# ObjectCoref & Falcon-AO: Results for OAEI 2010

Wei Hu, Jianfeng Chen, Gong Cheng, and Yuzhong Qu

[1] Department of Computer Science and Technology, Nanjing University, China
[2] State Key Laboratory for Novel Software Technology, Nanjing University, China
{whu, yzqu}@nju.edu.cn

**Abstract.** In this report, we mainly present an overview of ObjectCoref, which follows a self-training framework to resolve object coreference on the Semantic Web. Besides, we show preliminary results of Falcon-AO (2010) for this year's OAEI campaign, including the benchmark and conference tracks.

## 1   Presentation of the system

### 1.1   State, purpose, general statement

The Semantic Web is an ongoing effort by the W3C Semantic Web Activity to actualize data integration and sharing across different applications and organizations. To date, a number of prominent ontologies have emerged to publish data for specific domains, such as the Friend of a Friend (FOAF). These specifications recommend common identifiers for classes and properties in the form of *URIs* [1] that are widely and consistently used across data sources.

On the instance level, however, it is far from achieving agreement among sources on the use of common URIs to identify specific *objects* on the Semantic Web. In fact, due to the decentralized and dynamic nature of the Semantic Web, it frequently happens that different URIs from various sources, more likely originating from different RDF documents, are used to identify the same real-world object, i.e., refer to an identical thing (as known as URI aliases [5]). Examples exist in the domains of people, academic publications, encyclopedic or geographical resources.

*Object coreference resolution*, also called consolidation or identification [2], is a process for identifying multiple URIs of the same real-world object, that is, determining URI aliases (called coreferent URIs in this report) that denote a unique object. At present, object coreference resolution is recognized to be useful for data-centric applications, e.g. heterogeneous data integration or mining systems, semantic search, query and browsing engines.

We introduce a new approach, *ObjectCoref*, for bootstrapping object coreference resolution on the Semantic Web. The architecture of the proposed approach follows a common self-training framework (see Fig. 1). Self-training [6] is a major kind of semi-supervised learning, which assumes that there are abundant unlabeled examples in the real world, but the number of labeled training examples is limited. We believe that self-training is an appropriate way for resolving object coreference on the Semantic Web.

*Falcon-AO* [4] is an automatic ontology matching system with acceptable to good performance and a number of remarkable features. It is written in Java, and is open

source. ObjectCoref and Falcon-AO together help better enable interoperability between applications that use heterogeneous Semantic Web data.
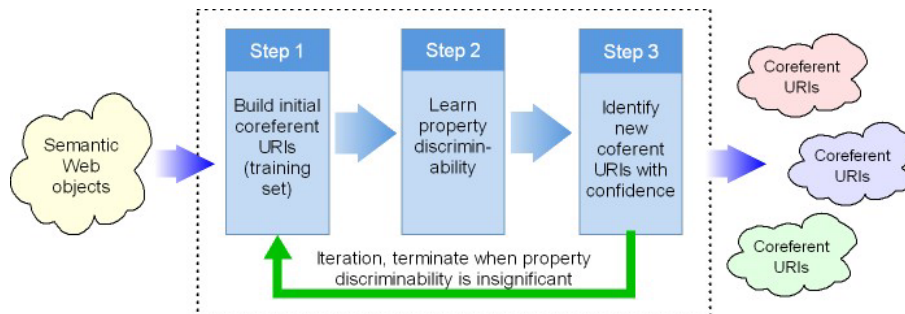


**Fig. 1.** Self-training process

## 1.2 Specific techniques used

ObjectCoref builds an initial set of coreferent URIs mandated by the formal and explicit semantics of owl:sameAs, owl:InverseFunctionalProperty, owl:FunctionalProperty, owl:cardinality and owl:maxCardinality.

The semantics of owl:sameAs dictates that all the URIs linked with this property have the same identity; if a property is declared to be inverse functional (IFP), then the object of each property statement uniquely determines the subject (some individual); a functional property (FP) is a property that can have only one unique value for each object; while cardinality (or max-cardinality) allows the specification of exactly (or at most) the number of elements in a relation, in the context of a particular class description, and when the number equals 1, it is somehow similar to the FP, but only applied to this particular class.

Next, ObjectCoref learns the discriminability of pairs of properties based on the coreferent URIs, in order to find more coreferent URIs for extending the training set. The discriminability reflects how well each pair of properties can be used to determine whether two URIs are coreferent or not. As an extreme example, IFPs (e.g. foaf:mbox) have a very good discriminability.

In RDF graphs, each URI is involved in a number of RDF triples whose subject is the URI, and the predicates and objects in these RDF triples form some <property, value> pairs, which can be considered as features for describing such URI. ObjectCoref compares the values between the <property, value> pairs from coreferent URIs, and finds which two properties have similar values and how frequent. The significance is the percentage of the number of coreferent URIs that can found by the discriminant properties in all the coreferent URIs in the training set. If the significance is greater than a given threshold, such the property pair is chosen for further resolution. Please note that for different domains, same property pairs may have different discriminability.

For example, a pair of rdfs:labels is discriminant for the biomedical domain but not for people.

If new coreferent URIs are found, ObjectCoref selects highly accurate ones and adds them into the training set. The whole process iterates several times and terminates when the property discriminability is not significant enough or cannot find more discriminant property pairs.

### 1.3 Adaptations made for the evaluation

For ObjectCoref, there is no explicit equivalence semantics in the DI and PR tracks. In order to establish the initial training set of coreferent URIs, we randomly extract 20 mappings from the reference alignment for each test case. All the mappings generated by ObjectCoref are based on the same parameters.

For Falcon-AO, we do not make any specific adaptation in the OAEI 2010 campaign. All the mappings for the benchmark and conference tracks outputted by Falcon-AO are uniformly based on the same parameters.

### 1.4 Link to the system and parameters file

We implement an online service for ObjectCoref, and run it over a large-scale dataset collected by the Falcons [3] search engine up to Sept. 2008. The dataset consists of nearly 600 million RDF triples describing over 76 million URIs. It is still under development. Please visit: `http://ws.nju.edu.cn/objectcoref`.

Besides, we follow the SEALS platform to publish Falcon-AO (2010) as a service. Please access it from `http://219.219.116.154:8083/falconWS?wsdl`. The offline version can be downloaded from our website: `http://ws.nju.edu.cn/falcon-ao`.

### 1.5 Link to the set of provided alignments (in align format)

The alignments for this year's OAEI campaign should be available at the official website: `http://oaei.ontologymatching.org/2010/`.

## 2 Results

In this section, we will present the results of ObjectCoref and Falcon-AO (2010) on the tracks provided by the OAEI 2010 campaign.

### 2.1 DI

In this track, we use ObjectCoref to resolve object coreference between three pairs of datasets, namely diseasome vs. sider, dailymed vs. sider and drugbank vs. sider. Table 1 shows the discriminant property pairs that ObjectCoref learns by self-training. For example, diseasome:name and sider:siderEffectName are a pair of discriminant properties, and if some URI in the diseasome dataset has a value w.r.t. diseasome:name that

is similar to some URI in the sider dataset w.r.t. sider:siderEffectName, these two URIs can be considered as coreferent. In this track, the training process converges at two iterations, respectively.

**Table 1.** Property discriminability on the DI track

|  | Property in dataset1 | Property in dataset2 |
|---|---|---|
| diseasome vs. sider | rdfs:label | sider:sideEffectName |
|  | diseasome:name | sider:siderEffectName |
|  | rdfs:label | rdfs:label |
|  | diseasome:name | rdfs:label |
| dailymed vs. sider | dailymed:genericMedicine | sider:drugName |
|  | dailymed:name | sider:drugName |
|  | dailymed:genericMedicine | rdfs:label |
|  | dailymed:name | rdfs:label |
| drugbank vs. sider | drugbank:genericName | sider:drugName |
|  | rdfs:label | sider:drugName |
|  | drugbank:genericName | rdfs:label |
|  | rdfs:label | rdfs:label |
|  | drugbank:synonym | sider:drugName |
|  | drugbank:synonym | rdfs:label |
|  | drugbank:pubchemCompoundId | sider:siderDrugId |
|  | drugbank:brandName | sider:drugName |

With these discriminant property pairs, ObjectCoref finds a number of coreferent URIs for each pair of datasets. As shown in Table 2, the precision and recall is moderate. Without considering the type of each object, the precision is not very good, so further inference-based debugging on coreferent URIs is needed for future work.

**Table 2.** Performance of ObjectCoref on the DI track

|  | Found | Existing | Precision | Recall | F-measure |
|---|---|---|---|---|---|
| diseasome vs. sider | 190 | 238 | 0.837 | 0.668 | 0.743 |
| dailymed vs. sider | 2903 | 1592 | 0.548 | 0.999 | 0.708 |
| drugbank vs. sider | 933 | 283 | 0.302 | 0.996 | 0.464 |

### 2.2 PR

In this track, ObjectCoref uses the same self-training process to recognize coreferent URIs for each pair of datasets, two of which are related to persons and the other is about restaurants. The discriminant property pairs are listed in Table 3. Based on these discriminant properties, ObjectCoref finds a set of coreferent URIs, where the precision and recall are pretty good (see Table 4). In particular, the good recall reflects that our

learning approach identifies the key properties for resolving object coreference in this track. But we also notice that some combination of properties may be also helpful. For example, first_name + last_name can be used for identifying same people.

**Table 3.** Property discriminability on the PR track

|  | Property in dataset1 | Property in dataset2 |
|---|---|---|
| person1 | person11:has_address | person12:has_address |
|  | person11:phone_number | person12:phone_number |
|  | person11:soc_sec_id | person12:soc_sec_id |
| person2 | person21:has_address | person22:has_address |
|  | person21:phone_number | person22:phone_number |
|  | person21:soc_sec_id | person22:soc_sec_id |
| restaurants | restaurant1:has_address | restaurant2:has_address |
|  | restaurant1:name | restaurant2:name |

**Table 4.** Performance of ObjectCoref on the PR track

|  | Found | Existing | Precision | Recall | F-measure |
|---|---|---|---|---|---|
| person1 | 499 | 500 | 1.000 | 0.998 | 0.999 |
| person2 | 362 | 400 | 1.000 | 0.900 | 0.947 |
| restaurants | 91 | 112 | 0.989 | 0.804 | 0.887 |

### 2.3 Benchmark & conference

We use Falcon-AO (2010) to participate in the benchmark and conference tracks. The average precision and recall are depicted in Table 5. As compared to OAEI 2007, the benchmark track adds some new cases. Falcon-AO failed in several cases due to the Jena parsing errors. For the detailed results, please see Appendix.

**Table 5.** Performance of Falcon-AO (2010) on the benchmark and conference tracks

|  | Precision | Recall |
|---|---|---|
| Benchmark | 0.76 | 0.64 |
| Conference | 0.60 | 0.60 |

## 3 General comments

In this section, we will firstly discuss several possible ways to improve ObjectCoref, and then give comments on the OAEI 2010 test cases.

### 3.1 Discussions on the way to improve the proposed system

The preliminary results of ObjectCoref demonstrate that using property discriminability is feasible to find coreferent URIs on the Semantic Web. However, we also see several shortcomings of the proposed approach, which will be considered in the next version.

1. *How to divide objects into different domains?* For the tasks in this year's OAEI, we may not see the importance of recognizing domains, but on the whole Semantic Web, different domains may have different discriminant properties, and a single property pair may have different discriminability in different domains. So, a uniform measurement is ineffective.

2. *How to avoid error accumulation?* In self-training, an important issue is to prevent error accumulation, since a wrong labeled example would lead to misclassification in further propagation. In our evaluation, because the training process converges in a few iterations, so this situation is not so significant. But in real world, it is imperative to consider that.

3. *How to find discriminant property combinations?* A single property may be not good enough for resolving object coreference, while the combination of several properties would be more discriminant. However, we need to avoid overfitting. So, we plan to mine frequent patterns in the RDF data for describing objects and refine these frequent patterns to form property combinations.

### 3.2 Comments on the OAEI 2010 test cases

The proposed matching tasks cover a large portion of real world domains, and the discrepancies between them are significant. Doing experiments on these tasks are helpful to improve algorithms and systems. In order to enhance applicability, we list some problems in our experiment procedure, which might aid organizers to improve in the future.

1. In the DI track, the organizers provide 4 downloadable datasets for the biomedical domain, however, the interlinking track also involves a number of others, e.g., linkedct, lifescience, bio2rdf. The datasets are not only very large, but also difficult to find the latest versions, most of which are even not allowed to download. Furthermore, using SPARQL endpoints in the experiment is very time-consuming, especially for such a large scale. So, we would expect that all the datasets can be (perhaps temporarily) offline in the next year.

2. Falcon-AO (2010) uses Jena 2.6.3 as the RDF parser. In the benchmark track, some ontologies may have problems and cause the Jena exception "Unqualified typed nodes are not allowed. Type treated as a relative URI". So, we would expect the organizers to fix this in the next year.

## 4 Conclusion

Object coreference resolution is an important way for establishing interoperability among (Semantic) Web applications that use heterogenous data. We implement an online system for resolving object coreference called ObjectCoref, which follows a self-training framework focusing on learning property discriminability. From the experiments in this year's DI and PR tracks, we find some positive and negative experience for improving our system. In the near future, we look forward to making a stable progress towards building a comprehensive object coreference resolution system for the Semantic Web.

## Acknowledgements

## References

1. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifier (URI): Generic Syntax. RFC 2396 http://www.ietf.org/rfc/rfc3986.txt
2. Bleiholder, J., Naumann, F.: Data Fusion. ACM Computing Surveys 41(1), 1–41 (2008)
3. Cheng, G., Qu, Y.: Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. International Journal on Semantic Web and Information Systems 5(3): 49–70 (2009)
4. Hu, W., Qu, Y.: Falcon-AO: A Practical Ontology Matching System. Journal of Web Semantics 6(3), 237–239 (2008)
5. Jacobs, I., Walsh, N.: Architecture of the World Wide Web, Volume One. W3C Recommendation 15 December 2004. http://www.w3.org/TR/webarch/
6. Zhou, Z., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems 24(3), 415–439 (2009)

## Appendix: Complete results

In this appendix, we will show the complete results of Falcon-AO (2010) on the *benchmark* and *conference* tracks. Tests were carried out on two Intel Xeon Quad 2.40GHz CPUs, 8GB memory with Redhat Linux Enterprize Server 5.4 (x64), Java 6 compiler and MySQL 5.0.

### Matrix of Results

In the following tables, the results are shown by precision (Prec.) and recall (Rec.).

| Bench | Description | Prec. | Rec. |
|---|---|---|---|
| 101 | Reference | 1.00 | 1.00 |
| 102 | Irrelevant | NaN | NaN |
| 103 | Lang. generalization | 1.00 | 1.00 |
| 104 | Lang. Restriction | 1.00 | 1.00 |
| 201 | No names | 0.97 | 0.97 |
| 201-2 | | 0.98 | 0.98 |
| 201-4 | | 1.00 | 1.00 |
| 201-6 | | 0.92 | 0.92 |
| 201-8 | | 0.98 | 0.98 |
| 202 | No names, comments | Jena error | |
| 202-2 | | 0.70 | 0.70 |
| 202-4 | | 0.70 | 0.70 |
| 202-6 | | 0.72 | 0.72 |
| 202-8 | | 0.72 | 0.72 |
| 203 | Misspelling | 1.00 | 1.00 |
| 204 | Naming conventions | 0.96 | 0.96 |
| 205 | Synonyms | 0.97 | 0.97 |
| 206 | Translation | 0.94 | 0.94 |
| 207 | | 0.96 | 0.96 |
| 208 | | 0.98 | 0.98 |
| 209 | | 0.65 | 0.65 |
| 210 | | 0.68 | 0.68 |
| 221 | No specialization | 1.00 | 1.00 |
| 222 | Flattened hierarchy | 1.00 | 1.00 |
| 223 | Expanded hierarchy | 1.00 | 1.00 |
| 224 | No instances | 1.00 | 0.99 |
| 225 | No restrictions | 1.00 | 1.00 |
| 228 | No properties | 1.00 | 1.00 |
| 230 | Flattened classes | 0.94 | 1.00 |
| 231 | Expanded classes | 1.00 | 1.00 |
| 232 | | 1.00 | 0.99 |
| 233 | | 1.00 | 1.00 |
| 236 | | 1.00 | 1.00 |
| 237 | | 1.00 | 0.99 |
| 238 | | 1.00 | 0.99 |
| 239 | | 1.00 | 1.00 |
| 240 | | 1.00 | 1.00 |
| 241 | | 1.00 | 1.00 |
| 246 | | 1.00 | 1.00 |
| 247 | | 1.00 | 1.00 |
| 248 | | Jena error | |
| 248-2 | | 0.69 | 0.68 |
| 248-4 | | 0.71 | 0.69 |
| 248-6 | | 0.73 | 0.71 |
| 248-8 | | 0.69 | 0.68 |
| 249 | | Jena error | |
| 249-2 | | 0.70 | 0.70 |
| 249-4 | | 0.71 | 0.71 |
| 249-6 | | 0.73 | 0.73 |
| 249-8 | | 0.73 | 0.73 |
| 250 | | Jena error | |
| 250-2 | | 1.00 | 0.79 |
| 250-4 | | 1.00 | 0.61 |
| 250-6 | | 0.93 | 0.42 |
| 250-8 | | 0.88 | 0.21 |

| Bench | Description | Prec. | Rec. |
|---|---|---|---|
| 251 | | Jena error | |
| 251-2 | | 0.99 | 0.78 |
| 252-4 | | 0.53 | 0.53 |
| 252-6 | | 0.55 | 0.55 |
| 252-8 | | 0.55 | 0.55 |
| 253 | | Jena error | |
| 253-2 | | 0.71 | 0.69 |
| 253-4 | | 0.69 | 0.68 |
| 253-6 | | 0.67 | 0.66 |
| 253-8 | | 0.69 | 0.68 |
| 254 | | Jena error | |
| 254-2 | | 1.00 | 0.79 |
| 254-4 | | 1.00 | 0.61 |
| 254-6 | | 0.93 | 0.42 |
| 254-8 | | 0.88 | 0.21 |
| 257 | | Jena error | |
| 257-2 | | 1.00 | 0.79 |
| 257-4 | | 1.00 | 0.61 |
| 257-6 | | 0.93 | 0.42 |
| 257-8 | | 0.88 | 0.21 |
| 258 | | Jena error | |
| 258-2 | | 0.99 | 0.78 |
| 258-4 | | 1.00 | 0.59 |
| 258-6 | | 0.97 | 0.40 |
| 258-8 | | 0.95 | 0.22 |
| 259 | | Jena error | |
| 259-2 | | 0.55 | 0.55 |
| 259-4 | | 0.51 | 0.51 |
| 259-6 | | 0.55 | 0.55 |
| 259-8 | | 0.54 | 0.54 |
| 260 | | Jena error | |
| 260-2 | | 0.96 | 0.79 |
| 260-4 | | 1.00 | 0.62 |
| 260-6 | | 0.92 | 0.41 |
| 260-8 | | 0.88 | 0.24 |
| 261 | | Jena error | |
| 261-2 | | 1.00 | 0.79 |
| 261-4 | | 1.00 | 0.79 |
| 261-6 | | 1.00 | 0.79 |
| 261-8 | | 1.00 | 0.79 |
| 262 | | Jena error | |
| 262-2 | | 1.00 | 0.79 |
| 262-4 | | 1.00 | 0.61 |
| 262-6 | | 0.93 | 0.42 |
| 262-8 | | 0.88 | 0.21 |
| 265 | | Jena error | |
| 266 | | Jena error | |
| 301 | BibTeX/MIT | 0.91 | 0.68 |
| 302 | BibTeX/UMBC | 0.87 | 0.56 |
| 303 | BibTeX/Karlsruhe | 0.73 | 0.73 |
| 304 | BibTeX/INRIA | 0.95 | 0.92 |

| Conference | Prec. | Rec. |
|---|---|---|
| cmt-conference | 0.50 | 0.56 |
| cmt-confof | 0.55 | 0.38 |
| cmt-edas | 0.69 | 0.69 |
| cmt-ekaw | 0.55 | 0.55 |
| cmt-iasted | 0.50 | 1.00 |
| cmt-sigkdd | 0.77 | 0.83 |
| conference-confof | 0.53 | 0.60 |
| conference-edas | 0.48 | 0.59 |
| conference-ekaw | 0.50 | 0.48 |
| conference-iasted | 0.63 | 0.36 |
| conference-sigkdd | 0.71 | 0.67 |
| confof-edas | 0.45 | 0.53 |
| confof-ekaw | 0.62 | 0.65 |
| confof-iasted | 0.36 | 0.44 |
| confof-sigkdd | 0.80 | 0.57 |
| edas-ekaw | 0.65 | 0.57 |
| edas-iasted | 0.64 | 0.37 |
| edas-sigkdd | 0.88 | 0.47 |
| ekaw-iasted | 0.54 | 0.70 |
| ekaw-sigkdd | 0.78 | 0.64 |
| iasted-sigkdd | 0.59 | 0.87 |