

Ontology Alignment with Near-Miss Clone Detection

Paul L. Geesaman, James R. Cordy, Amal Zouaq



Introduction

The Web Ontology Language (OWL) is designed for knowledge representation on the Semantic Web. A large number of experts create different ontologies within the same domain, which causes heterogeneous data with overlapping information. Identifying two ontologies and their components – classes, properties, and instances – as being similar or identical is the task of ontology alignment. The goal is to allow for the cross-referencing of the aforementioned components of ontologies in the larger system of the Semantic Web. By analyzing OWL source code, we propose using near-miss clone detection in the task of ontology alignment may provide beneficial results and overcome some of the problems faced by other ontology alignment algorithms.

Method

Near-miss clone detection is useful in finding source code that is copied from one source to another[4]. NiCad is the tool used for clone detection by this system. Each line of a potential clone is compared to the line of another potential clone. If the percentage of lines that are identical are below a set threshold, then the potential clones are said to be similar. Potential clones are clustered together and labeled classes. Analyzing code fragments as a string allows for the recognition of clones despite the presence of missing fragments – a task that proves to be very difficult when dealing with a graph. Near-miss clone detection appears to be applicable to OWL files. The first step of near-miss clone detection is the extraction stage: all relevant lines of code, in this case class elements, are put into a new file. After extraction, normalization is performed to the ontologies so that the clone detector may yield better results. The most important normalization technique so far has been inlining. Whenever a class is referenced in the ontology, this reference is replaced by the class description and the class axioms. Currently the axioms only include subClassOf statements, but it planned to expand the extraction to cover all OWL constructs. Below is an example of inlining:

```
<owl:Class rdf:ID="Beer">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:Class rdf:ID="Ale">
  <rdfs:subClassOf rdf:resource="Beer"/>
</owl:Class>
```

is normalized into:

```
<owl:Class rdf:ID="Beer">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:Class rdf:ID="Ale">
  <owl:Class rdf:ID="Beer">
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  </owl:Class>
</owl:Class>
```

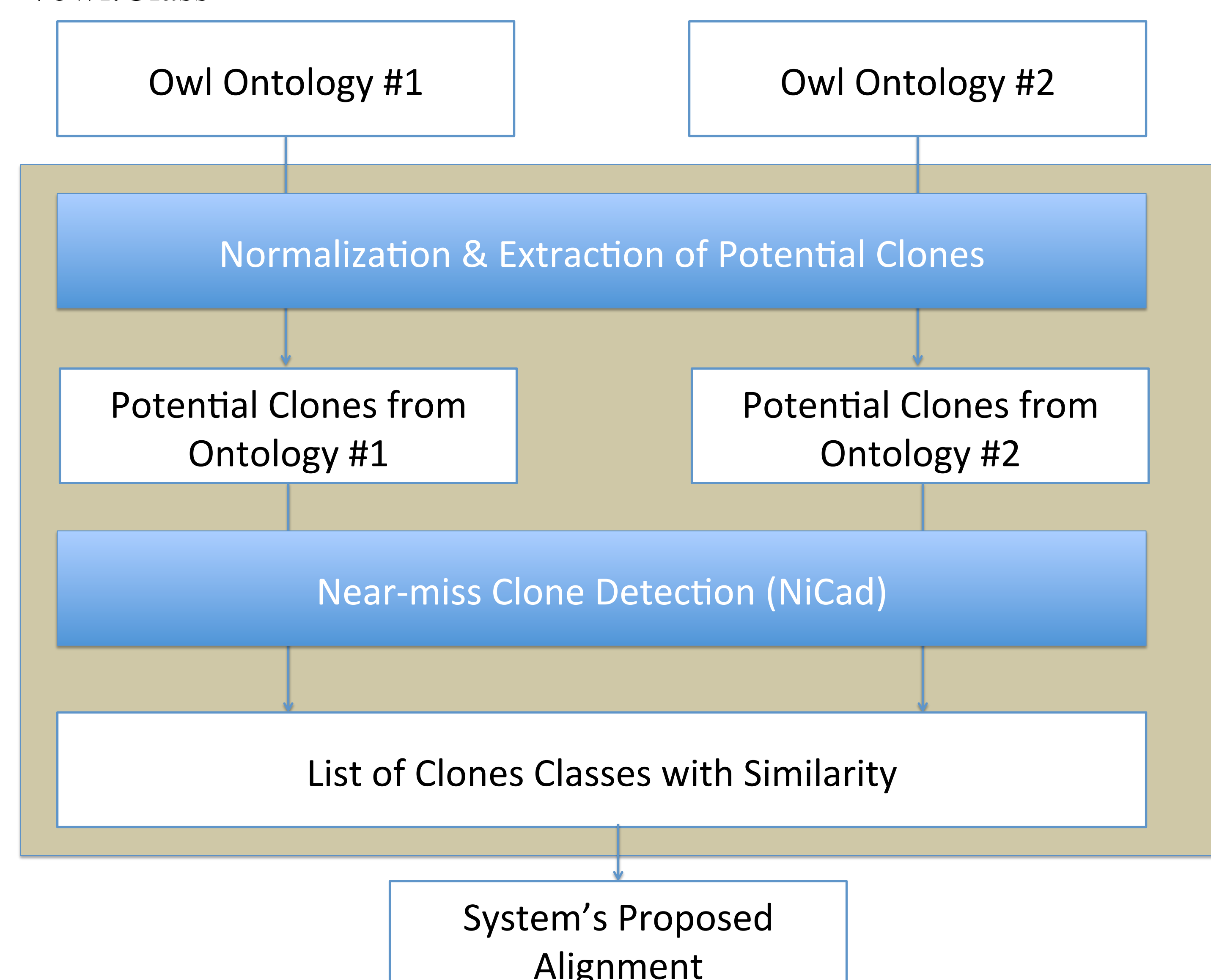


Fig. 1 System's Architecture

Evaluation

The evaluation is based on categorizing components of ontologies as being identical, one component being a superset of the other, and ontology components as being incompatible [2]. Evaluation is based on ontologies provided by the Ontology Alignment Evaluation Initiative website[1]. The gold standard is provided by the initiative. The effectiveness of our tool will be measure by precision, recall, and f-measure.

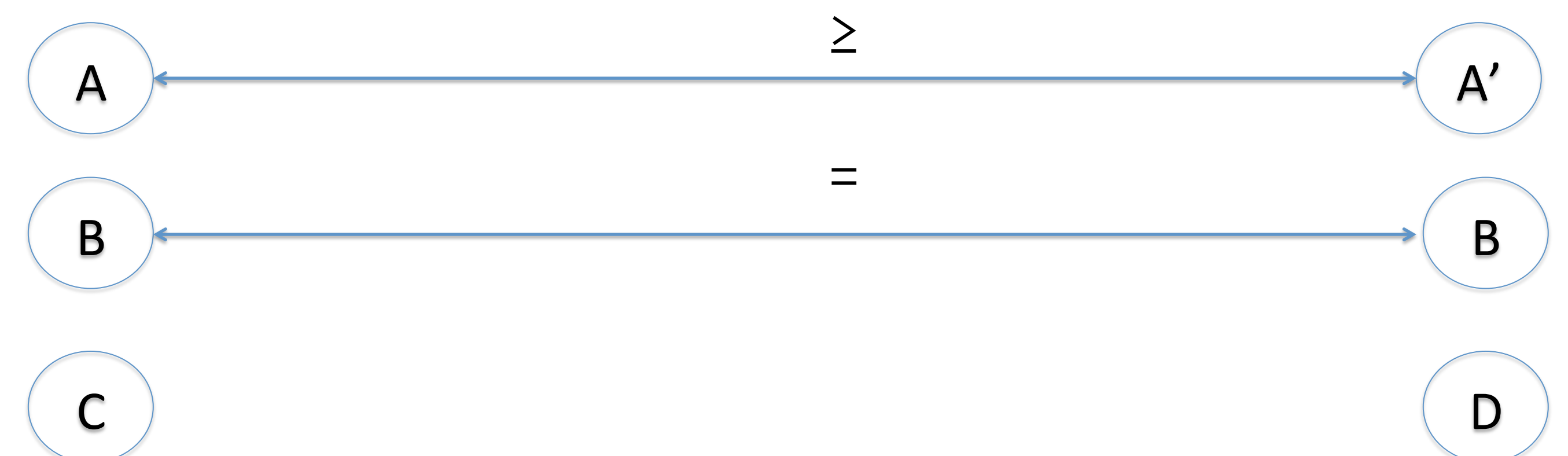


Fig. 2 Example of Ontology Alignment

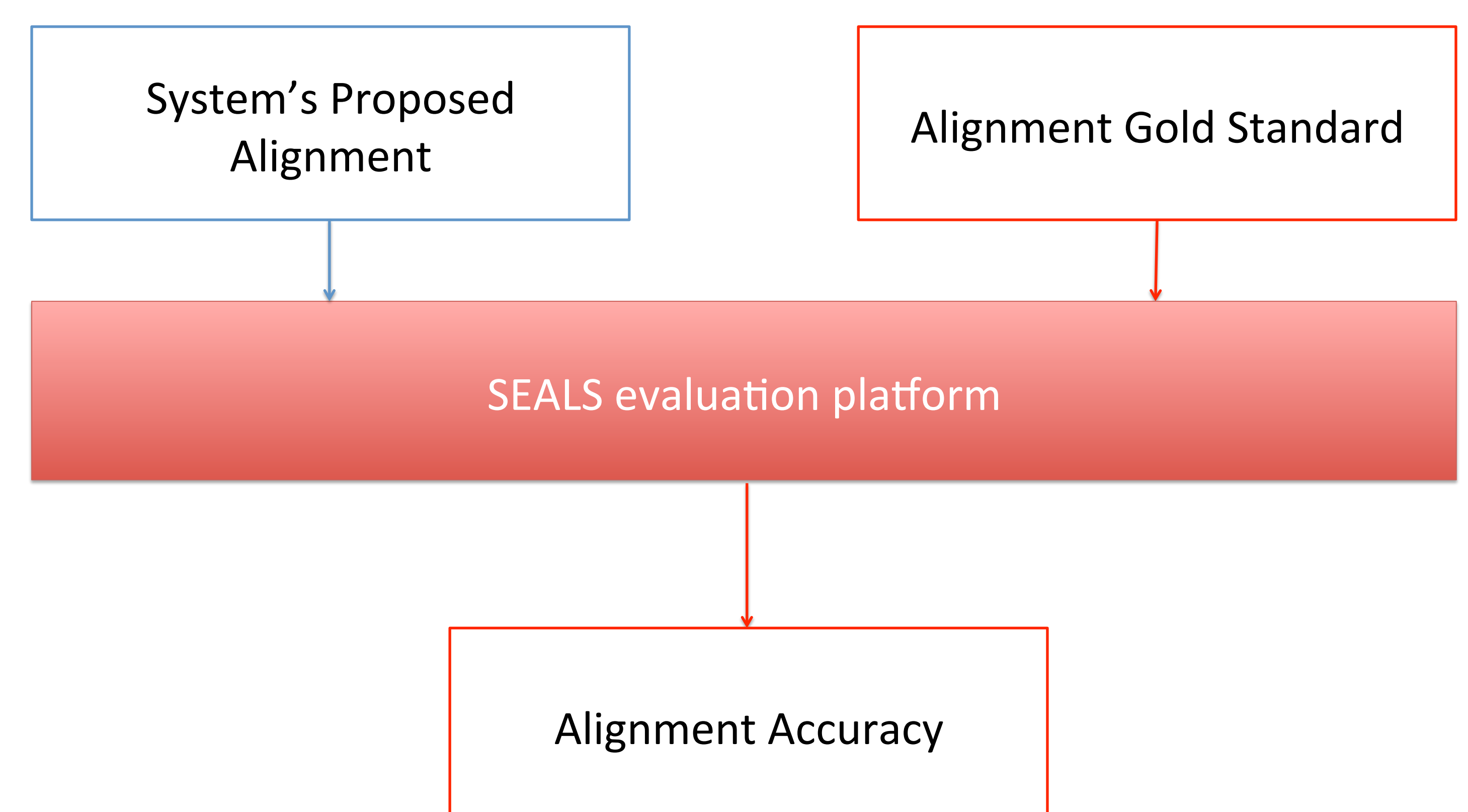


Fig. 3 Alignment Evaluation

Preliminary Results

Our system has had success in identifying clones in some of the ontology tests that have proven to be difficult for other ontology alignment algorithms. For example, we have had success in comparing ontologies which describe the same domain in different languages.

As it stands, precision is very high. As we change the parameters of NiCad, recall and f-measure will increase. F-measure is the parameter by which we use to measure a normalization's effectiveness.

Future Work

Before this research is completed, we have the following steps to implement:

- Further normalization so that NiCad may obtain better results.
- Include properties and instances of an ontology for comparison.
- Remove redundant lines of xml in the hopes of increasing recall.
- Generalize results from the ontology alignment evaluation initiative and perform tests on real-world ontologies.

References

- [1] Ontology alignment evaluation initiative - 2011 campaign. <http://oaei.ontologymatching.org/2011/>.
- [2] Jérôme Euzenat and Pavel Shvaiko. Ontology matching tutorial. <http://sssw.org/2009/slides/SSSW09-JE-slides.pdf>.
- [3] Jérôme Euzenat and Pavel Shvaiko. Ontology matching. Springer, 2007.
- [4] C.K. Roy and J.R. Cordy. Nicad: Accurate detection of near-miss intentional clones using flexible pretty-printing and code normalization. In Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on, pages 172 {181, june 2008.