TOMATO : results of the 2023 OAEI evaluation campaign

Philippe Roussille[1], Olivier Teste[2]

[1]*École 3iL, Limoges, Institut de Recherche en Informatique de Toulouse, Toulouse, France*
[2]*Université Toulouse 2 Jean Jaurés, Institut de Recherche en Informatique de Toulouse, Toulouse, France*

Abstract

This paper presents the results obtained by TOMATO in the OAEI 2023 evaluation campaign. We describe here the results in the Conference track. We report a general discussion on the results and future improvements of the system.

1. Presentation

1.1. Overview

TOMATO (***TO**olkit for **MAT**ching **O**ntologies*) takes inspiration from previous work on ontology matching systems such as POMAP++ [1]. TOMATO is designed as a pairwise matcher, aligning pairs of input ontologies against each other. At its core, TOMATO utilizes machine learning approaches to learn from element similarities. In earlier versions, it focused mainly on string-based similarity measures of ontology elements [2].

However, in 2023 we started working beyond this initial approach. While string similarities continue to provide baseline features, we have incorporated additional structural and semantic similarity measures. This includes leveraging relationships between ontology entities, as well as leveraging external knowledge sources. The goal is to move beyond solely lexical matching and capture more of the intended meaning during the alignment process.

Lacking a robust ground truth beyond the reference alignment, this line of work was shelved temporarily to investigate supplemental evaluation methods.

In this iteration of TOMATO, we took a more direct approach to developing the optimal matching strategy. Rather than starting with local strategies as in the previous year[3], our aim was to first identify the best global strategy across all entity types.

To achieve this, we focused on empirically determining the most pertinent similarity measures to incorporate. In our evaluations, we considered a broader set of 11 measures: Levenshtein, Jaccard, Jaro-Winkler, cosine, iSub, dice, 3-gram, Monge-Elkan Levenshtein, Monge-Elkan Jaro-Winkler, and overlap coefficient.

By systematically evaluating these measures in various combinations through our machine learning models, our goal was to arrive at the global configuration that performed best when

Proceedings of the 18th International Workshop on Ontology Matching
Philippe.Roussille@irit.fr (P. Roussille); Olivier.Teste@irit.fr (O. Teste)
© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC > BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3591/oaei23_paper10.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

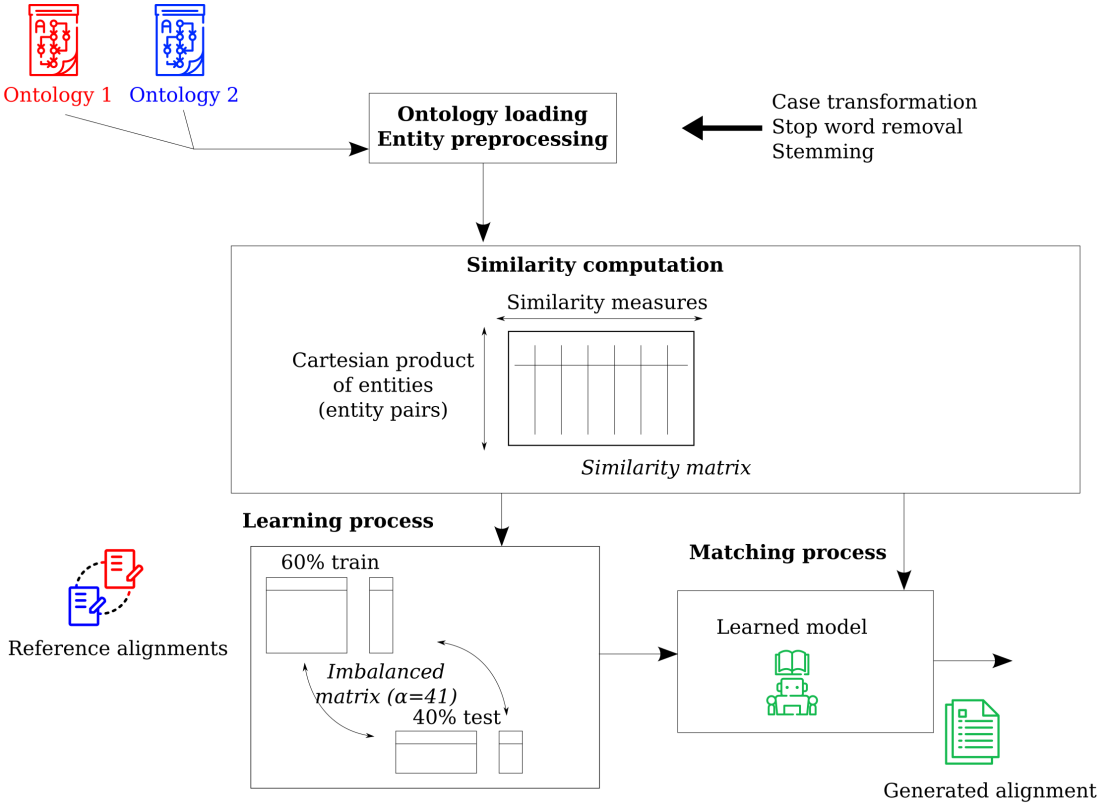

Figure 1: System workflow

applied uniformly. Once identified, we could then examine how to potentially refine this global strategy by incorporating localized decompositions based on entity characteristics, as we had explored in earlier versions of TOMATO.

This new evaluation-driven approach aimed to build up the optimal strategy incrementally, starting from the best global foundation before incorporating targeted local enhancements. The rationale was that this may yield stronger baseline results compared to our prior method of starting with localized configurations.

The workflow in TOMATO begins by taking as input two OWL ontologies to be aligned, along with their associated reference RDF alignment files. These reference alignments provide examples of entity matches that serve as ground truth for the learning process.

TOMATO then prepares the data by combining all ontology entity pairs (both matched and unmatched based on the references) into a single dataset. This set contains all possible entity couples across the input ontologies, along with their reference match status.

From here, TOMATO can be used in one of two modes:

1. Learning mode: A machine learning model is trained on the full mixed dataset to learn the patterns of matching vs non-matching entities.

2. Matching mode: A pre-trained model is applied to new ontology entity pairs to predict their alignment status.

Unlike previous versions where we explored a variety of similarity measures, this year our focus is on determining which measures are most pertinent to the matching task and which can be eliminated.

By fully combining all entity pairs and reference alignments upfront, TOMATO is able to learn from all available matching examples without viewing the ontologies separately.

1.2. Matching steps

The initial step in TOMATO's workflow is to parse and ingest the input ontologies. We leverage the `owlready2` Python library to build an in-memory representation containing all ontology elements and relationships.

Each entity such as classes, data properties and object properties is indexed via a unique identifier, with the preferred identifier being the element's label if present. As a fallback, the final segment of the entity's URI is used.

Structural relationships between classes are also extracted and stored. This includes superclass-subclass links as well as relations defined through object properties.

By fully populating an internal graph structure in this way, TOMATO is able to consider both lexical properties of entities as well as their positions and connections within the ontology taxonomy during the matching process. This combined view aims to capture more contextual evidence about intended semantic correspondence compared to considering elements in isolation.

The loaded ontologies can then be queried as needed during the various steps of similarity computation, model training and alignment prediction.

1.2.1. Ontology Preprocessing

As in prior iterations, we apply standard text preprocessing techniques to clean and normalize entity labels before computing similarities. This includes:

- Converting CamelCase to snake_case
- Replacing non-alphanumeric symbols with spaces
- Performing English stemming
- Removing stop words

However, unlike previous versions that employed matching strategies based on element types, in this work we utilize K-means clustering as described below.

1.2.2. Similarity Score Computation

All possible entity pairs between the input ontologies are generated without regard to element type (e.g. class vs property). For each pair, we compute similarities across several measures (Levenshtein, Jaro-Winkler, etc.).

New to this year, rather than hardcoding similarity computation strategies, we compute directly over the multi-dimensional similarity space. As we showed last year, this will allow us to focus on the worst possible outcome, which could be further improved in a case-by-case local strategy.

1.2.3. Train and match

Training a Model We use GridSearchCV from scikit-learn to determine the best combination of similarity measures for predicting matches. The dataset is split 60/40 for training/testing with 10-fold cross validation. The reference alignments are used to label entity pairs as matches (1) or non-matches (0). This labeled similarity matrix is fed to an SVM classifier to train a matching model.

The TOMATO system is based on a machine learning approach that exploits 60% of data for training and 40% of data for testing. The training set is classicaly used to learn a model that consists in finding an optimal weighting of different similarity measures. Clearly, we use a subset of the reference alignments provided by OAEI that is considered as an overfitting. However, it is important to consider that the obtained model never considered the 40% set aside for testing. To avoid this overfitting, we plan to use external resources as ground truth during the training phase[4].

Computing Alignments A similarity matrix is constructed for the target ontology pair. This matrix is then input to the pre-trained matching model, which outputs the predicted entity alignments.

Unlike previous versions, in this work we utilize a distributed approach to avoid crashes or slowdowns when processing large or dense ontologies. The classification tasks are handled in parallel across computational threads/processes for improved robustness and performance.

1.3. Adaptations made for OAEI

1.4. Technical Adaptations for OAEI

In preparation for the OAEI evaluation, we focused on improving our ability to handle large ontologies in a distributed manner. Previously, training classifiers on the full matching dataset could lead to memory issues or lengthy processing times.

Therefore, in this iteration we implemented a multi-threaded resampling scheme during model training. The matching data is partitioned and resampled in parallel across CPU threads, with intermediate results merged to update the shared model.

Additionally, we experimentally found that training classifiers with a class imbalance favoring non-matching examples (e.g. 90% negative, 10% positive) led to more effective models compared to a balanced class distribution. This better reflects the natural skew in real-world ontology alignments.

To validate different modeling configurations, models trained with both global and local strategies were considered. However, due to time constraints of the competition format, we only submitted results using our global strategy models for evaluation.

Table 1
TOMATO class balancing result

	$\alpha = 1$	$\alpha = 41$
Macro Precision (P)	0.062	0.442
Macro Recall (R)	0.737	0.429
Macro F1	0.114	0.430
Micro Precision (P)	0.048	0.444
Micro Recall (R)	0.754	0.391
Micro F1	0.090	0.408

The enhancements to our data processing and sampling methodology aimed to enable TOMATO to efficiently learn from very large ontologies in real-world settings, while still optimizing for high matching accuracy.

2. Evaluation of Class Balancing Techniques

2.1. Class Imbalance Problem

We face a significant class imbalance problem in our ontology matching task, where the number of positive matches between ontologies is much smaller than the number of negative matches. This poses challenges for training effective machine learning models.

During model training, the non-matching or negative examples dominate the training data distribution. As a result, models can become biased towards predicting negatives and fail to properly learn from the minority positive class.

2.2. Proposed Resampling Method

To address class imbalance, we implement a resampling scheme during model training based on a resampling proportion hyperparameter α.

The dataset consists of positive matches labeled 1 and negative matches labeled 0. We partition the negative examples into $\lfloor \alpha n_1 \rfloor$ subsets of size n_1, where n_1 is the number of positives.

We then create a balanced training set by combining the positive examples with a randomly selected negative subset. This ensures equal representation of both classes for learning. We vary α to study its impact.

2.3. Experiments and Results

2.3.1. Raw results

We train several models on matching data, varying $\alpha \in [1, 100]$. Performance is evaluated using macro/micro precision, recall and F1.

Table shows optimal performance is achieved with $\alpha = 41$, with macro F1=0.43. Lower α leads to poor predictive ability for negatives, while higher α may worsen learning of positives.

This table summarizes the key performance metrics for our models across different values of the resampling proportion α, showing the optimal value is α=41.

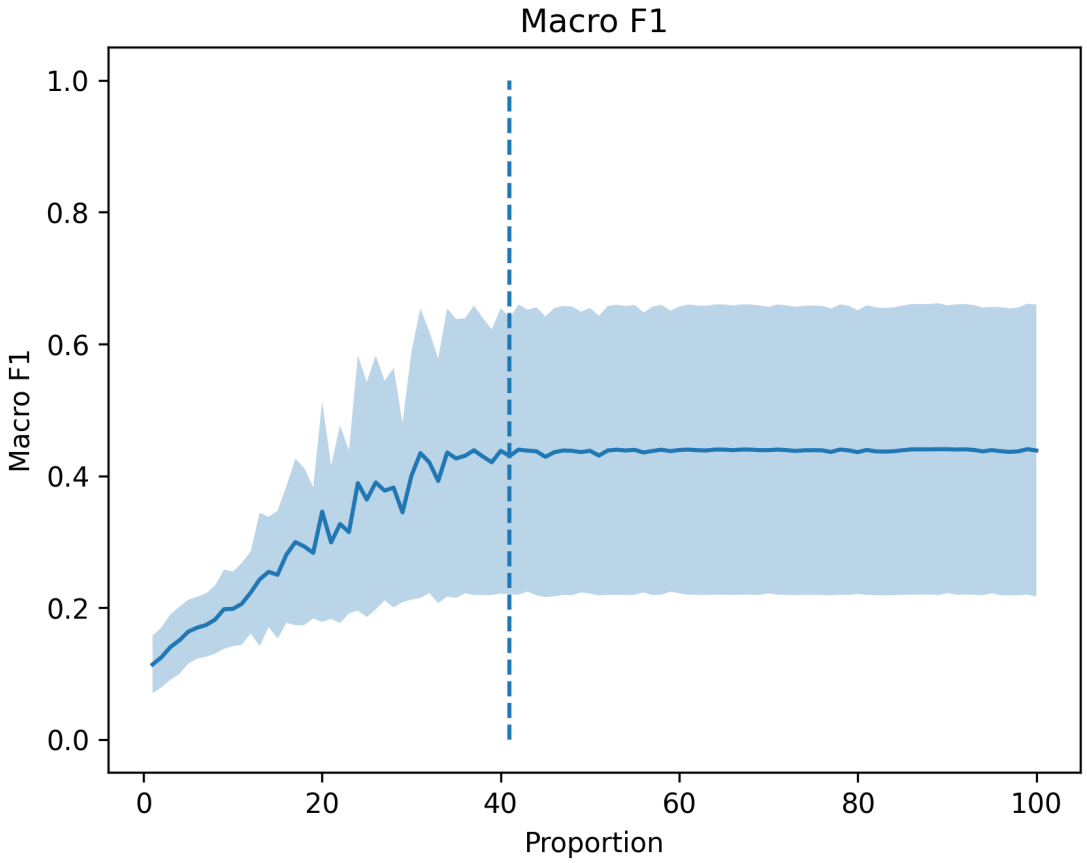

Figure 2: This figure plots the macro F1 score on the y-axis against different values of the resampling proportion α on the x-axis. It shows that the Macro F1 score increases as α is increased from 1, peaks at around α=41. The variability then starts increasing as α is increased further. The optimal resampling proportion based on this metric is α=41.

The macro F1 score is a weighted average of F1 scores calculated for each class individually, giving equal importance to each class regardless of prevalence.

The micro F1 score aggregates counts across all classes to determine a single F1 score. It places more weight on accuracy for frequent, majority classes compared to macro F1 which balances across classes.

Together these metrics provide a more comprehensive assessment of model performance for both frequent and rare classes in a multiclass classification problem like ontology matching.

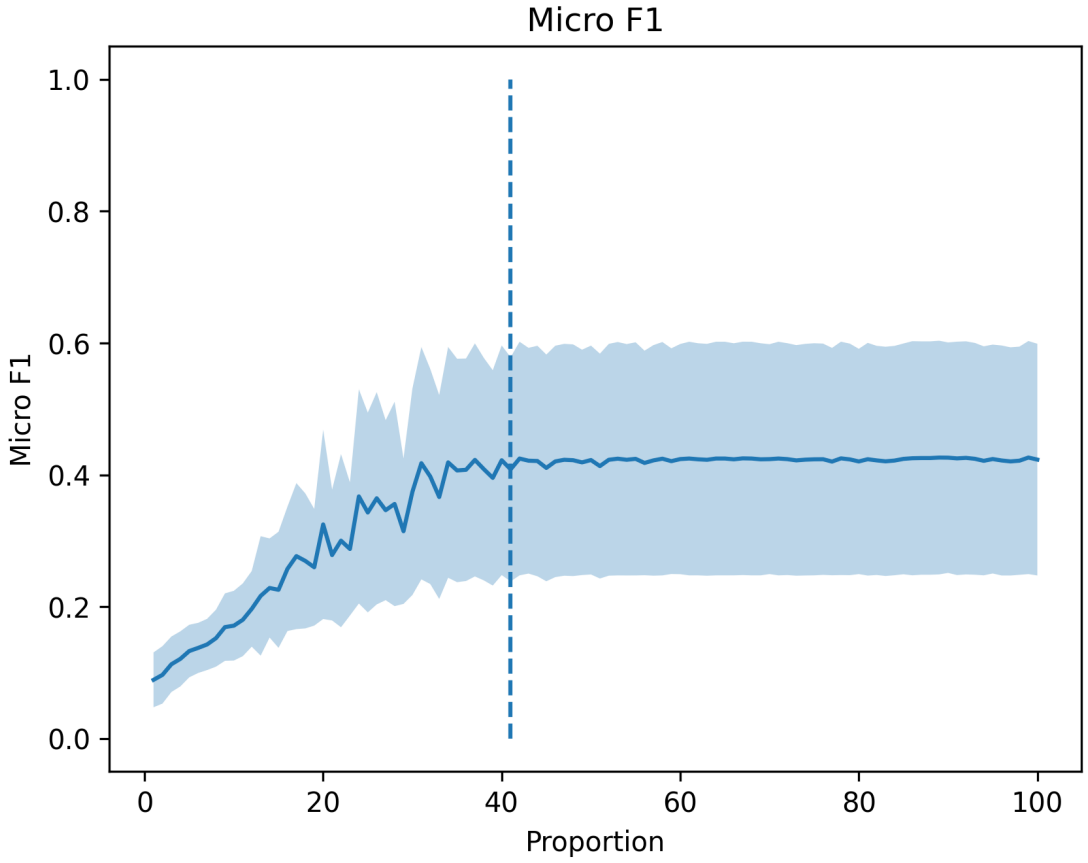

Figure 3: This figure shows the variation of Micro F1 score with respect to the resampling proportion α. Similar to the Macro F1 plot, the Micro F1 score increases initially with α, reaches a maximum at around α=41, and then shows high unstability with higher values of α. The trend confirms that α=41 provides the best balance between positive and negative class representations during training to optimize overall classification performance.

2.4. Analysis

The resampling balances overall class distributions during training, addressing biases towards negatives. Higher proportions improve discrimination of matches by emphasizing difficult positive examples. Optimally, sufficient data from both classes is provided for robust learning.

However, residual recall indicates mismatches remain challenging. Future work involves combining resampling with techniques like data augmentation to further boost positive example diversity and learning.

In conclusion, resampling addresses class imbalance effectively, enhancing ontology matching model performance. Careful tuning of resample rates leads to improved classification accuracy.

Table 2
TOMATO performance on rar2-M3

Metric	Value
Precision	0.57
Recall	0.47
F-Measure	0.52

Table 3
Comparison of TOMATO 2022-2023 results

	2022	2023	Change
Precision	0.53	0.57	↑
Recall	0.43	0.47	↓
F-Measure	0.47	0.52	↑

2.5. OAEI results

3. Evaluation Results for the OAEI 2023 Conference Track

This section analyzes the evaluation results published on the OAEI 2023 Conference Track results page (https://oaei.ontologymatching.org/2023/results/conference/).

3.1. Participating Matching Systems

A total of 11 matching systems participated in this track and produced alignments over the Conference domain ontologies: ALIN, AMD, GraphMatcher, LogMap, LogMapLt, LSMatch, Matcha, OLaLa, ProMatch, SORBETMtch, and TOMATO.

3.2. Evaluation Based on Crisp Reference Alignments

The systems were evaluated based on the main reference alignment rar2-M3 using precision, recall, F-measure, and other metrics.

TOMATO achieved a precision of 0.57, recall of 0.47, and F-measure of 0.52 on rar2-M3 as shown in Table 2. This placed it below the StringEquiv baseline but above ProMatch. Compared to 2022 results in Table 3, TOMATO's precision and F-measure increased slightly while recall decreased.

TOMATO's performance remains in the mid-range. While recall could improve, gains were seen in precision and F-measure compared to 2022.

4. Conclusion and Future Work

In this paper, we presented the results of the TOMATO ontology matching system on the OAEI 2023 Conference track benchmark. TOMATO employs a machine learning approach combined with strategy selection based on domain characteristics.

Our evaluation showed that TOMATO achieved moderate performance. While its precision and F-measure improved slightly over last year, recall decreased. Nonetheless, TOMATO demonstrated the utility of adapting its strategy based on ontology features.

There remain opportunities for enhancing TOMATO's matching capabilities. In future work, we plan to explore:

- Additional machine learning classifiers and similarity measures
- Deeper analysis of ontology structures to better leverage structural context
- Techniques like data augmentation to improve generalization of learned models
- Integration of external sources like knowledge graphs to strengthen matching

We are optimistic that such improvements can lift TOMATO's rankings on this benchmark by further refining its abilities to identify correct mappings. Continued participation in OAEI also allows ongoing evaluation versus state-of-the-art matchers. Overall, the results provide motivation to push the boundaries of adaptive ontology matching.

References

[1] A. Laadhar, F. Ghozzi, I. Megdiche, F. Ravat, O. Teste, F. Gargouri, POMap++ results for OAEI 2019: fully automated machine learning approach for ontology matching, in: 14th International Workshop on Ontology Matching co-located with the International Semantic Web Conference (OM@ISWC 2019), Auckland, New Zealand, 2019, pp. 169–174. URL: https://hal.archives-ouvertes.fr/hal-02942337.

[2] M. Cheatham, P. Hitzler, String similarity metrics for ontology alignment, in: H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, K. Janowicz (Eds.), The Semantic Web – ISWC 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 294–309.

[3] P. Roussille, O. Teste, Tomato: results of the 2022 oaei evaluation campaign, in: 17th International Workshop on Ontology Matching co-located with the International Semantic Web Conference (OM@ISWC 2022), Hangzhou, China, 2022, pp. 1–13. URL: http://disi.unitn.it/~pavel/om2022/papers/oaei22_paper13.pdf.

[4] A. Laadhar, Local matching learning of large scale biomedical ontologies, Ph.D. thesis, Université Paul Sabatier - Toulouse III, 2019. URL: https://theses.hal.science/tel-02651332.