Supervised Ontology and Instance Matching with MELT

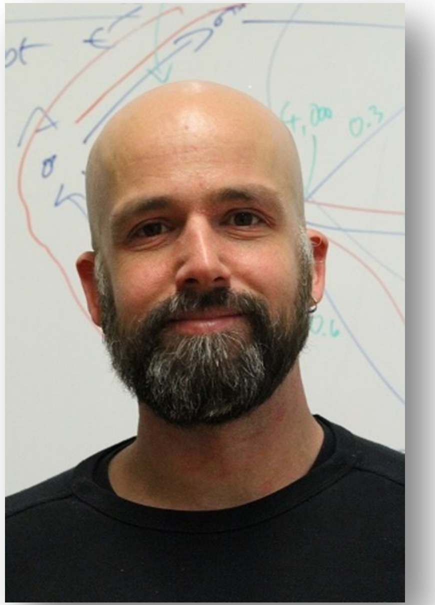
UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Joint Work

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Sven Hertling
Data and Web Science Group, University of Mannheim
sven@informatik.uni-mannheim.de

Jan Portisch
Data and Web Science Group, University of Mannheim / SAP SE
jan@informatik.uni-mannheim.de



Prof. Dr. Heiko Paulheim
Data and Web Science Group, University of Mannheim
heiko@informatik.uni-mannheim.de

Agenda

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

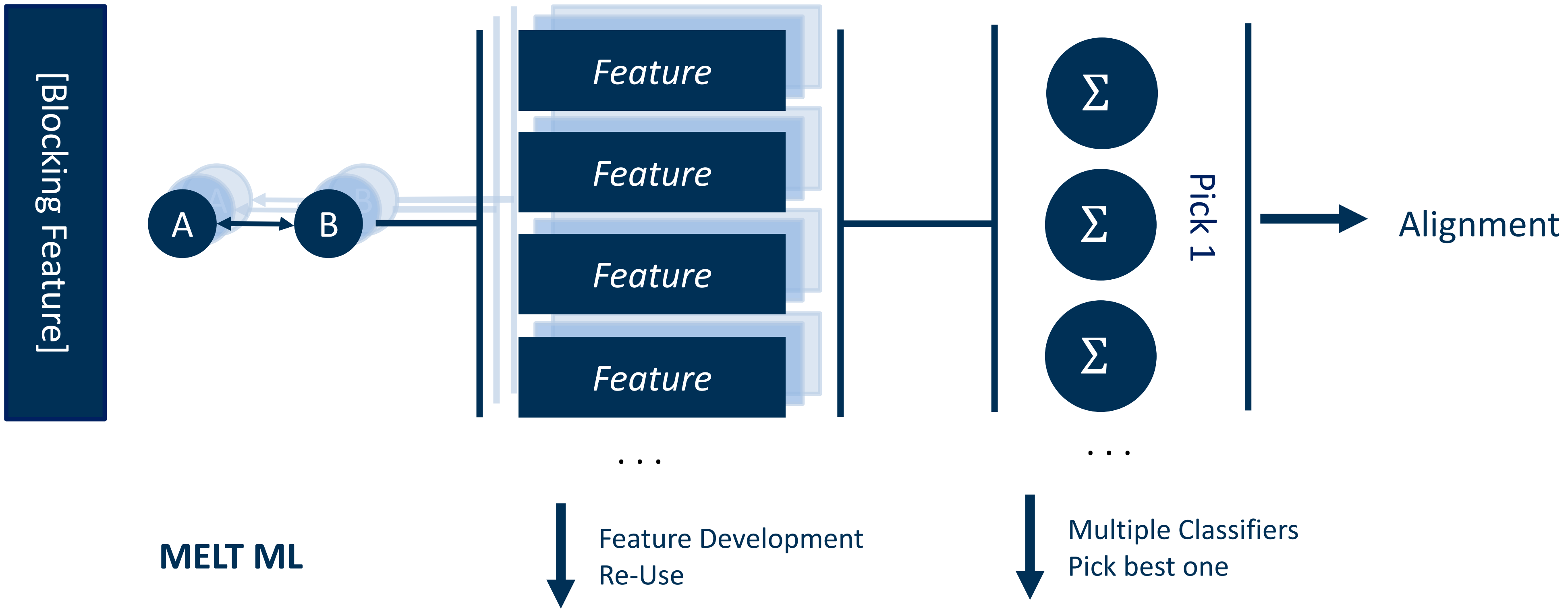
- Motivation
- What is MELT
- ML-Extension
- Quantitative Evaluation
- Q&A

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

MOTIVATION

Motivation

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics



Challenges in ML-based Matchers

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

- Python (or non-Java) libraries vs. SEALS environment
- Simple feature aggregation is not sufficient
- Many classifiers available
- Not much out-of-the-box ML functionality available for OM/IM
- No OAEI ML track

University of Mannheim
School of Business Informatics and Mathematics

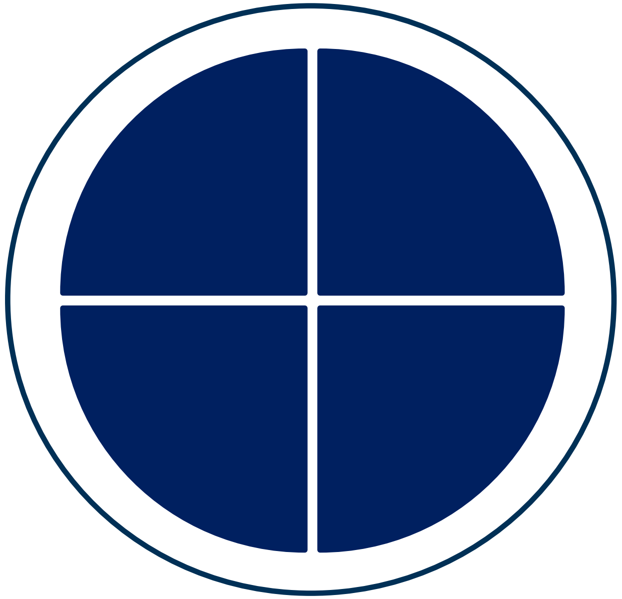
WHAT IS MELT?

What is MELT?

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

- **Easy** matcher development
- **Non-Java** matcher development
- **Maven** support

- Allow **for parameter optimization**



- Facilitate **matcher packaging**
- Facilitate **matcher submission**

- **Advanced evaluation** capabilities
- **Evaluation before packaging**
- Allow for **interactive visualization**

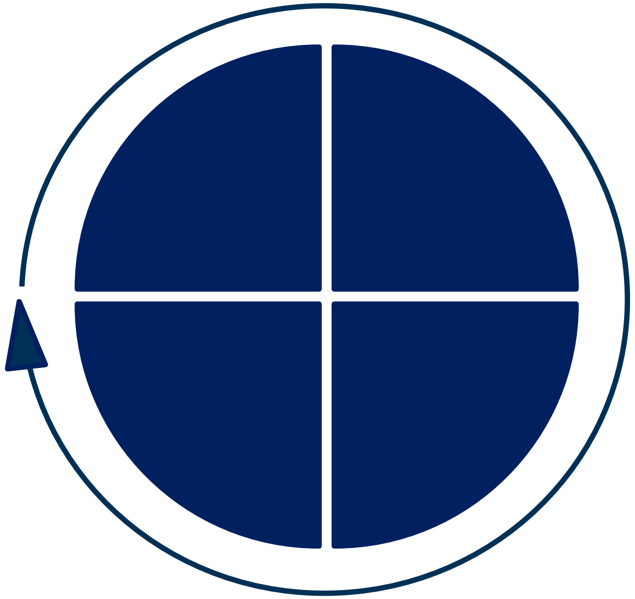
- **Streamlined** development process
- **Integration** with existing tooling
- **OAEI support**
- **Extensibility**

What is MELT?

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Matcher **Development** with **ML**

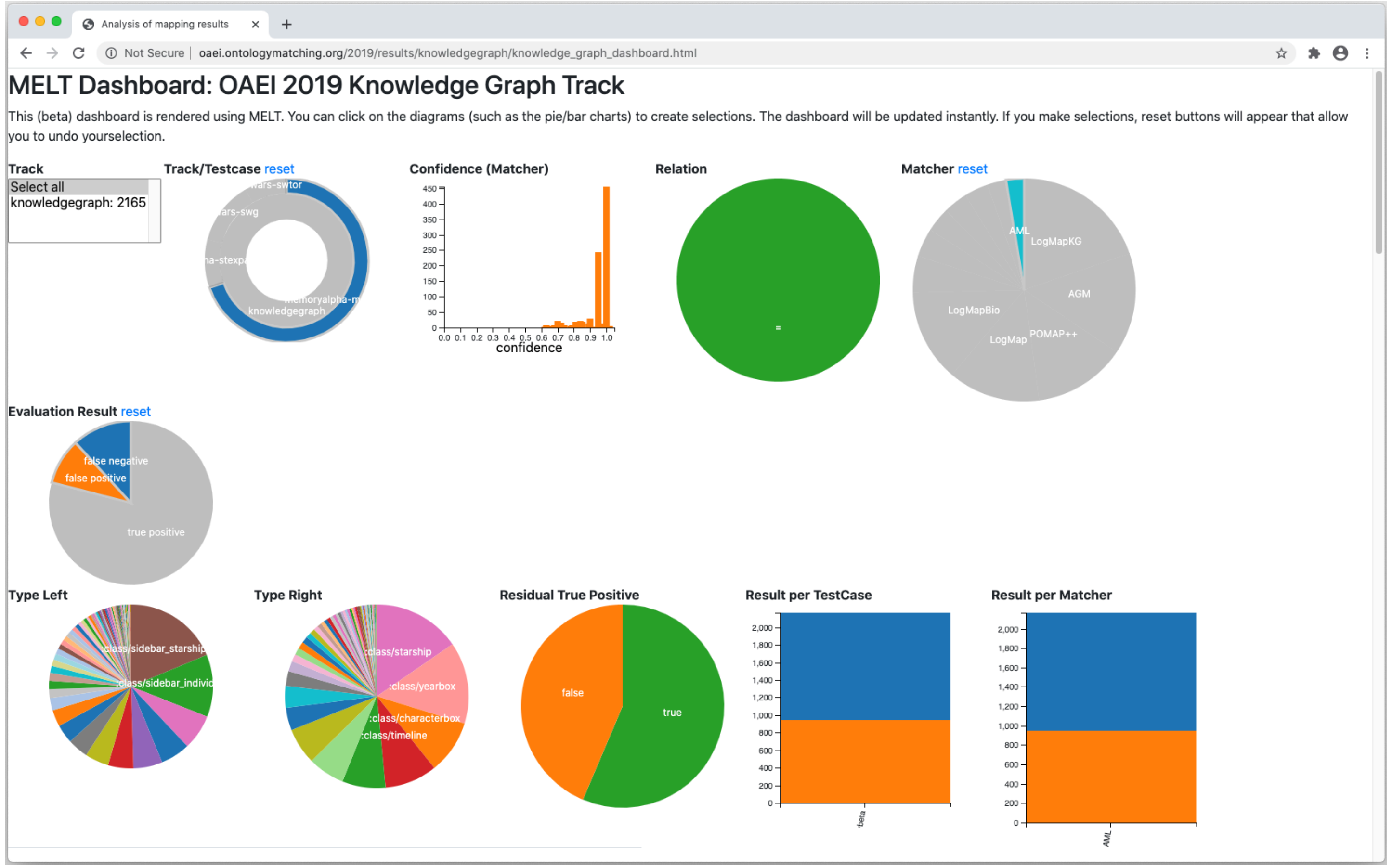
Matcher **Fine-Tuning**



Matcher **Submission**

Matcher **Evaluation**

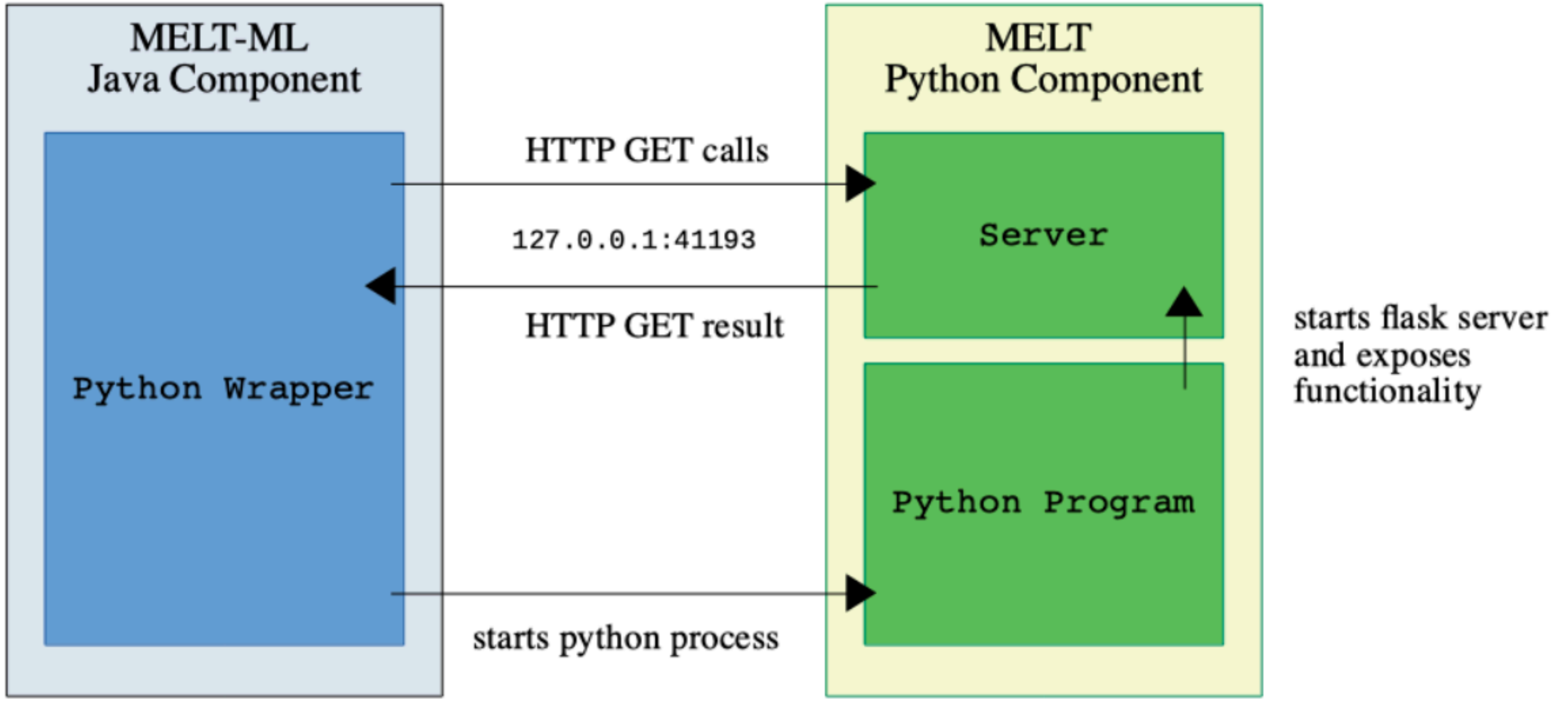
MELT Dashboard



ML-Extension

Python Code Execution in MELT

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics



Data for Training (Gold Standard)

- Any `Alignment` instance can be used as gold standard.
- New instance methods:
  - `sample(int n)`
    Samples *n* instances.
  - `sampleByFraction(double fraction)`
    Samples the specified percentage.

Feature Generation

- **`Matcher`**
  - Adds or removes correspondences from/to an alignment.
  - Can be used as partitioner.
- **`Filter`**
  - Adds or changes confidence values of existing correspondences.
  - Removes correspondences from an alignment.
  - Can also be thought of a feature generator for given correpondences.

21 Filters in MELT 2.6

17 Matchers in MELT 2.6

Feature Values in MELT

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

`Correspondence`

$$< E_1, E_2, R, C >$$

```
.addAdditionalExplanation(Class, String)

.addAdditionalConficence(Class, Double)
```

- Add an arbitrary number of confidences (using multiple filters, a confidence vector can be built).
- Possible to add explanations.
- Can be serialized in the Alignment Format (as simple correspondence extension).

Feature Generating Filter Examples

- `SimilarNeighboursFilter` (for instances)
  - analyzes for each instance correspondence how many “neighbors” (resources one predicate away) are already matched
  - multiple set similarities implemented (boolean, absolute, min, max, jaccard, dice)

- `CommonPropertiesFilter` (for instances)
  - intuition: equal instances share properties
  - default properties (`rdfs:label`) are excluded
  - multiple set similarities implemented (boolean, absolute, min, max, jaccard, dice)

Feature Generating Filter Examples

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

- `SimilarHierarchyFilter` (for instances)
  - computes similarity based on matched classes in the hierarchy of an instance
  - multiple possibilities to calculate the confidence based on hierarchy matches

Filter Examples: Machine Learning Scikit Filter

- `MachineLearningScikitFilter`
  - Ideally towards the end of a matching pipeline
  - Applies a five-fold cross validation (adjustable)…
  - … on multiple classifiers (see next slide)
  - Picks the best classifier
  - Trains a model and applies it automatically to the current alignment

Filter Examples:
Machine Learning Scikit Filter

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

- Classifiers (detailed configurations in the paper)
  - Decision Trees
  - Gradient Boosted Trees
  - Random Forest
  - Naïve Bayes
  - Support Vector Machines
  - Neural Netowrks

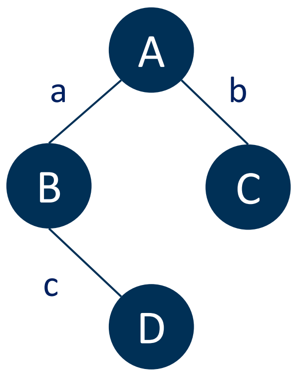
UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Quantitative Evaluation

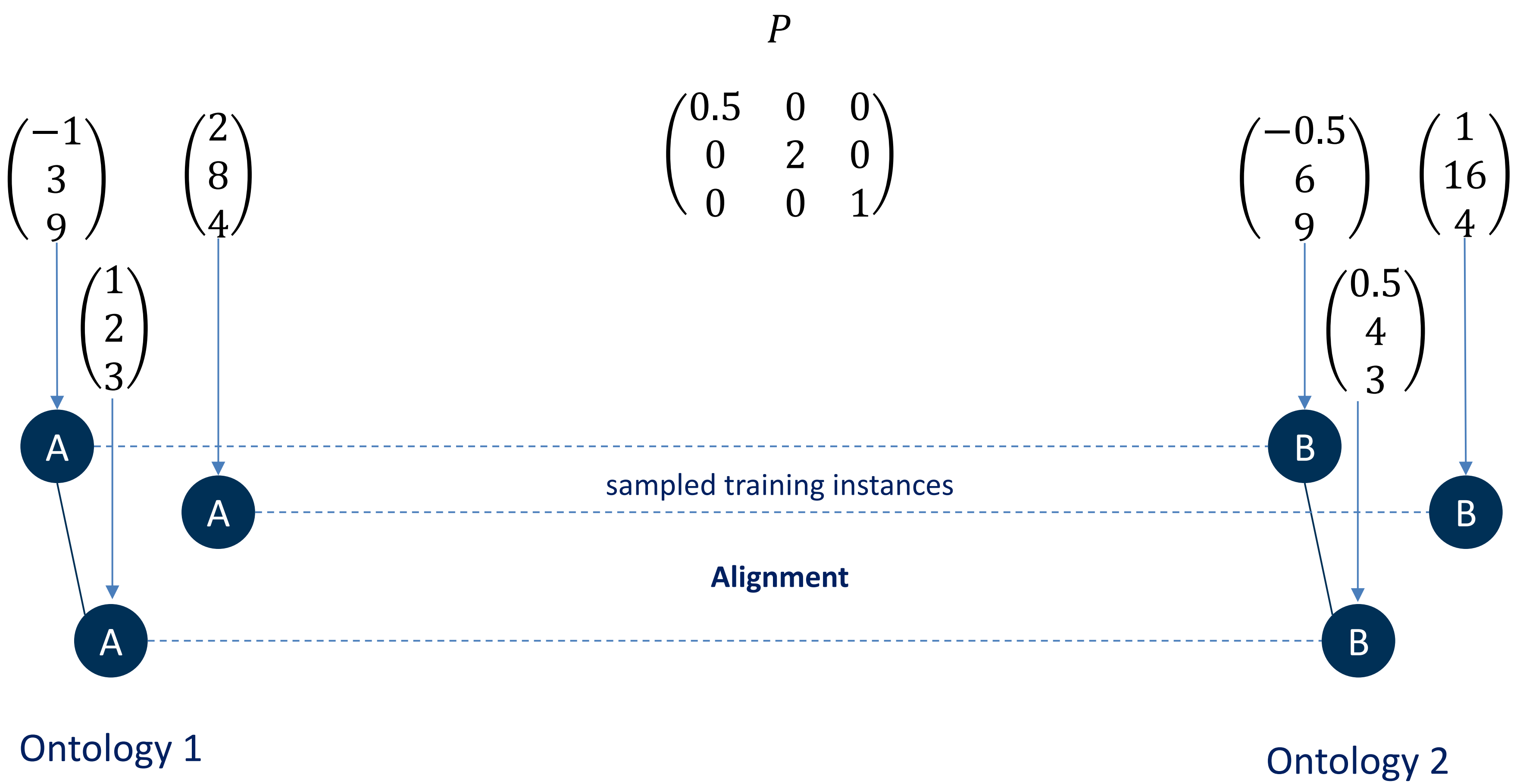
Use Case 1: RDF2Vec Vector Transformation
What is RDF2Vec?

University of Mannheim
School of Business Informatics and Mathematics

- Simple embedding approach for knowledge graphs.
- How does it work?
  - Generate random walks through the graph
    - sample walk
      ```
      A b C
      A a B c D
      ...
      ```
  - Apply word2vec
    - Sample result:
      ```
      A → (1, 2, 3)
      B → (2, 3, 4)
      a → (-0.5, 6, 12)
      ...
      ```



Use Case 1: RDF2Vec Vector Transformation



$$P = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 \\ 3 \\ 9 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 2 \\ 8 \\ 4 \end{pmatrix} \qquad \begin{pmatrix} -0.5 \\ 6 \\ 9 \end{pmatrix} \quad \begin{pmatrix} 0.5 \\ 4 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 16 \\ 4 \end{pmatrix}$$

Use Case 1: RDF2Vec Vector Projections

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

- RDF2Vec embeddings do contain structural information.
- Good results on same ontologies but bad performance in other cases.

Multifarm Test Case	P	R	R+	F	# of TP	# of FP	# of FN
iasted-iasted	0.8232	0.7459	0.6111	0.7836	135	29	46
conference-conference	0.7065	0.5285	0.1967	0.6047	65	27	58
confOf-confOf	0.9111	0.5541	0.1081	0.6891	41	4	33

Use Case 2: Supervised Classifier for the Knowledge Graph Track

- Base-Correspondences (recall oriented alignment): `BaseMatcher` on `rdfs:label` and `skos:altLabel`
- Filters (features)
  - `CommonPropertiesFilter`
  - `SimilarHierarchyFilter`
  - `BagOfWordsSetSimilarityFilter`
  - `SimilarNeighboursFilter`
  - `SimilarTypeFilter`
  - `MachineLearningScikitFilte`
  - `NaiveDescendingExtractor`

Use Case 2: Supervised Classifier for the Knowledge Graph Track

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

	mcu-marvel			memoryalpha-memorybeta			memoryalpha-stexpanded			starwars-swg			starwars-swtor		
Approach	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
BaseMatcher	0.8548	0.6796	0.7572	0.8740	0.8978	0.8858	0.8675	0.9264	0.8960	0.9001	0.7318	0.8072	0.9007	0.9146	0.9076
CommonPropertiesFilter	0.8823	0.6614	0.7560	0.9310	0.8785	0.9040	0.9370	0.8968	0.9165	0.9257	0.7162	0.8076	0.9371	0.8999	0.9181
SimilarHierarchyFilter	0.8823	0.6614	0.7560	0.9361	0.8830	0.9088	0.9527	0.9107	0.9312	0.9281	0.7181	0.8097	0.9440	0.9057	0.9245
BagOfWordsSetSimilarityFilter	0.8823	0.6614	0.7560	0.9340	0.8810	0.9067	0.9406	0.8991	0.9194	0.9292	0.7190	0.8107	0.9348	0.8976	0.9159
SimilarNeighboursFilter	0.8912	0.6687	0.7641	0.9467	0.8916	0.9183	0.9600	0.9171	0.9380	0.9375	0.7254	0.8179	0.9317	0.8947	0.9128
SimilarTypeFilter	0.8823	0.6614	0.7560	0.9247	0.8727	0.8980	0.9303	0.8899	0.9096	0.9222	0.7135	0.8045	0.9326	0.8962	0.9140
ML (sample=0.2)	0.8831	0.6620	0.7567	0.9636	0.8592	0.9084	0.9648	0.8887	0.9252	0.9292	0.7190	0.8107	0.9621	0.8778	0.9180
	SVM			Random Forest			SVM			SVM			Random Forest		
ML (sample=0.4)	0.8831	0.6620	0.7567	0.9636	0.8599	0.9088	0.9734	0.8690	0.9182	0.9315	0.7199	0.8121	0.9445	0.8903	0.9166
	Random Forest			Random Forest			Neural Network			Neural Network			Random Forest		
ML (sample=0.6)	0.8831	0.6620	0.7567	0.9685	0.8575	0.9096	0.9667	0.8916	0.9276	0.9367	0.7153	0.8112	0.9565	0.8903	0.9222
	Random Forest			Decision Tree			Neural Network			SVM			SVM		

There is MUCH more to MELT

Multi-Threaded Matcher Execution

Ontology **Caching** Services

Execution of SEALS packages from within MELT

TRY IT!

MELT @GitHub

Alignment **Extensions**

OAEI-Track Organizer Tools

Pre-Trained KG vectors through KGvec2go client

ExecutionResult **Indexing**

One-Time **Auto-Download** of OAEI Tracks

Matcher **Pipelining**

Thank you!

UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Sven Hertling
Data and Web Science Group, University of Mannheim
sven@informatik.uni-mannheim.de

Jan Portisch
Data and Web Science Group, University of Mannheim
jan@informatik.uni-mannheim.de

Heiko Paulheim
Data and Web Science Group, University of Mannheim
heiko@informatik.uni-mannheim.de

Preprint