PUFFME: Probabilistic Unkeyed Feature Fusion for Matching Entities

Andrea Leoni[1], Andrea Molinari[2,3], Filippo Costamagna[1], and Simone Sandri[1]

[1] Okkam s.r.l, Via del Brennero, 260/G, Trento TN 38121, Italy
leoni,costamagna,sandri@okkam.it

[2] Dept. Of Industrial Engineering, University of Trento, Italy
andrea.molinari@unitn.it

[3] Lappeenranta University of Technology, Finland

Introduction

Entity matching (or entity resolution) seeks to identify records that refer to the same real-world entity across messy, heterogeneous data sources. Traditional methods depend on structured schemas, which limits their use when data is unstructured or inconsistent. PUFFME (Probabilistic Unkeyed Feature Fusion for Matching Entities) addresses this by comparing entities directly through bag-of-words attributes, modeling similarity with probabilistic distributions instead of schema alignment or manual feature engineering. Experiments on benchmark datasets show PUFFME reaching or surpassing state-of-the-art performance, proving its robustness and practicality for data integration under minimal assumptions.

Process

The main objective is to match entities through a function $\mathcal{M}(e_1, e_2)$ which returns a matching score between two entities. If the score is greater than a threshold θ there is a match, otherwise the entities refer to different objects as shown in eq. 1:

$$\text{match}(e_1, e_2) = \begin{cases} 1 & \text{if} \mathcal{M}(e_1, e_2) \geq \theta \\ 0 & \text{if } \mathcal{M}(e_1, e_2) < \theta \end{cases} \tag{1}$$

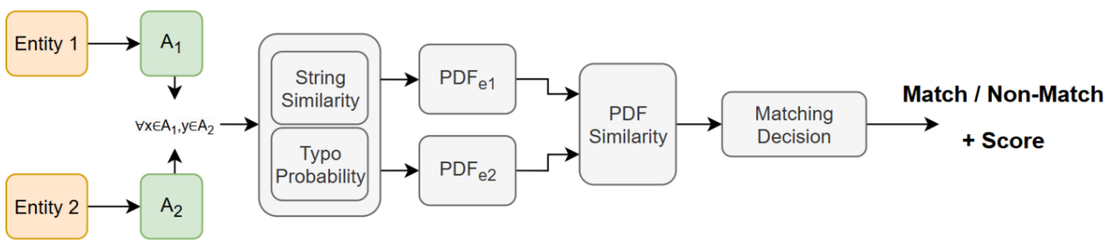

Figure 1: The entity matching process

The matching decision comes from the comparison of the entities' properties. Attributes are flattened into a bag of words (BOW) and a scoring function $\mathcal{M}_{\text{prop}}(e_1, e_2)$ is applied on top of BOW comparisons. Token similarity is computed by classical string similaity algorithms and best matching couples for each token in both BOWs are collected as shown in figure 2 and 3.

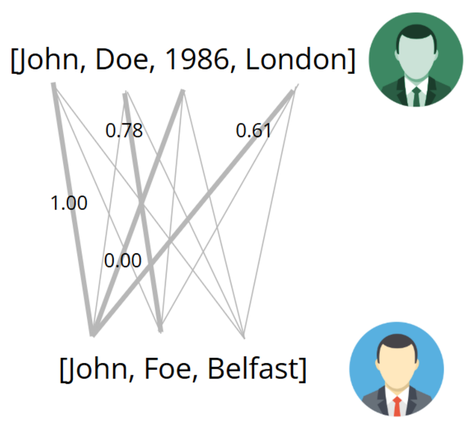

Figure 2: BOW comparison between e_1 and e_2

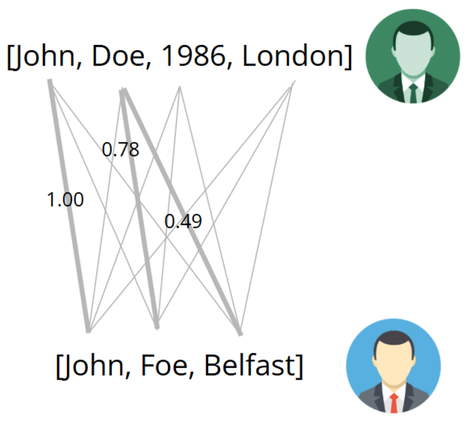

Figure 3: BOW comparison between e_2 and e_1

For each property p_1 from e_1 only the highest string similarity score w.r.t properties p_2 from e_2 is bucketed into bins to approximate probability distributions (PDF_{e1}). The same is done in order to create PDF_{e2}, taking the best matches between each property p_2 from e_2 w.r.t properties p_1 from e_1. Best similarity scores are now bucketed into b bins following the distribution $p(\frac{1}{b}, \frac{2}{b}, \ldots, \frac{b}{b})$. Each bucket will count how many string similarity scores $S(p_1, p_2)$ fall within that bin. The count is then normalized to form a PDF as in eq 2.

$$\text{PDF}_e(x) = \frac{\text{count of scores in bin } x \text{ for } e}{\text{total number of tokens in } e} \tag{2}$$

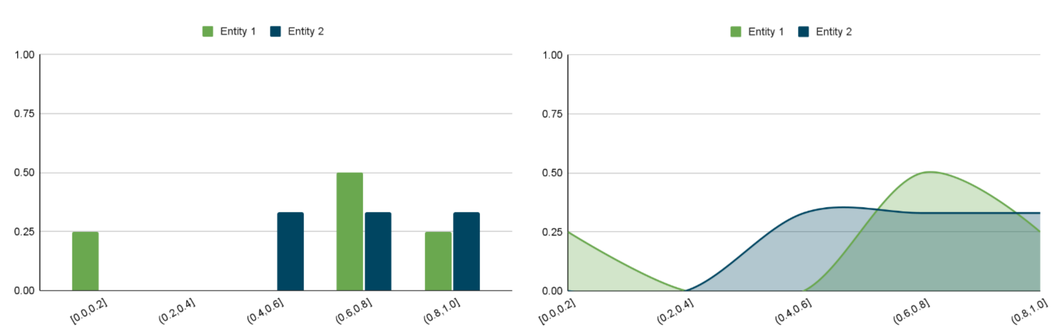

Figure 4: Histogram building and PDF approximation

Once the two distributions are built it is possible to apply a distance measure to quantify how different the two distributions are.

$$Dist(P, Q) = f(D(P||M), D(Q||M)) \tag{3}$$

This divergence measures the distance between the PDFs of e_1 and e_2, and the final similarity score would be:

$$\mathcal{M}_{\text{prop}}(e_1, e_2) = 1 - Dist(\text{PDF}_{e_1}||\text{PDF}_{e_2}) \tag{4}$$

To find out the best candidate distribution distances, we conducted preliminary testing on a fixed sample of 4000 pairs with a 0.5 rate of true matches. The top 5 measures with the best average F1-Scores are reported in Table 1.

Distance Measure	Plain Formula	Weighted Formula
Bhattacharyya	$D = -\log\left(\sum_{i=1}^{n}\sqrt{p_i q_i}\right)$	$D = -\log\left(\frac{\sum_{i=1}^{n} w(i)\sqrt{p_i q_i}}{\sum_{i=1}^{n} w(i)}\right)$
Harmonic Mean	$D = \sum_{i=1}^{n} \frac{2p_i q_i}{p_i + q_i}$	$D = \sum_{i=1}^{n} w(i) \cdot \frac{2p_i q_i}{p_i + q_i}$
Hellinger	$D = 2 \cdot \sqrt{\left(1 - \sum_{i=1}^{n}\sqrt{p_i q_i}\right)}$	$D = 2 \cdot \sqrt{\left(1 - \frac{\sum_{i=1}^{n} w(i)\sqrt{p_i q_i}}{\sum_{i=1}^{n} w(i)}\right)}$
Inner Product	$D = \sum_{i=1}^{n} p_i q_i$	$D = \sum_{i=1}^{n} w(i) \cdot p_i q_i$
Matusita	$D = \sqrt{2\left(1 - \sum_{i=1}^{n}\sqrt{p_i q_i}\right)}$	$D = \sqrt{2\left(1 - \frac{\sum_{i=1}^{n} w(i)\sqrt{p_i q_i}}{\sum_{i=1}^{n} w(i)}\right)}$

Table 1: Top 5 measures with best average F1-score. Plain formula and weighted transformation. In the Harmonic Mean case, when both p_i and q_i are 0, the contribution to the distance of bin i is set to 0 to avoid division by 0.

To increase sensitivity to a subset of components, we introduced non-negative weights w_i into the overlap sum. and, when necessary, normalizes by $\sum_{i=1}^{n} w_i$. Weighted versions of the plain formulas are shown in Table 1. For our purposes we used the weighting function described in eq. 5 which, in all cases, offered better discriminative power.

$$w(i) = \begin{cases} 1, & \text{if } i \leq \lceil \frac{n}{2} \rceil \\ \frac{n^2}{i \cdot (n-i)}, & \text{if } i > \lceil \frac{n}{2} \rceil \end{cases} \tag{5}$$

Resuls

The datasets taken in consideration to evaluate the performance of the method is Music-Brainz 20K, derived from real song records in the MusicBrainz database with artificial duplicates introduced.

	PDF Metric	bins	F1
PUFFME (JARO)	Inner Product	100	**0.982**
PUFFME (LCS)	Hellinger	10	**0.982**
PUFFME (JW)	Bhattacharyya	25	**0.981**
PUFFME (LD)	Bhattacharyya	10	**0.977**
PUFFME (LEVENSHTEIN)	Bhattacharyya	10	**0.976**
ALMSER-GB			0.951
FAMER-SplitMerge			0.880
FAMER-SPLIT			0.840

Table 2: Best parameter combinations for Music Brainz 20k dataset and comparison with other benchmarked techniques. For the other techniques different from PUFFME we took the best possible F1-score resulting from fine tuning described in the original papers.

Conclusions

- PUFFME (Probabilistic Unkeyed Feature Fusion for Matching Entities) is a new algorithm for entity matching that tackles the problem of aligning records without relying on schemas, semantic interpretation, or predefined keys. Instead, it operates directly on unstructured collections of properties expressed as bag-of-words, using a probabilistic framework to combine feature-level similarities. This schema-free and domain-independent design makes PUFFME broadly applicable in contexts such as data integration, record linkage, and entity matching system (EMS) alignment.
- Empirical tests on two heterogeneous datasets confirm its effectiveness: PUFFME reached an F1 score of 0.982 on a music dataset, slightly surpassing the state of the art. These results show that PUFFME can deliver accurate and robust entity matching without the need for domain-specific tuning, handcrafted features, or semantic enrichment, making it a versatile solution for diverse data conditions.
- Future directions for PUFFME involve broadening its evaluation to a wider range of benchmark datasets to test generalization across domains, systematically analyzing its robustness by perturbing or removing parts of entity descriptions, and improving the model with feature weighting schemes. Since PUFFME currently treats all features as equally informative, introducing statistical or learned weightings would allow it to better emphasize discriminative attributes and handle real-world data asymmetries more effectively.

Acknowledgements

The research presented in this paper is partially supported by the MAESTRO project, sponsored by the Autonomous Province of Trento - Italy, through L.P. n. 6/99 and L.P. n. 6/23