.31	**0.09**	**0.09**	0.12	0.46
DIA	**0.01**	0.17	0.06	0.15	**33.66**	34.21	30.18	30.84	**0.12**	0.23	0.19	0.51
DWA	**0.04**	0.05	0.05	0.23	**11.81**	14.15	17.78	23.5	**0.07**	0.08	0.08	0.09

Table 2. Faithfulness evaluation on saliency explanations.

	DeepER				DeepMatcher				Ditto			
Dataset	CERTA	LandMark	Mojito	SHAP	CERTA	LandMark	Mojito	SHAP	CERTA	LandMark	Mojito	SHAP
AB	**0.021**	0.026	0.025	0.023	**0.016**	0.12	0.096	0.099	0.098	0.121	0.14	**0.045**
AG	0.113	0.15	0.214	**0.098**	**0.015**	0.101	0.048	0.021	0.01	0.01	0.01	0.01
BA	**0.02**	0.05	0.03	**0.02**	0.11	0.126	0.12	**0.10**	**0.298**	0.326	0.474	0.376
DA	**0.182**	0.32	0.221	0.663	**0.002**	0.005	0.003	0.003	**0.104**	0.151	0.126	0.115
DS	**0.213**	0.308	0.292	0.248	0.046	0.049	**0.018**	0.032	**0.046**	0.049	0.054	0.047
FZ	0.488	0.488	**0.396**	1.93	**0.002**	0.103	0.009	0.055	**0.039**	0.223	0.186	0.064
IA	**0.238**	0.342	0.325	0.358	**0.281**	0.364	0.295	0.289	**0.071**	0.094	0.129	0.13
WA	**0.041**	0.081	0.089	0.053	**0.021**	0.231	0.044	0.035	**0.015**	0.08	0.051	0.046
DDA	**0.375**	0.247	0.356	0.252	**0.204**	0.276	0.407	0.294	0.325	0.149	0.424	**0.07**
DDS	**0.102**	0.144	0.171	0.14	**0.08**	0.09	0.09	**0.08**	**0.062**	0.133	0.128	0.127
DIA	0.225	**0.198**	0.23	0.233	**0.047**	0.054	0.05	0.08	**0.047**	0.054	0.05	**0.047**
DWA	**0.131**	0.225	0.207	0.145	**0.251**	0.269	0.272	0.263	**0.204**	0.276	0.407	0.294

Table 3. Confidence Indication evaluation on saliency explanations.

also a tie between SHAP and CERTA on the AG dataset. For the Ditto model CERTA is the most faithful in almost all the cases, SHAP has a slightly better faithfulness measure for the FZ dataset; there are also two ties between CERTA and Mojito (DA and AG) and one between CERTA and LandMark (DDS).

In Table 3 we report an evaluation of the confidence indication of the saliency explanations generated using CERTA versus the all identified baselines. CERTA is the most indicative of the confidence of DeepER for most of the datasets, SHAP wins on the AG dataset, Mojito wins on the FZ dataset while LandMark wins on the DIA dataset. CERTA is the most indicative of the confidence of DeepMatcher for most of the datasets, two exceptions relate to BA dateset (SHAP wins) and DS (Mojito wins). Finally, CERTA is the most indiciative of the confidence for Ditto on most of the datasets, SHAP performs better for AB and DDA datasets and ties on DIA.

	DeepER				DeepMatcher				Ditto			
Dataset	CERTA	DiCE	SHAP-C	LIME-C	CERTA	DiCE	SHAP-C	LIME-C	CERTA	DiCE	SHAP-C	LIME-C
AB	**0.74**	0.72	0.35	0.42	**0.56**	0.55	0.51	0.48	**0.55**	0.52	0.52	0.28
AG	**0.51**	0.49	0.33	0.31	0.66	**0.72**	0.62	0.66	**0.94**	0.51	0.38	0.49
BA	0.37	**0.59**	0.35	0.41	0.3	**0.53**	0.18	0.28	**0.37**	0.22	0.2	0.35
DA	**0.49**	0.44	0.18	0.41	**0.58**	**0.58**	0.41	0.52	**0.58**	0.49	0.48	0.38
DS	**0.63**	0.6	0.38	0.55	0.55	0.55	**0.62**	0.52	**0.39**	0.32	0.36	0.32
FZ	**0.52**	0.41	0.39	0.48	**0.63**	0.49	0.53	0.48	**0.92**	0.48	0.74	0.81
IA	0.59	**0.67**	0.21	0.55	**0.52**	0.25	0.36	0.43	0.14	0.09	0.04	**0.34**
WA	0.41	**0.61**	0.39	0.4	0.35	0.3	**0.39**	**0.39**	**0.49**	0.35	0.31	0.15
DDA	**0.67**	0.66	0.57	0.59	**0.58**	0.55	0.44	0.55	**0.59**	0.4	0.25	0.39
DDS	**0.45**	0.41	0.25	0.39	**0.59**	**0.59**	0.58	0.39	**0.34**	0.41	0.41	0.44
DIA	**0.49**	0.38	0.39	0.35	0.67	**0.72**	0.62	0.67	**0.66**	0.49	0.4	0.55
DWA	**0.52**	0.51	0.38	0.49	**0.76**	0.72	**0.76**	0.62	**0.68**	0.59	0.51	0.39

Table 4. Proximity evaluation on counterfactual explanations.

	DeepER				DeepMatcher				Ditto			
Dataset	CERTA	DiCE	SHAP-C	LIME-C	CERTA	DiCE	SHAP-C	LIME-C	CERTA	DiCE	SHAP-C	LIME-C
AB	**0.9**	0.81	0.89	0.87	**0.91**	0.82	0.85	0.86	**0.93**	0.87	0.85	0.1
AG	0.88	0.87	0.9	**0.91**	**0.94**	0.92	0.93	0.92	**0.88**	0.87	0.78	0.63
BA	**0.89**	0.83	**0.89**	0.78	**0.96**	0.89	0.95	0.93	**0.96**	0.95	0.24	0.92
DA	**0.96**	0.81	0.95	0.88	**0.94**	0.88	0.91	0.9	**0.92**	0.89	0.71	0.7
DS	**0.91**	0.81	0.89	0.89	**0.98**	0.93	0.92	0.89	**0.91**	**0.91**	0.64	0.65
FZ	**0.92**	0.91	0.83	0.88	**0.93**	**0.93**	0.77	0.92	0.91	0.75	0.89	**0.93**
IA	**0.93**	0.92	0.84	0.9	**0.99**	0.97	0.96	0.95	**0.99**	**0.99**	**0.99**	0.96
WA	0.89	0.83	**0.94**	0.91	**0.92**	0.89	0.89	0.81	**0.96**	0.94	0.9	0.74
DDA	**0.91**	0.85	0.87	0.84	**0.94**	0.78	**0.94**	0.93	**0.95**	0.93	0.84	0.72
DDS	0.9	0.87	0.89	**0.91**	**0.95**	0.85	0.94	0.91	**0.98**	0.88	0.72	0.81
DIA	0.89	0.78	**0.93**	**0.93**	**0.94**	0.92	0.92	0.92	**0.91**	0.86	0.69	0.71
DWA	**0.92**	0.9	**0.92**	0.91	**0.93**	0.9	0.86	0.88	**0.97**	0.95	0.76	0.88

Table 5. Sparsity evaluation on counterfactual explanations.

	DeepER				DeepMatcher				Ditto			
Dataset	CERTA	DiCE	SHAP-C	LIME-C	CERTA	DiCE	SHAP-C	LIME-C	CERTA	DiCE	SHAP-C	LIME-C
AB	**0.54**	0.45	0	0	**0.61**	0.44	0.1	0.1	**0.53**	0.3	0.17	0.04
AG	0.41	**0.51**	0.1	0.1	0.54	**0.64**	0.1	0.1	**0.46**	0.29	0.14	0.1
BA	0.38	**0.49**	0.12	0	0.31	**0.52**	0	0.01	**0.37**	0.22	0.1	0.12
DA	**0.33**	0.05	0	0	**0.65**	0.51	0	0	0.43	**0.44**	0.03	0.05
DS	0.39	**0.41**	0.05	0	**0.67**	0.53	0	0	**0.31**	0.29	0.01	0.05
FZ	**0.35**	0.31	0	0	0.45	**0.55**	0	0	0.34	**0.38**	0.1	0.13
IA	**0.31**	0.29	0	0	**0.8**	0.29	0.24	0	0.12	0.04	0.13	**0.14**
WA	**0.39**	0.38	0	0	**0.56**	0.5	0	0	0.38	**0.41**	0.05	0.01
DDA	**0.38**	0.36	0.04	0	**0.49**	**0.49**	0.1	0	**0.39**	0.28	0.01	0.04
DDS	**0.39**	0.31	0	0	**0.63**	0.55	0.11	0	**0.35**	0.23	0.08	0.09
DIA	**0.41**	0.35	0	0	0.54	**0.65**	0.05	0	**0.46**	0.19	0.12	0.19
DWA	**0.39**	0.34	0	0.01	0.48	**0.56**	0.01	0	0.37	**0.45**	0.15	0.03

Table 6. Diversity evaluation on counterfactual explanations.

Counterfactual explanations. In Table 4 we report the evaluation of CERTA and baselines for the proximity metric. For the the DeepER model CERTA reports better proximity values in 9 out of 12 datasets, in the 3 remaining cases DiCE reports the best proximity value. In the case of the DeepMatcher model there's a slightly less clear winner, CERTA and DiCE reach the best proximity on almost the same number of datasets (6 for CERTA, 4 for DiCE) while they reach a tie on one dataset. On the WA dataset SHAP-C and LIME-C reach the highest proximity, whereas SHAP-C wins on the DS dataset. Finally, CERTA reports best proximity on all but one datasets for the Ditto classifier, where LIME-C reaches a higher proximity for the iTunes-Amazon dataset.

In Table 5 we report the evaluation of CERTA and baselines for the sparsity metric. For the DeepER case, CERTA reports the best sparsity on 6 datasets out of 12, a tie is reached between CERTA and SHAP-C on the BA and DWA datasets. LIME-C reaches the highest sparsity on the AG and DDS datasets. CERTA reaches the highest sparsity measure on all datasets, when adopting the DeepMatcher classifier. There are still a couple of ties with DiCE (FZ dataset) and SHAP-C (DDA). For the Ditto classifier, CERTA achieves the highest sparsity on 8 out of 12 datasets, a tie is reached on the iTunes-Amazon, involving both DiCE and SHAP-C. Another tie involves CERTA and DiCE for the DS dataset.

In Table 6 we report the evaluation of CERTA and baselines for the diversity metric. Across all datasets and models, CERTA and DiCE reach the best diversity measure, except for the IA case with the Ditto classifier. For DeepER CERTA gets the highest diversity on 9 out of 12 datasets, DiCE instead provides more diverse counterfactual explanations for BA, AG and DS datasets. On the DeepMatcher classifier CERTA gets the highest diversity for 6 datasets, DiCE does the same on 5 datasets, while they obtain a tie on the remaining dataset (DDA). Finally, in Figure 10 we report the average number of counterfactual explanations generated by CERTA and baselines for the three considered classifiers. CERTA is capable of generating more counterfactual explanations for all the models. Note also that SHAP-C and LIME-C are

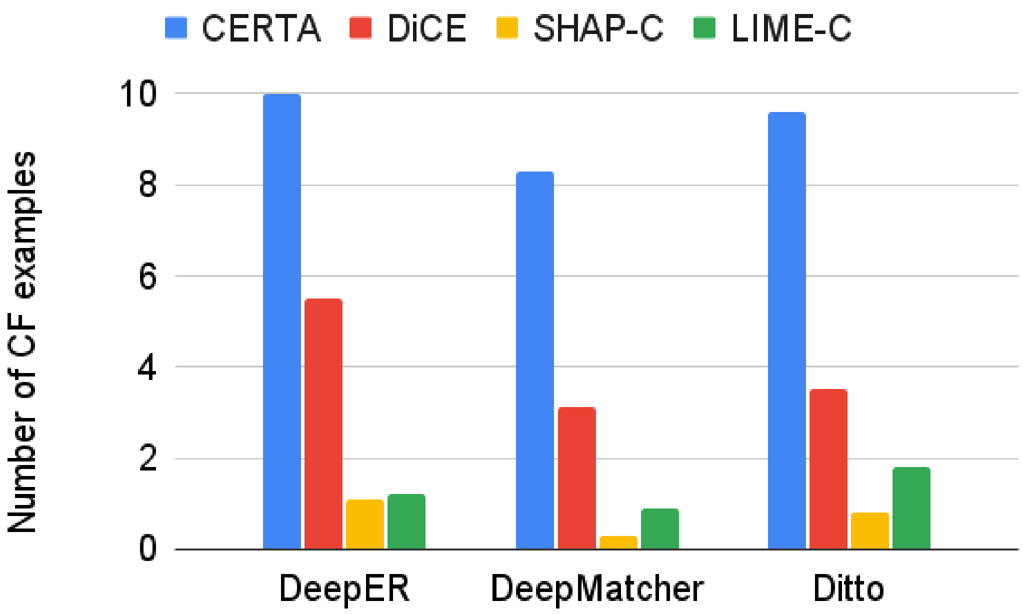


Fig. 10. Average number of CF examples generated by CF methods across all considered classifiers and datasets.

sometimes not able to generate even a single explanation, as a result the mean number of explanations is below 1 with SHAP-C for both DeepER and DeepMatcher.

5.5 Impact of number of triangles

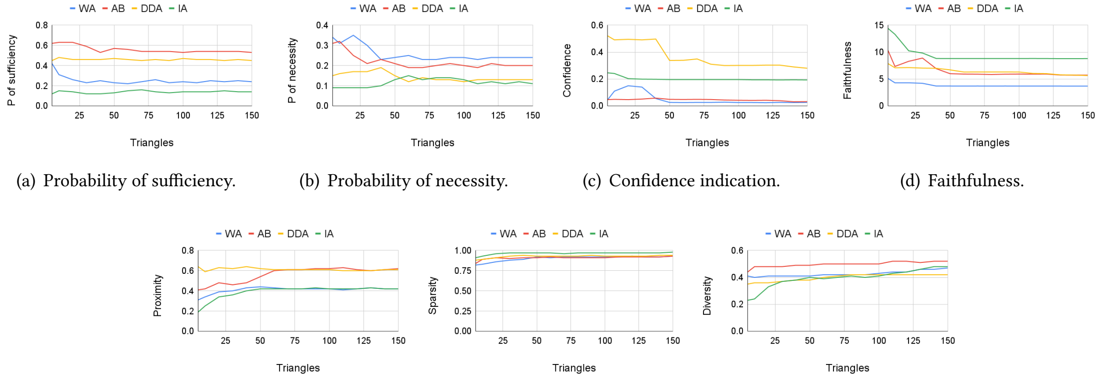
CERTA relies on the use of open triangles in order to identify different ways to perturb the records in the original prediction to explain and calculate the probability of sufficiency and necessity associated to the changed attributes.

In this section we study the impact of the number of open triangles adopted in CERTA along different perspectives. We report how the number of open triangles influences:

- the average probability of sufficiency of a set of attributes in Figure 11(a);
- the average probability of necessity of an attribute in Figure 11(b);
- the quantitative metrics reported in Section 5.4 for saliency explanations in Figure 11(c) and Figure 11(d);
- the quantitative metrics reported in Section 5.4 for counterfactual explanations in in Figure 11(e), Figure 11(f) and Figure 11(g).

The evaluations are performed on all three classifiers (DeepER, DeepMatcher and Ditto) on four different datasets (WA, AB, DDA, IA). The results show the average of the reported measure across the three classifiers, for each dataset.

Each of the reported measures in this study tends to converge as the number of triangles used increases. More specifically, we observe that when CERTA uses more than 75-80 triangles, it has a generally stable behavior on all the reported metrics. The only metric that increases steadily with the number of triangles is the *Diversity* measure on the Dirty DBLP-ACM and iTunes-Amazon datasets.



(a) Probability of sufficiency.

(b) Probability of necessity.

(c) Confidence indication.

(d) Faithfulness.

(e) Proximity.

(f) Sparsity.

(g) Diversity.

Fig. 11. Probability of Sufficiency (a), Probability of Necessity (b), Confidence Indication (c), Faithfulness (d), Proximity (e), Sparsity (f), Diversity (g) averages, as number of triangles increases.

5.6 Evaluation of monotonicity assumption

In Section 4 we described how CERTA builds on the monotone classifier assumption from [30] in order to perform as few predictions as possible while tagging the nodes of the lattice structures. With such an assumption, a flip for a set of attributes A is expected to be propagated in any superset of $A' \supset A$. As soon as CERTA finds a flip $\overline{y}$ for a given A, it stops exploring all the upward nodes $A' \supset A$, hence the outcomes for any such A' are assumed to be $\overline{y}$ without being computed.

Assuming a flip for a set of attributes A induces a flip in any superset $A' \supset A$ might overestimate both the probability of sufficiency of A' and the probability of necessity of all attributes $a_i \in A$, in case any such predictions for A' doesn't result in an actual flip. This might happen if, following the example in Figure 8, perturbing the value of the attribute $Name_{Abt}$ in the lattice results in a flip, while perturbing the values of the attributes $Name_{Abt}$ and $Price_{Abt}$ doesn't result in a flipped outcome (whereas in Figure 8, where the monotonicity assumption holds, the prediction flips in both cases).

We conduct an experiment to quantify, for a given lattice, how many predictions we save on average, as compared with the number of mistakes we do by assuming monotone classification.

To do so we run CERTA with and without such an optimization and compare the actual outcomes for all the predictions with the case where predictions are propagated based on monotonicity. We report, for a given lattice:

- the number l of attributes associated to the lattice (*Attributes*);
- the number of predictions CERTA needs to make without computing probabilities exactly (*Expected*, equals to $2^l - 2$);
- the number of predictions performed when CERTA assumes monotone classification (*Performed*);
- the number of predictions saved by CERTA when assuming monotone classification ($Saved = Expected - Performed$);
- the ratio between the number of predictions whose *monotone* outcome is different from the actual outcome and the number of saved predictions (*Error rate*).

In Table 7 we report the average number of such measures for a given lattice with all classifiers mentioned in Section 5.1, on four different datasets.

Dataset	Attributes	Expected	Performed	Saved	Error rate
AB	3	6	3.03	2.97	0.01
BA	3	6	2.93	3.07	0.04
WA	4	14	6.04	7.96	0.03
DDS	4	14	4.68	9.32	0.04
IA	9	510	45.19	464.81	0.04

Table 7. Average number of expected, performed, saved and wrong predictions on a single lattice.

Our comparison reveals the monotone classification assumption allows CERTA to save $\sim 50\%$ of the predictions for small sets of attributes (AB, BA datasets), with a relatively small error rate, between $1-4\%$. With slightly bigger sets of attributes (WA, DDA datasets) CERTA saves between $57-64\%$ of the predictions, with an error rate between $3-4\%$. The best gain is seen with bigger sets of attributes (IA dataset), where CERTA saves $\sim 91\%$ of the predictions with an error rate of $\sim 4\%$. From our empirical evaluation, the monotone classifier assumption provides an overestimation of the probabilities of at most 4%, which seems a reasonable tradeoff especially for bigger sets of attributes, where this allows CERTA to only perform $\sim 9\%$ of the requested predictions.

6 CONCLUDING REMARKS AND FUTURE WORK

In this paper, we introduced the novel CERTA method for computing saliency and counterfactual explanations for the Entity Resolution (ER) task. Our key insights are the following. (i) Given a pair of records $\langle u, v\rangle$, we identify records w that can form *open triangles*, that is, records from which we can progressively copy values so as to make $\langle u, v\rangle$ less likely to match when initially declared as a match, or more similar when initially declared as a non-match. (ii) Given a triangle (u, v, w), we leverage *lattice data structures* to identify minimal changes to attribute values that can yield a flip in prediction, with few targeted attempts.

Our experimental comparison with baseline solutions demonstrated that CERTA can find saliency and counterfactual explanations that are more effective on existing deep learning based classifiers, according to established quantitative evaluation metrics.

Future work includes application of CERTA to other scenarios where the goal is to learn how similar or related two objects are, and thus can display a transitive structure analogous to ER. Examples of such scenarios include schema matching, recommendation systems and handwriting verification. In addition to that, extension of CERTA's principled explanation framework for ER to token-level explanations is another line of future research.

REFERENCES

[1] V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović, et al. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*, 2019.

[2] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. A diagnostic study of explainability techniques for text classification. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3256–3274. Association for Computational Linguistics, 2020.

[3] A. Baraldi, F. D. Buono, M. Paganelli, and F. Guerra. Using landmarks for explaining entity matching models. In Y. Velegrakis, D. Zeinalipour-Yazti, P. K. Chrysanthis, and F. Guerra, editors, *Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021*, pages 451–456. OpenProceedings.org, 2021.

[4] N. Barlaug and J. A. Gulla. Neural networks for entity matching: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(3):1–37, 2021.

[5] U. Brunner and K. Stockinger. Entity matching with transformer architectures-a step forward in data integration. In *International Conference on Extending Database Technology, Copenhagen, 30 March-2 April 2020*, 2020.

[6] P. Christen. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 151–159, 2008.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] V. Di Cicco, D. Firmani, N. Koudas, P. Merialdo, and D. Srivastava. Interpreting deep learning models for entity resolution: an experience report using lime. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, pages 1–4, 2019.

[9] A. Ebaid, S. Thirumuruganathan, W. G. Aref, A. K. Elmagarmid, and M. Ouzzani. EXPLAINER: entity resolution explanations. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 2000–2003. IEEE, 2019.

[10] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang. Distributed representations of tuples for entity resolution. *PVLDB*, 11(11):1454–1467, 2018.

[11] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[12] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[13] A. Jacovi and Y. Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

[14] R. Kommiya Mothilal, D. Mahajan, C. Tan, and A. Sharma. Towards unifying feature attribution and counterfactual explanations: Different means to the same end. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 652–663, 2021.

[15] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[16] Y. Li, J. Li, Y. Suhara, A. Doan, and W. Tan. Deep entity matching with pre-trained language models. *Proc. VLDB Endow.*, 14(1):50–60, 2020.

[17] A. V. Looveren and J. Klaise. Interpretable counterfactual explanations guided by prototypes. In N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, and J. A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part II*, volume 12976 of *Lecture Notes in Computer Science*, pages 650–665. Springer, 2021.

[18] S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017.

[19] D. Martens and F. Provost. Explaining data-driven document classifications. *MIS quarterly*, 38(1):73–100, 2014.

[20] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In M. Hildebrandt, C. Castillo, L. E. Celis, S. Ruggieri, L. Taylor, and G. Zanfir-Fortuna, editors, *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, pages 607–617. ACM, 2020.

[21] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34, 2018.

[22] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[23] A. Primpeli and C. Bizer. Profiling entity matching benchmark tasks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3101–3108, 2020.

[24] K. Qian, L. Popa, and P. Sen. Systemer: A human-in-the-loop system for explainable entity resolution. *Proc. VLDB Endow.*, 12(12):1794–1797, 2019.

[25] Y. Ramon, D. Martens, F. J. Provost, and T. Evgeniou. A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: Sedc, LIME-C and SHAP-C. *Adv. Data Anal. Classif.*, 14(4):801–819, 2020.

[26] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[28] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[29] I. Stepin, J. M. Alonso, A. Catalá, and M. Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001, 2021.

[30] Y. Tao. Entity matching with active monotone classification. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 49–62, 2018.

[31] S. Thirumuruganathan, M. Ouzzani, and N. Tang. Explaining entity resolution predictions: Where are we and what needs to be done? In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pages 1–6, 2019.

[32] S. Verma, J. P. Dickerson, and K. Hines. Counterfactual explanations for machine learning: A review. *CoRR*, abs/2010.10596, 2020.

[33] S. Wachter, B. D. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399, 2017.

[34] W. Wang, M. Zhang, G. Chen, H. Jagadish, B. C. Ooi, and K.-L. Tan. Database meets deep learning: Challenges and opportunities. *ACM SIGMOD Record*, 45(2):17–22, 2016.

[35] X. Wang, L. Haas, and A. Meliou. Explaining data integration. *Data Engineering Bulletin*, 41(2), 2018.

[36] D. Watson, L. Gultchin, A. Taly, and L. Floridi. Local explanations via necessity and sufficiency: unifying theory and practice. *arXiv preprint arXiv:2103.14651*, 2021.