DLinker Results for OAEI 2022
Happi Bill Gates, Géraud Fokou Pelap, Danai Symeonidou, Pierre Larmande

To cite this version:
Happi Bill Gates, Géraud Fokou Pelap, Danai Symeonidou, Pierre Larmande. DLinker Results for OAEI 2022. 17th International Workshop on Ontology Matching, OM 2022, Oct 2022, Online, France. pp.166-173. hal-03967103

HAL Id: hal-03967103
https://hal.inrae.fr/hal-03967103
Submitted on 1 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License
DLinker Results for OAEI 2022*

Bill Gates Happi Happi1,∗,†, Géraud Fokou Pelap2,†, Danai Symeonidou3,† and Pierre Larmande1,†

1DIADE, University of Montpellier, IRD, CIRAD, 911 Av. Agropolis, 34394 Montpellier, France
2University of Dschang, Dschang ville, Cameroon
3INRAE, SupAgro, UMR MISTEA, University of Montpellier, 2 place pierre viala 34060 Montpellier, France

Abstract

DLinker is a system for matching instances of two RDF data sources. Its performance is mainly based on the deep comparison of literals. The main comparison algorithm is based on the search for the longest common subsequence (LCS[1]). DLinker performs the matching in two steps. First, during the training step, the algorithm learns from the hyperparameters to optimize the prediction of pairs of similar instances. Finally the tool exploits the model tuning for the prediction step. DLinker participated in 2 tracks composed of 9 challenges. DLinker demonstrated its ability to process different data with good accuracy and in a very short time. Additionally, in the context of the SPATIAL challenge DLinker has outperformed the state of the art finishing first with the shortest time. Overall, DLinker exposes different strengths and weaknesses that are discussed in this work.

Keywords

Instance Matching, Syntactic Similarity, Data Linking Algorithm, Data Processing, Synonyms

1. Presentation of DLinker

1.1. General Statement

DLinker is a generic instance matching system. Its performance is mainly based on a recursive algorithm which compares literals using the longest common subsequence (LCS[1]). DLinker performs the matching in two steps. First, during the training step, the algorithm learns from the hyperparameters to optimize the prediction of pairs of similar instances. Finally the tool exploits the model tuning for the prediction step. DLinker participated in 2 tracks composed of 9 challenges.
challenges during the OAEI campaign. The tool achieved good results during the SPATIAL track, as it was the best on temporal and accuracy performances on small and large scale EQUALS and OVERLAPS topological tasks for TOMTOM and SPATEN. During the SPIMBENCH track, we participated on small datasets while providing good accuracy in a considerable time. The main similarity measure of the tool is based on the Longest Common Sub-Sequence (LCS[1]) algorithm, which is seen as a deep variant of this one, imposing a very large capacity for various applications. The temporal factor of DLinker is justified by the parallelization of the pairs of literal objects to be compared. DLinker was wrapped by Hobbit Framework [2].

1.2. System overview

Initially, the theoretical formalization of DLinker consisted in setting up a multilingual data binding tool during data processing. However, the current version performs this task properly when the sources to be linked belong to the same language. As shown in figure 1, DLinker matches only instances. We give as inputs two data sources to be linked in different formats, linked or not, to produce as output a set of alignments in a suitable data format.

![Figure 1: Overview of DLinker’s pipeline](image-url)
1. **Loading**: loads the input files and returns an RDF data graph in the form of triples. Also loads the hyperparameters (HP) from the data training. Let us give a brief description of these hyperparameters:

- **Predicate threshold**: represents the numerical value of similarity recognition between two predicates;
- **Literal threshold**: represents the numerical value of similarity recognition between two literals;
- **Acceptation threshold**: represents the minimal number of pairs of similar objects belonging to the instances that can be linked;
- **Measurement depth**: represents the number of times we should search for the longest common subsequences between the pairs of objects of the instances. It can also be seen as the depth of recursion during the search for similar literals;

2. **Compute Similar Predicates**: Retrieves the unique predicates of each graph and returns pairs of similar or complementary predicates existing between them using the threshold of the predicate;

3. **Select and Filter the Specific (s, o) Pairs to Compare**: First, a selection of each pair (s, o) (respectively (s’, o’)) in each of the two data sources from the predicate pairs (pi, p’j) (similar) is performed. Then, a construction of sub-lists of cartesian products ((s, o),(s’, o’)) for the (s, o) (respectively (s’, o’)) of these predicates, we proceed to the step of reducing the number of unnecessary comparisons between the literal objects of the pairs of the sub-sets by calculating the completeness or proximity ratio between them;

4. **Compare Pairs and Validations**: Here we realize the comparison of the literal objects associated to the couples ((s, o),(s’, o’)). These comparisons are validated using the threshold of the literal before or after the similarity search depth (measurement depth hyperparameter). The validation is performed once the Acceptation Threshold (AT) is reached after several successful object comparisons. To ensure proper validation of comparison peer topics, we perform a hash of the concatenation of the topics which are then stored as <key, value>. The value of the key increments to reach AT if the associated object comparisons are positive;

5. **Generate Alignment**: The process of comparisons being parallel, a list of pairs (s, s’, relation, 1) is filled after validations by each sub-processing of the subset of pairs of ((s, o), (s’, o’)) based on the acceptation threshold hyperparameter.

6. **Output File**: the alignments are generated according to the format expected in output, for the moment we provide “.rdf” and “.nt”.

## 2. Results

This section describes the results of the DLinker system on two tracks namely: SPATIAL and SPIMBENCH. The evaluation was executed on a Linux virtual machine with 256 GB of RAM and 32 vCPUs (2.4 GHz) processors. The table below presents the different hyperparameters that were used to obtain the following results:
Table 1
Hyperparameters of DLinker for each track.

<table>
<thead>
<tr>
<th>Tracks</th>
<th>Predicate Threshold</th>
<th>Literal threshold</th>
<th>Acceptation threshold</th>
<th>Measurement depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial</td>
<td>1.0</td>
<td>0.3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Spimbench</td>
<td>1.0</td>
<td>0.8</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2.1. SPATIAL Data

This track\(^1\) concerns data that are part of SPATIAL data management systems and that store topological relationships in the form of SPATIAL resources that can be linked together. These SPATIAL resources are described from a large information set such as LinkedGeoData. These data are sent from a SPATIAL benchmark generator\(^2\). This Benchmark supports several topological relations (Equals, Disjoint, Touches, Contains/Within, Covers/CoveredBy, Intersects, Crosses, Overlaps). This SPATIAL generator contains three data generators (TomTom, Spaten and DEBS).

1. Spaten is an open-source configurable spatio-temporal and textual dataset generator, that can produce large volumes of data based on realistic user behavior. Spaten extracts GPS traces from realistic routes utilizing the Google Maps API, and combines them with real POIs and relevant user comments crawled from TripAdvisor. Spaten publicly offers GB-size datasets with millions of check-ins and GPS traces;
2. TomTom provides a Synthetic Trace Generator developed in the context of the HOBBIT Project, that facilitates the creation of an arbitrary volume of data from statistical descriptions of vehicle traffic. More specifically, it generates traces, with a trace being a list of (longitude, latitude) pairs recorded by one device (phone, car, etc.) throughout one day. TomTom was the only data generator in the first version of SPgen;
3. DEBS provides a selection of AIS data collected from the MarineTraffic coastal network. It has been used for the EU H2020 Research Project BigDataOcean and the ACM DEBS Grand Challenge 2018.

The results below are available at this address: https://hobbit-project.github.io/OAEI_2022.html.

2.1.1. Evaluation on SandBox LINESTRINGS - LINSTRINGS

- Under the EQUALS topological relationship, DLinker terminated in 1667ms for Spaten and 10487ms for TomTom providing the highest accuracy and smallest overall time;
- Under the OVERLAPS topological relationship, DLinker finished in 3236ms for Spaten and 3087ms for TomTom providing the highest accuracy and lowest overall time. The summary can be found in the figure 2.

\(^1\)https://project-hobbit.eu/hobbit-spatial-benchmark-v2-0/
\(^2\)https://github.com/hobbit-project/SpatialBenchmark
2.1.2. Evaluation on MainBox LINESTRINGS - LINESTRINGS

- Under the OVERLAPS topological relationship, DLinker terminated in 2026ms for Spaten and 37072ms for TomTom providing the highest accuracy and smallest overall time;
- Under the OVERLAPS topological relationship, DLinker terminated in 2547ms for Spaten and 5458ms for TomTom providing the highest accuracy and smallest global time. The summary can be found in the figure 3.

2.2. SPIMBENCH Data

The datasets [3] in this track are produced using SPIMBENCH benchmark generator [4] with the aim to generate descriptions of the same entity where valuebased, structure-based and semantics-aware transformations are employed on a source dataset in order to create the target dataset(s). The goal of the SPIMBENCH task is to determine when two instances describe the same Creative Work. A dataset is composed of a Tbox (contains the ontology and the instances) and a corresponding Abox (contains only the instances). The datasets share almost the same ontology (with some difference in the properties’ level, due to the structure-based transformations). What we expect from participants. Participants are requested to match instances in the source dataset (Tbox1) against the instances of the target dataset (Tbox2). The task goal is to produce a set of mappings between the pairs of matching instances that refer to the same real-world entity. An instance in the source dataset (Tbox1) can have none or one matching counterparts in the target dataset (Tbox2). Note that only instances of Creative Work are to be mapped in this task3. The instances of the other classes appearing in the sources are used to examine if the matching systems take into account RDFS [5] and OWL [6] constructs in order to discover correspondences between instances that can be found only by considering schema information. After processing 380 instances (10000 triples) for each file (source and target), we obtained the scores presented in the following section.

2.2.1. Evaluation on small data set

DLinker participated in the evaluation of small data set and finished in 15555ms with an accuracy of 0.791 as shown in the table 2 below.

Table 2
Results of DLinker on SPIMBENCH track 2022.

<table>
<thead>
<tr>
<th>Matcher</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Runtime(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLinker</td>
<td>0.791</td>
<td>0.632</td>
<td>0.703</td>
<td>15555</td>
</tr>
<tr>
<td>LogMap</td>
<td>0.938</td>
<td>0.763</td>
<td>0.841</td>
<td>6116</td>
</tr>
</tbody>
</table>

3https://hobbit-project.github.io/OAEI_2022.html
3. General comments and conclusions

Comments on the results. DLinker was ranked first on the SPACIAL track and second on the SPIMBENCH track. Its fast processing time is due to the parallelization of the processing of literal comparisons in the analysis of instances based on the LCS and its high precision. The high accuracy can be guaranteed on so-called in-place comparisons and not on those requiring synonym consideration. The non-participation of synonyms in the instance matching process is the main weakness of DLinker and prevents it from correctly projecting itself in the ontology alignment.

Space for improvement in our system. Although DLinker seems stable on several datasets already evaluated so far, we plan to make it even more robust and efficient for other challenges. We have seen many exciting possibilities for future work. For example, we intend to implement multilingual functionality to link two data sources of different languages and integrate the consideration of synonyms during comparisons. We also plan to realize an automatic hyperparameters generator from the training datasets.

Acknowledgments

This work was supported by the IRD DIADE UNIT and the FOOSIN project. We also thank the organisers of the OAEI evaluation campaign for providing test data and infrastructure.

References


A. Online Resources

The sources for DLinker python and Java version are available via:
• Python Version: https://github.com/BillGates98/DLinker,
• Java version: https://github.com/BillGates98/dlinker-adapter,

B. Diagrams
Figure 2: Results of DLinter on SandBox for TOMTOM and SPATEN.

Figure 3: Results of DLinter on MainBox for TOMTOM and SPATEN.