

Automatic Entity Labeling through Explanation Techniques

(Discussion Paper)

Silvana Castano¹, Alfio Ferrara¹, Donatella Firmani², Jerin George Mathew³ and Stefano Montanelli¹

¹Università degli Studi di Milano

²Università degli Studi Roma Tre

³Università di Roma Sapienza

Abstract

Entity resolution (ER) aims at matching records that refer to the same real-world entity, e.g., the same product sold by different websites. Recent solutions to this problem have reached unprecedented accuracy. Nonetheless, due to intrinsic limitations of automatic testing methods, it is known among researchers and practitioners that a significant manual effort is still required in production environments for verification and cleaning of ER results. In order to facilitate such activity, we are developing the E2L methodology (**Entity to Labels**) for automatic computation of human-readable labels of identified entities. Given a selection of entities for which the user wants to compute labels, E2L first extracts relevant features by training a classifier on the ER results, then it leverages the notion of *black-box model explanation* to select the most important terms for the classifier, and finally it uses those terms to compute labels. In this paper we report our first experiences with E2L. Preliminary results on a real-world application scenario show that E2L labels can provide an accurate description of entities and a natural way for humans to assess the trustworthiness of ER results at a glance.

1. Introduction


Entity Resolution (ER) is the task of finding records in a collection that refer to the same real-world entity. Recent works have investigated the application of machine learning (ML) and deep learning (DL) techniques, demonstrating impressive prediction accuracy [1]. Nonetheless, in production environments, humans are still required to manually inspect the entities identified by the ER process, in order to assess their trustworthiness. This can be a gruesome activity, especially when large datasets are considered, with entities consisting of hundreds of records. For this reason, tools for supporting the manual inspection of ER results and speeding up the search for possibly mismatched records are strongly demanded. Within this space, we focus on the problem of computing human-readable textual *labels* of identified entities, such as those in Table 1, which represent a natural way to support human comprehension of what is inside each clustered entity.

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy

✉ silvana.castano@unimi.it (S. Castano); alfio.ferrara@unimi.it (A. Ferrara); donatella.firmani@uniroma3.it (D. Firmani); mathew@diag.uniroma1.it (J. G. Mathew); stefano.montanelli@unimi.it (S. Montanelli)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

ID	Description	Resolution	Battery
1	Cannon EOS 1100D - Buy	32 mp	NULL
1	Camera Canon EOS 1100D	32000000 p	NP-400
1
2	Sony Ebay 4.7 stars	A7	NULL
2

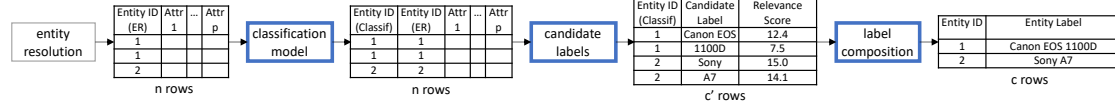
(a)

ID	Entity Label
1	Canon EOS 1100D
2	Sony A7

(b)

Table 1

(a) Sample records from the dataset in our experiments. (b) Labels returned by E2L. The column entity “ID” shows the result of the ER process.

**Figure 1:** The E2L approach. The E2L components are shown in blue.

Already available solutions for analogous tasks (see Section 4) typically require some form of human intervention, such as, providing external knowledge (e.g., vocabularies) or a selection of sample labels for training. Fully-automated solutions instead are based on token frequencies (e.g., TF-IDF) which may perform poorly in datasets with skewed entity size distribution. Our main intuition is to exploit (i) recent methods to process natural language such as [2, 3] to discover meaningful patterns in the association between records and entities with no human effort, and (ii) recent *explainable techniques* such as [4] to reveal such patterns and make them human-readable, by selecting the salient information.

In this paper, we formalize these intuitions by presenting the E2L (**Entity to Labels**) approach and report our experiences with a real world application scenario [5] where ER results need to be manually curated. Our current implementation, featuring two representative text classification methods [3, 2] and one popular explanation method [4] can achieve promising results and highlight errors in the ER results. A repository with all our data and scripts is publicly available for download at <https://github.com/jermathew/E2L>.

2. The E2L methodology

Let $R = \{r_1, \dots, r_n\}$ be a collection of *record descriptions* referring to a set of entities $\hat{E} = \{e_1, \dots, e_c\}$ with $n > c$. Each record $r \in R$ is related to an entity $e \in \hat{E}$, also referred as a *cluster* of records. Given two records $r_1 \in R, r_2 \in R$, we refer to them as *matching* records if they are associated to the same entity.

Our methodology, that we call E2L (**Entity to Labels**), comprises the sequence of modules in Figure 1 as described below.

1. Classification model. Given a set of entities $E \subseteq \hat{E}$ that ought to be manually checked by the final user, E2L trains a classifier to learn a function $M : R \rightarrow E$, such that $M(r) = e$ denotes that the record $r \in R$ is associated with the entity e . In order to build the training set, we use

standard text processing (e.g., stop-words removal) and *tokenization* techniques to represent a record $r \in R$ as sequence of tokens $T(r) = [t_1, \dots, t_s]$. Resulting tokens can be either single terms (e.g. Canon) or *noun chunks*. A noun chunk provides a singleton representation of a composite noun (e.g., digital camera, USA warranty). Note that this step requires no human effort as association between records and entities required for training are selected directly from the input ER results.

2. Candidate labels. Each element $l \in \bigcup_{r \in R \text{ s.t. } M(r)=e} T(r)$ represents a *candidate label* for the entity e . Given an entity e , this module computes a real-valued *relevance score* ρ_l for each candidate label l by leveraging a *black-box explanation technique* over model M . Intuitively, consider a token $t \in T(r)$ and let \hat{r} correspond to the record r without t . The candidate labels module assigns higher relevance score to tokens that yields more consistently $M(\hat{r}) \neq M(r)$, for all r s.t. $M(r) = e$. Specifically, we use LIME [4] as our black-box explanation technique. In order to compute relevance scores for a given record r and a model M , LIME creates a new set of records R_p by randomly removing tokens from r . In R_p , records are represented as binary vectors where each dimension corresponds to a different token. Then, given a class $e \in E$, each $r' \in R_p$ is labeled accordingly to whether $M(r') = e$ or not. Finally, LIME fits a linear model on R_p . Weights of the linear model represent how much each token contributed to $M(r)$. Given an entity e , the output of this module is a sorted list of candidate labels and associated relevance scores $L^e = [(l_1, \rho_{l_1}), (l_2, \rho_{l_2}), \dots], \rho_{l_i} \geq \rho_{l_{i+1}}$.

We now describe the candidate labels module in more details. Let $R_e \subseteq R$ be the set of records that M associates to a given entity e , that is $R_e = \{r \in R \mid M(r) = e\}$. For each record $r \in R_e$ we submit its tokens $T(r)$ to the black-box explanation function in order to get their relevance scores $L_r^e = \{ \langle t, \rho \rangle : t \in T(r), \rho \in \mathbb{R} \}$. Tokens with positive relevance are then sorted by non-increasing relevance value and selected until their cumulative relevance is greater or equal to a user-specified fraction $\beta \in [0, 1]$ of the total. As a result a selection $L_r^{e'} \subseteq L_r^e$ of tokens is obtained for the record r . The set of candidate labels L^e consists of the union of the selected tokens $L_r^{e'}$ for each $r \in R_e$, and, for each label t , the label relevance $L^e[t]$ is the sum of the relevance scores of $L_r^{e'}[t]$ over all the records in $r \in R_e$. We repeat these steps for each entity $e \in E$

Running the aforementioned steps can be infeasible if (i) R_e contains a massive number of records or (ii) records in R_e consist of thousands of tokens. In both cases, the black-box explanation function could take a significant amount of time to process R_e . In order to address both points, we include in E2L a record sampling step and a token sampling step – described below – to be optionally executed before the black-box model explanation computation.

(i) During the record sampling step, we aim at picking a subset $R_e' \subseteq R_e$ such that the tokens in R_e' cover most of the relevant tokens in R_e . In order to do so, we run the *k-means* clustering algorithm with parameter k_r on a vector representation¹ of the input records R_e and then, for each cluster, we select the closest record to its centroid based on ℓ^2 -norm. As a result, we obtain k_r vectors from which we retrieve the corresponding records, which collectively make up R_e' . The value k_r , corresponding to the sample size, is set such that as the number of records $|R_e|$ grows, the fraction of sampled records decreases via linear interpolation.

¹The selected vector representation can be arbitrarily chosen, e.g a tf-idf vector representing a record r or the mean word embedding of its constituent tokens

(ii) During the token sampling step, given a record $r \in R_e$ we aim at picking a selection $T'(r) \subseteq T(r)$ of its most representative tokens. To that end, given a record $r \in R_e$ we sort its tokens $T(r)$ based on their Term Frequency (TF) in decreasing order, prioritizing noun chunks over singleton text tokens. Afterwards, we select the top k_t tokens as those to be included in the sample for the record r . Analogously to the record sampling step, the value k_t is set via linear interpolation so that as the number of tokens in $T(r)$ grows, the fraction of selected tokens decreases.

3. Label composition. Candidate labels and associated relevance scores are finally processed to return to the user a label for each entity. Given a user parameter k , we return as label the composition (e.g concatenation) of the top k labels in L^e .

3. Experiences with E2L

The E2L approach is evaluated on the camera dataset in the Alaska Benchmark, an end-to-end benchmark tailored for a variety of tasks related to Data Integration, including ER [5], and has been recently used for the 2020 SIGMOD Programming Contest² and for the two editions of the DI2KG challenge³. The dataset comprises i) a set of camera descriptions collected over different web sources, and ii) a manually-curated ground truth consisting of camera names (i.e., brand name and model name) for each description, such that multiple descriptions can refer to the same camera. In the evaluation, we take into account the 20 entities with the highest number of records, ranging from 184 to 53 records per entity. The resulting dataset consists of 2171 records. We use the `page_title` attribute from each description to compose a dataset (hereinafter called Alaska dataset) as a list of `<page_title>`, `<model_name>` pairs, where `<model_name>` represents the correct label expected for each group of descriptions referring to the same camera. The longest `page_title` field in the dataset contains 42 words, while the shortest one contains 3 words.

Our experiments were performed on a server environment using an Intel Xeon E5-2966 v4 CPU, 512 GB of RAM, and 4 NVIDIA Tesla P100-SXM2 GPUs. The operating system is Ubuntu 17.10.

Classification model. We exploited two models, a LSTM-based neural network [6] and a pretrained DistilBERT model [2], and we generated two versions of E2L, namely E2L-Bert and E2L-Glove. The LSTM-based network consists of a pre-trained embedding layer based on GloVe [3] followed by a bidirectional LSTM (Bi-LSTM) layer whose memory dimension is 100. The output of the last time step in the Bi-LSTM is then fed to a fully connected layer of size 64 using ReLu as the activation function. Finally, the resulting output is passed to a fully connected layer of size 20 where softmax is used as the activation function. As for the second model, we leveraged the Transformers library⁴ to set up a pretrained DistilBERT model for a multiclass classification task. This model comprises two parts: the body, consisting in a pretrained DistilBERT model, and a classification head on top of the body whose last layer consists in a fully connected layer

²<http://www.inf.uniroma3.it/db/sigmod2020contest>

³<http://di2kg.inf.uniroma3.it>

⁴<https://github.com/huggingface/transformers>

of size 20 with softmax as the activation function.

Baselines. As baselines for comparison against E2L, we exploit two different approaches for entity labeling, named TFIDF and BART. The choice of TFIDF is motivated by the fact that this is almost a standard solution for terminology retrieval and it provides good results on the entity labeling task. The choice of BART is motivated by the idea of comparing E2L against a solution for document summarization, based on the idea that summarizing entity descriptions is an effective way to enforce entity labeling. Both approaches start by joining the `page_title` fields referring to the same camera name in the Alaska dataset. This way, we obtain a set P of 20 pseudo-descriptions, one for each camera. These pseudo-descriptions are then tokenized by exploiting the same procedure used in E2L.

- For the TFIDF baseline, we compute Tf-Idf on P . Then, for each pseudo-description $p \in P$, tokens are sorted by their Tf-Idf weights in descending order.
- As for the BART baseline, we feed each $p \in P$ to a pretrained BART model, namely `BART.large.cnn` [7]. As a result, we obtain a summary of p , that is a concise and shorter version of p . Then, we tokenize and process the summary as in the E2L approach. Tokens are sorted according to their position.

3.1. Experimental comparison

Let be $L_m^e = [(l_1, \rho_{l_1}), (l_2, \rho_{l_2}), \dots]$ a list of candidate labels for the entity e produced by the approach m , either one of the E2L versions or one of the baselines, sorted by their relevance score ρ_l from the most relevant to the less relevant. For each label e , we know the gold label t_e (i.e., the correct camera name) and we aim to evaluate the capability of E2L to build t_e by combining the candidate labels in L_m^e . Moreover, we aim to assess how many of the L_m^e labels we need to employ to obtain exactly the gold label t_e . The effectiveness of an entity labeling solution can be measured by observing how many candidate labels are required to obtain the gold label. The lower is the number of needed candidates (taken with relevance score in descending order), the higher is the effectiveness. According to this, the quality of each approach is measured as follows. First, we create the set T_{t_e} of the tokens in the gold label t_e , by extracting single terms (i.e., separated by spaces). Then, we do the same for the most relevant candidate label l_1 , by defining T_{l_1} as the set of tokens of l_1 . Given T_{t_e} , we define $T_1^m = T_{l_1}$ and we evaluate precision (P_1^m) and recall (R_1^m) of m at candidate 1 as:

$$P_1^m = \frac{|T_{t_e} \cap T_1^m|}{|T_1^m|}; R_1^m = \frac{|T_{t_e} \cap T_1^m|}{|T_{t_e}|}$$

This process is repeated for each of the k top candidate labels produced by m . At each step $k > 1$, we define T_k^m as:

$$T_k^m = T_{k-1}^m \cup T_{l_k}.$$

The F1-measure (F_k^m) at k is the harmonic mean of P_k^m and R_k^m . By exploiting these measures of precision and recall at k , we can easily check when the gold label t_e has been completely obtained (i.e., the k value where we have $R_k^m = 1$) and how many wrong tokens we have

	Avg. P_m^*	$P_m^* = 1$	$P_m^* = 1$ (fraction)
TFIDF	0.84	12	0.6
BART	0.71	7	0.35
E2L-Glove	0.86	14	0.7
E2L-Bert	0.92	17	0.85

Table 2

Precision at full coverage.

Merged Clusters	TFIDF	E2L-Bert
canon eos 7d (178), nikon 1 j3 (54)	canon eos 7d 013803117493 ebay 18	canon eos 7d 1 nikon j3
nikon d610 (79), nikon d3300 (79), nikon 1 j1 (78)	10 j1 nikon d610 ebay	j1 d610 nikon d3300
nikon d5200 (144), nikon d5100 (137), nikon d7000 (130)	nikon d7000 16 d5100	d7000 d5100 d5200 nikon

Table 3

Clusters are denoted by their gold label and cluster sizes are reported between parentheses.

collected during the process (i.e., P_k^m). Thus, we measure the overall quality of E2L and the baselines through the notion of *Precision at full coverage* (P_m^*) that is defined as follows:

$$P_m^* = P_k^m : R_k^m = 1$$

In Table 2, we report the values of precision at full coverage (P_m^*) for all the approaches, together with the number and fraction of entities that are correctly retrieved when recall is equal to 1 (i.e., $P_m^* = 1$), which means that the gold label has been not only completely retrieved by also retrieved by not introducing any noisy token, that is with no errors. The experimental results show that the use of black-box explanation techniques in E2L allows to extract relevant terminology for composing the correct label of entities as a final stage in a ER process. Indeed, if the statistical techniques seem to be effective for retrieving relevant terminology, they appear also to be more prone to introduce noisy terms in the candidate labels. On the other hand, data summaries, especially for text, tend to produce longer descriptions that are not enough synthetic to be taken as a good entity label. By contrast, the terms found by E2L appear as a good compromise in that they are more specifically related to the entities at hand, but also short enough to be useful for the task of labeling entities.

ER errors. Limitations of statistical techniques such as TFIDF are even clearer when there are errors in the input ER results. Consider for instance *entity merge errors*, where different real-world entities are mis-clustered as one entity. Table 3 reports preliminary results on a selection of clusters with different sizes from our Alaska dataset. Specifically, we considered clusters of different sizes, merged them, and computed labels with TFIDF and E2L-Bert. In the table, we show the labels with k equal to the size of the gold label for each of the considered merged clusters. In presence of merge errors, statistical techniques like TFIDF fail at identifying relevant terminology for all the sub-clusters in the merged cluster, while E2L-Bert can return the labels corresponding to the merged entities, thus supporting manual inspection of results and error detection.

4. Related work

Works related to the proposed E2L approach are about *entity labeling* as well as *machine learning interpretation*.

Entity labeling. A number of solutions has been proposed in the literature for entity labeling intended as the problem of finding a representative label to a set of records that refers to the same real-world object. A common solution is based on the idea to rely on an external knowledge base that works as a reference vocabulary for selecting the most appropriate label to assign to a given entity [8]. Entity labeling can be considered as a task of semantic data mining where labels emerge from record descriptions and they are selected according to the results of text processing techniques usually based on conventional information retrieval metrics (e.g., [9]). Machine learning techniques are also employed for entity labeling [10]. Automatic solutions to entity labeling can be integrated within *human-in-the-loop* workflows where domain experts are involved to validate the results of automated solutions (e.g., [11]).

Machine learning interpretation. In the recent years there was a surge of interest in the novel field of *interpretability* (see [12]). Explanation techniques can be distinguished between *black-box* and *white-box*. The former come with a model-agnostic interface while the latter rely on the internal mechanisms of the model. In E2L, we adopt LIME [4] that is a widely-employed black-box method. Other methods in the same category include SHAP [13] and Anchor [14].

5. Future Work

In this paper, we presented the E2L approach to entity labeling based on the use of techniques for classification and model explanation. Our current implementation features two representative text classification methods [3, 2] and one popular explanation method [4]. We plan as future works the inclusion of a wider choice of text classification and the inclusion of more explanation methods, such as SHAP [13] and Anchor [14].

Furthermore, a limitation of the current approach is that it depends on the ability of a supervised classifier to capture the entity properties. In principle, if the classifier is underperforming, the extracted labels can be less satisfactory. A simple solution could consist in training an ensemble of different models (as opposed to a single classifier) and select labels by using a voting system. A more sophisticated solution could be to model the ER process as a binary model indicating whether two records are matching and then apply directly the explanation engine, analogously to [15]. This can be non-trivial and it is left as future work. Indeed (i) computing pair-wise explanations exhaustively can be unfeasible for large datasets and (ii) different record pairs in the same entity can be matched for different reasons (e.g., some camera pairs could share only the model name while others could share not only the model name but also other technical specifications) and thus important tokens may vary significantly among pairs.

References

- [1] Y. Li, J. Li, Y. Suhara, A. Doan, W.-C. Tan, Deep entity matching with pre-trained language models, arXiv:2004.00584 (2020).
- [2] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv:1910.01108 (2019).
- [3] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.
- [4] M. T. Ribeiro, S. Singh, C. Guestrin, “Why Should I Trust you?” Explaining the Predictions of Any Classifier, in: KDD, 2016, pp. 1135–1144.
- [5] V. Crescenzi, A. De Angelis, D. Firmani, M. Mazzei, P. Merialdo, F. Piai, D. Srivastava, Alaska: A flexible benchmark for data integration tasks, arXiv:2101.11259 (2021).
- [6] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (1997) 1735–1780.
- [7] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, arXiv preprint arXiv:1910.13461 (2019).
- [8] D. Dou, H. Wang, H. Liu, Semantic Data Mining: A Survey of Ontology-based Approaches, in: ICSC, 2015, pp. 244–251.
- [9] X. Sun, Y. Xiao, H. Wang, W. Wang, On Conceptual Labeling of a Bag of Words, in: Int. Joint Conference on Artificial Intelligence, 2015.
- [10] F. N. C. de Araújo, V. P. Machado, A. H. M. Soares, R. de M.S. Veras, Automatic Cluster Labeling Based on Phylogram Analysis, in: IJCNN, 2018, pp. 1–8.
- [11] D. R. Karger, S. Oh, D. Shah, Efficient Crowdsourcing for Multi-class Labeling, in: SIGMETRICS, 2013, pp. 81–92.
- [12] Z. C. Lipton, The Mythos of Model Interpretability, *ACM Queue* 16 (2018) 31–57.
- [13] S. M. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.
- [14] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision Model-agnostic Explanations, in: Proc. of the 32th AAAI Conf. on Artificial Intelligence, 2018.
- [15] V. D. Cicco, D. Firmani, N. Koudas, P. Merialdo, D. Srivastava, Interpreting deep learning models for entity resolution: an experience report using LIME, in: aiDM@SIGMOD, 2019, pp. 8:1–8:4.