Using Machine Learning for Link Discovery on the Web of Data

Axel-Cyrille Ngonga Ngomo[1] and **Daniel Obraczka** [2] and **Kleanthi Georgala**[3]

Abstract. Link discovery is of key importance during the creation of Linked Data. A large number of link discovery frameworks for RDF data has thus been created over the last years. In this demo, we aim to present the machine-learning features of the novel graphical user of the LIMES framework, which is a state-of-the-art link discovery framework that implements a large number of machine-learning features. We will guide the participants in the demo and show how LIMES uses machine learning between the selection of attributes to the linking of data. In particular, we will focus on how LIMES uses different paradigms such as batch learning, active learning and unsupervised learning to support link discovery. We will use both real and synthetic data to demonstrate the scalability of our implementations.

1 INTRODUCTION

The amount of RDF data available on the Web has grown significantly over the last years.[4] The need for scalable *Link discovery frameworks*, i.e., frameworks that can compute typed relations between RDF resources (e.g., link all the capital cities in GeoNames with the corresponding countries in DBpedia) has hence become over more pronounced over the last years. The problem of scalable and time-efficient Link Discovery has been thus addressed by several frameworks [2] including *LIMES*[4], SILK [10], *KnoFuss* [8] and *Zhishi.links* [9]. These tools incorporate declarative approaches towards Link Discovery. For example, SILK and *KnoFuss* using blocking techniques to identify links between knowledge bases so as to avoid unnecessary comparisons between resources. *LIMES* reduces the time-complexity of the LD procedure by combining techniques such as *PPJoin+* [11], HR^3 [3] and AEGLE [1] with set-theoretical operators and planning algorithms.

In addition to implementing time-efficient approaches for link discovery, *LIMES* implements a number of machine-learning algorithms based on genetic programming [5, 7] as well as hierarchical grid search [6]. In particular, *LIMES* implements three different machine-learning paradigms: batch learning, active learning and unsupervised learning. The aim of this demo is to present the novel interface of LIMES and to show how these different categories of machine learning can be used to achieve Link Discovery on the Web of Data under different circumstances. Here, we refrain to explain the machine learning approaches in detail (see [5, 7, 6] for explanations) and focus of the novel user interface through which these machine learning can be used. This paper is a companion paper to the ECAI 2016 paper AEGLE [1].

[1] AKSW Research Group, University of Leipzig, Germany, email: ngonga@informatik.uni-leipzig.de
[2] AKSW Research Group, University of Leipzig, Germany, email: daniel@obraczka.de
[3] AKSW Research Group, University of Leipzig, Germany, email: georgala@informatik.uni-leipzig.de
[4] `http://stats.lod2.eu`

2 GRAPHICAL USER INTERFACE

The core of LIMES' new user interface is to guide the end user during the Link Discovery process. Expert users are commonly able to decide upon which link specification they would like to execute to achieve a certain linking task (e.g., linking movies from LinkedMDB and DBpedia). Hence, our framework supports the manual creation of link specifications as shown in Figure 1. With these link specifications, an expert user can describe under which conditions a link potentially holds between two resources. Formally, this is equivalent to creating a binary classifier.

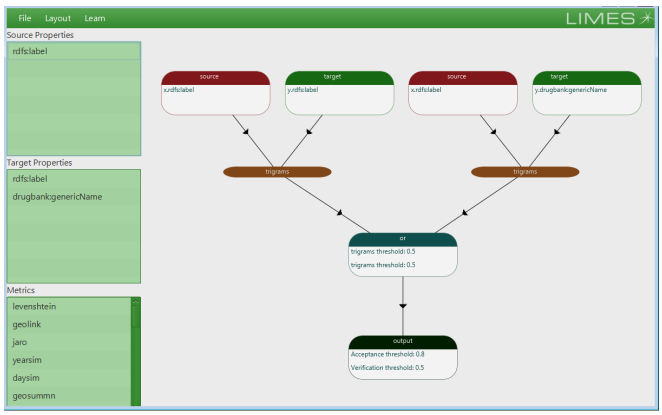

Figure 1. Link Specification Interface

Most users however do not know how to create link specifications and need to be helped throughout the process from RDF datasets to link discovery. To this end, LIMES supports three types of machine learning: batch, active and unsupervised (see Figure 2). In all cases, the user can select a machine learning algorithm (e.g., EAGLE, which uses genetic programming). After modifying the default configuration of the algorithm if need be, the user can simply run the said algorithm. It is then the job of the approach to detect links. In case of batch or unsupervised learning, the execution is carried out once and the results are displayed to the user, who also has the possibility to export them. In case of active learning, LIMES commonly relies on a commity-based approach to determine the most highly informative link candidates which are presented to the user [6, 5, 7]

(see Figure 3). The user can iteratively select the most information link candidates until (s)he is satisfied with the result.

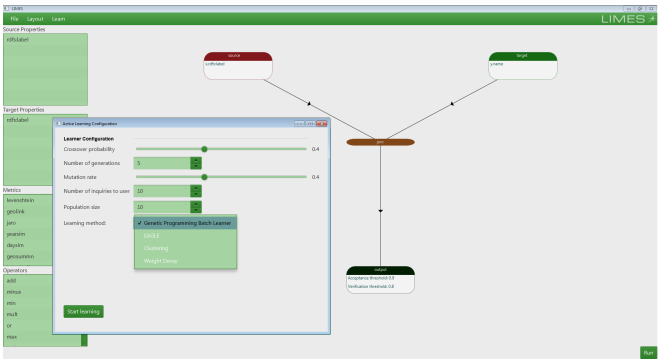

Figure 2. Selection of learning algorithm

The aim of this demo will be to show how the different algorithm implemented in LIMES (including EAGLE and COALA in its two variations "Clustering" and "Weight Decay") perform this deceptively simple machine-learning task. To this end, we will use both synthetic and real benchmark datasets as used in [7] as well as provide insights pertaining to the performance of the machine-learning framework on different datasets. The evaluation results of the algorithms presented can be found in the corresponding papers.

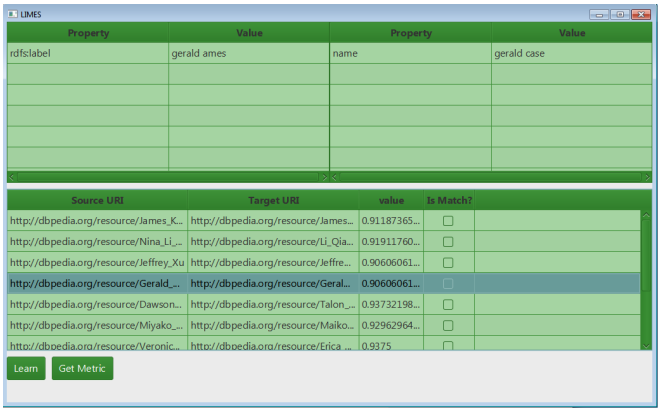

Figure 3. User feedback for active learning

3 CONCLUSIONS AND FUTURE WORK

We present the novel interface of the LIMES framework, one of the most used framework for Link Discovery of the Web of Data. We show how this interface allows using machine learning approaches of different types for link discovery. We also show that our machine learning approaches can indeed achieve good performance on the task at hand. We are constantly updating our machine learning algorithms and will keep on doing so in future works.

4 DEMONSTRATION REQUIREMENTS

LIMES is implemented in Java[5] and requires no special hardware. It runs on any operating system that supports Java 1.7. A standard network is required to gather data from SPARQL endpoints on the Web (e.g., `http://dbpedia.org/sparql`). A corresponding laptop will be brought to the venue. The only requirement of the authors is a WLAN access to the internet.

REFERENCES

[1] Kleanthi Georgala, Mohamed Sherif, and Axel-Cyrille Ngonga Ngomo, 'An efficient approach for the generation of allen relations', in *Proceedings of the European Conference on Artifical Intelligence*, (2016).

[2] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm, 'A survey of current link discovery frameworks', *Semantic Web*, (Preprint), 1–18, (2015).

[3] Axel-Cyrille Ngonga Ngomo, 'Link Discovery with Guaranteed Reduction Ratio in Affine Spaces with Minkowski Measures, booktitle = International Semantic Web Conference (1)', pp. 378–393, (2012).

[4] Axel-Cyrille Ngonga Ngomo, 'On link discovery using a hybrid approach', *Journal on Data Semantics*, **1**, 203 – 217, (December 2012).

[5] Axel-Cyrille Ngonga Ngomo and Klaus Lyko, 'Eagle: Efficient active learning of link specifications using genetic programming', in *Proceedings of ESWC*, (2012).

[6] Axel-Cyrille Ngonga Ngomo and Klaus Lyko, 'Unsupervised learning of link specifications: deterministic vs. non-deterministic', in *Proceedings of the Ontology Matching Workshop*, (2013).

[7] Axel-Cyrille Ngonga Ngomo, Klaus Lyko, and Victor Christen, 'Coala—correlation-aware active learning of link specifications', in *Proceedings of ESWC*, (2013).

[8] Andriy Nikolov, Mathieu D'Aquin, and Enrico Motta, 'Unsupervised Learning of Data Linking Configuration', in *Proceedings of ESWC*, (2012).

[9] Xing Niu, Shu Rong, Yunlong Zhang, and Haofen Wang, 'Zhishi.links results for oaei 2011', in *OM*, (2011).

[10] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov, 'Discovering and Maintaining Links on the Web of Data', in *ISWC*, pp. 650–665, (2009).

[11] Chuan Xiao, Wei Wang, Xuemin Lin, Jeffrey Xu Yu, and Guoren Wang, 'Efficient similarity joins for near-duplicate detection.', *ACM Trans. Database Syst.*, **36**(3), 15, (2011).

[5] `http://github.com/aksw/limes`