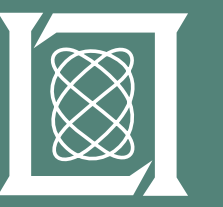


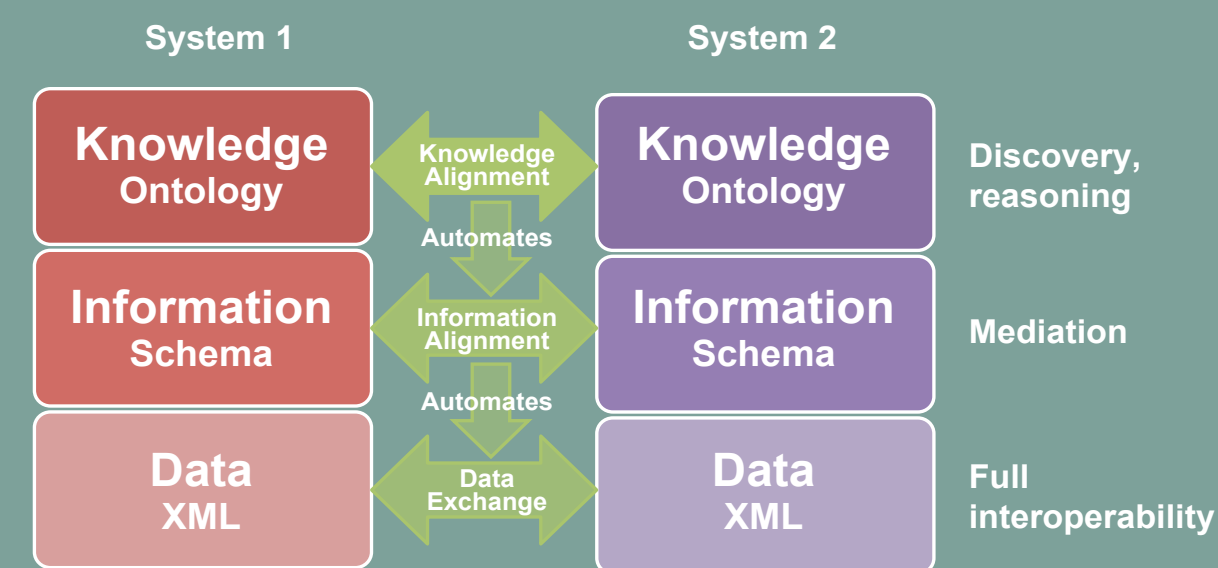
CompositeMatch: Detecting N-Ary Matches in Ontology Alignment



Kelly Moran, Kajal Claypool, and Benjamin Hescott MIT Lincoln Laboratory, Lexington, MA 02420

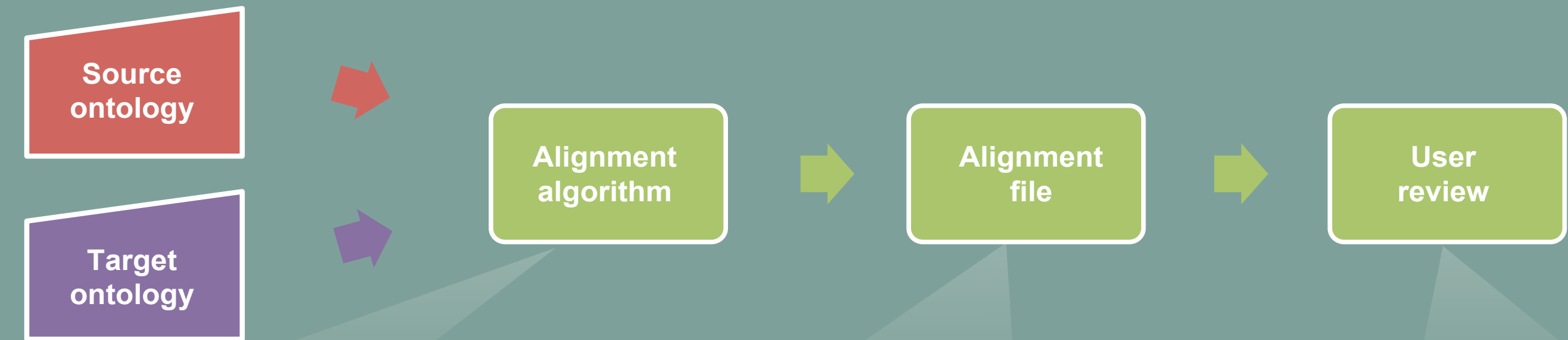
Introduction

- Goal
 - Improve accuracy of ontology alignment
 - Reduce user workload by more fully automating alignment process
 - Address a subset of the larger problem: identification of complete matches
- Motivation
 - Full systems interoperability requires alignment at all layers
 - Ontology alignment is a key component of information integration

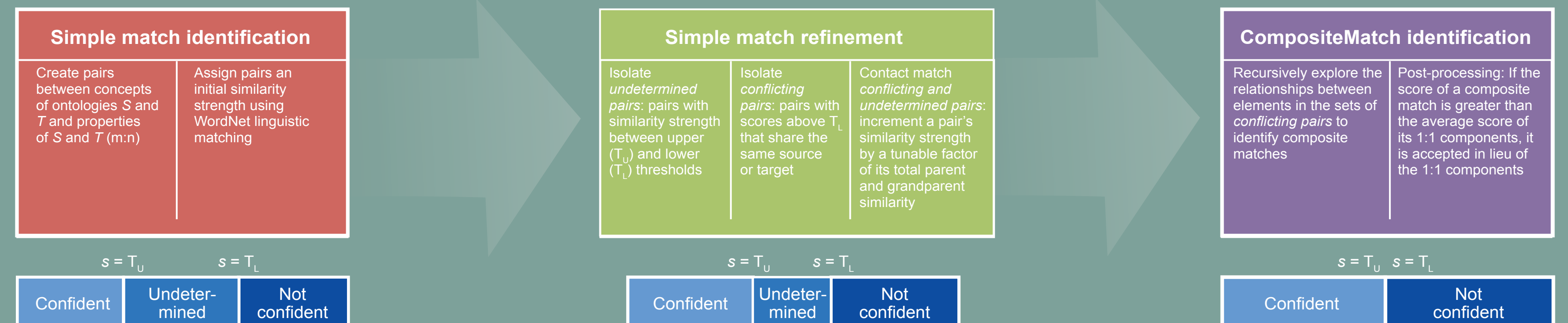


CompositeMatch Algorithm

- Overview
 - Takes in two ontologies and outputs an alignment file

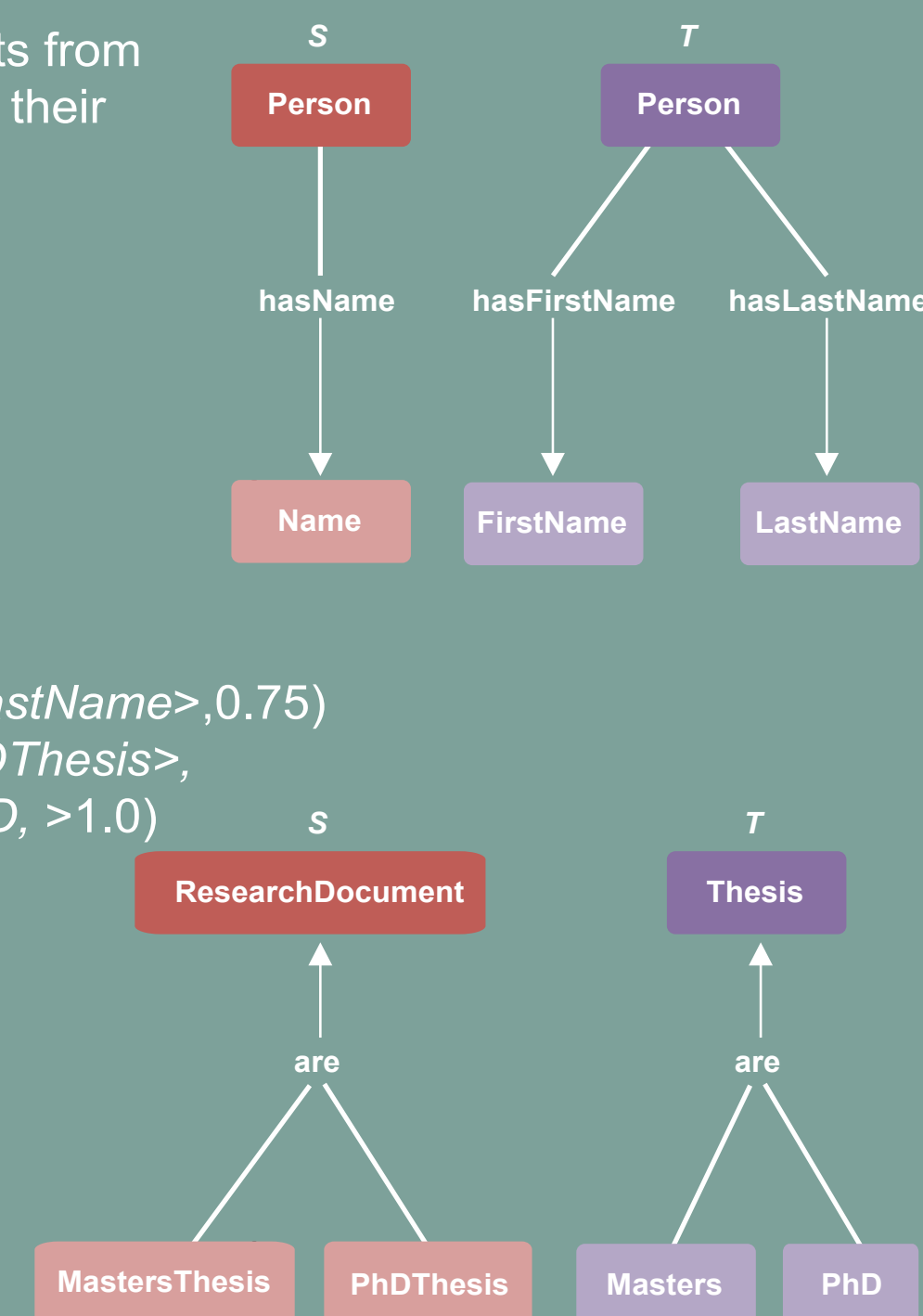


- 3-pass algorithm



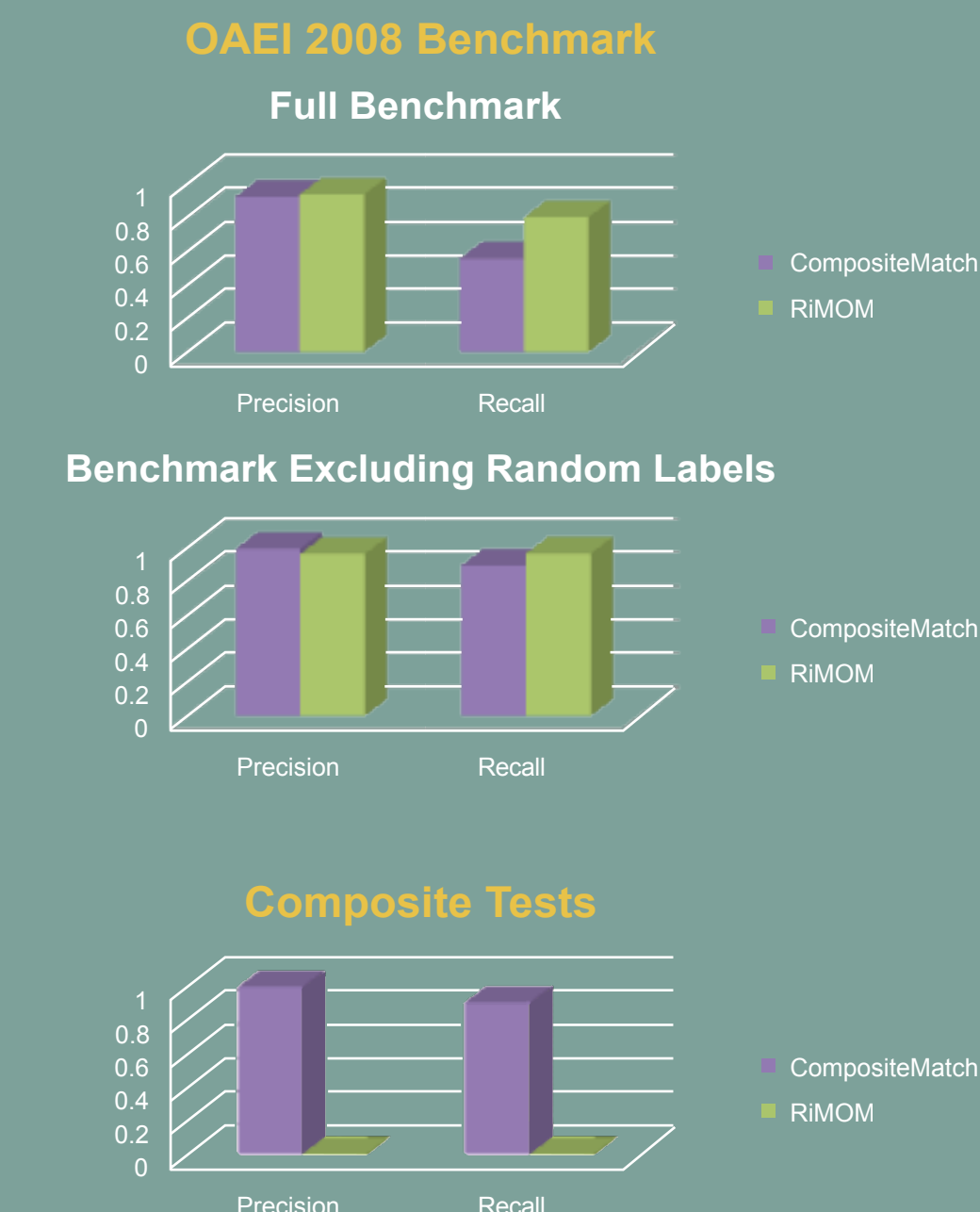
Types of Matches

- Simple matches
 - $m_s = (e, e', s)$ where e and e' are elements from ontologies S and T respectively, and s is their similarity strength
 - Ex.: $m_s = (Person, Person, 1.0)$
 - Handled by most algorithms
- Composite matches
 - $m_c = (C, C', s)$ where either C and C' is a composites of elements, or both
 - Ex.: $(1:n) := m_c = (Name, \langle FirstName, LastName \rangle, 0.75)$
 - Ex.: $(m:n) := m_c = (\langle MastersThesis, PhDThesis \rangle, \langle Thesis, Masters, PhD \rangle, >1.0)$
 - Arise from structural dissimilarities between ontologies
 - Rarely detected by even top-performing algorithms

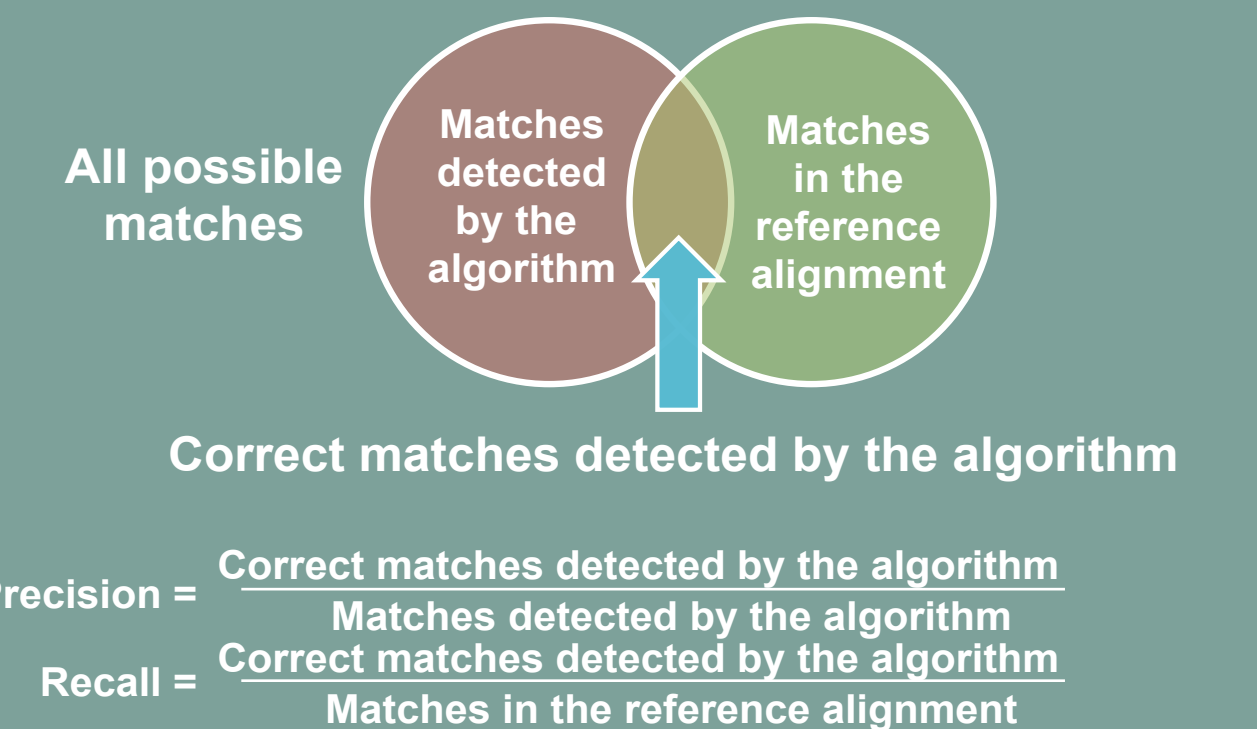


Evaluation

Performance of CompositeMatch and RiMOM



- Composite tests
 - Compare OAEI benchmark test to modifications intended to introduce composite matches
- Scored against hand-modified reference alignment



Conclusion

- 1:1 match detection alone is insufficient
- Absence of composite match detection results in missing matches or incorrect 1:1 matches
- Inclusion of composite matches is effective for increasing accuracy and reducing user workload
- Can aid in real-world applications where ontologies differ structurally
- Ultimately make up a portion of techniques to accurately integrate information or share knowledge