CANARD Complex Matching System: Results of the 2019 OAEI Evaluation Campaign*

Elodie Thiéblin, Ollivier Haemmerlé, Cassia Trojahn

IRIT & Université de Toulouse 2 Jean Jaurès, Toulouse, France
{firstname.lastname}@irit.fr

Abstract. This paper presents the results from the CANARD system in the OAEI 2019 campaign. CANARD is a system able to generate complex alignments. It is based on the notion of competency questions for alignment, as a way of expressing user needs. The system has participated in tracks where instances are available (populated Conference and Taxon datasets). This is the second participation of CANARD in the OAEI campaigns.

1 Presentation of the system

1.1 State, purpose, general statement

The CANARD (Complex Alignment Need and A-box based Relation Discovery) system discovers complex correspondences between populated ontologies based on Competency Questions for Alignment (CQAs). CQAs represent the knowledge needs of a user and define the scope of the alignment [4]. They are competency questions that need to be satisfied over two or more ontologies. Our approach takes as input a set of CQAs translated into SPARQL queries over the source ontology. The answer to each query is a set of instances retrieved from a knowledge base described by the source ontology. These instances are matched with those of a knowledge base described by the target ontology. The generation of the correspondence is performed by matching the subgraph from the source CQA to the lexically similar surroundings of the target instances.

In comparison with last year's version [3], CANARD can now deal with *binary* CQAs, *i.e.*, CQAs whose expected answers are pairs of instances or literal values. Last year it could only deal with *unary* CQAs (*i.e.*, CQAs whose expected answers are sets of instances). For example, here are examples of unary, binary and N-ary CQAs:

- A *unary* CQA expects a set of instances or values, *e.g.*, *Which are the accepted paper? (paper1), (paper2).*
- A *binary* CQA expects a set of instances or value pairs, *e.g.*, *Who wrote which paper? (person1, paper1), (person2, paper2).*

– An *n-ary* CQA expects a tuple of size 3 or more, *e.g.*, *What is the rate associated with which review of which paper? (paper1, review1, weak accept), (paper1, review2, reject).*

1.2 Specific techniques used

The approach has not changed much from last year [3]. The main difference with respect to binary CQAs is in Step (4), where two instances of the pair answer are matched instead of one (as in the case of unary CQAs), Step (5) and Step (8) which deal with the subgraph extraction and pruning.

The approach is detailed in the following steps over an example: the CQA expressed as a SPARQL query over the source knowledge base is:
`SELECT ?x ?y WHERE { ?x o1:paperWrittenBy ?y. }`

(1) Extract source DL formula e_s (*e.g.*, o_1*:paperWrittenBy*) from the SPARQL query.
(2) Extract lexical information from the CQA, L_s set labels of atoms from the DL formula (*e.g.*, "paper written by").
(3) Extract source answers ans_s of the CQA (*e.g.*, a pair of instances *(*o_1*:paper1,* o_1*:person1)*).
(4) Find equivalent or similar target answers ans_t to the source instances ans_s (*e.g.* o_1*:paper1* $\sim$ o_2*:paper1* and o_1*:person1* $\sim$ o_2*:person1*).
(5) Retrieve the subgraphs of target answers: for a binary query, it is the set of paths between two answer instances as well as the types of the instances appearing in the path (*e.g.*, a path of length 1 is found between o_2*:paper1* and o_2*:person1*). The path is composed of only one property and there are no other instances than o_2*:paper1* and o_2*:person1* in this path. Their respective types are retrieved: (o_2*:Paper*,o_2*:Document*) for o_2*:paper1* and (o_2*:Person*) for o_2*:person1*.
(6) For each subgraph, retrieve L_t the labels of its entities (*e.g.*, o_2*:writes* $\rightarrow$ "writes", o_2*:Person* $\rightarrow$ "person", o_2*:Paper* $\rightarrow$ "paper", *etc.*).
(7) Compare L_s and L_t.
(8) Select the subgraph parts with the best score, transform them into DL formulae. Keep the best path variable types if their similarity is higher than a threshold. (*e.g.*, the best type for the instance o_2*:paper1* is o_2*:Paper* because its similarity with the CQA labels is higher than the similarity of o_2*:Document*).
(9) Filter the DL formulae based on their confidence score (if their confidence score is higher than a threshold).
(10) Put the DL formulae e_s and e_t together to form a correspondence (*e.g.*, $\langle$ o_1*:paperWrittenBy* , *dom(*o_2*:Paper)* $\sqcap$ o_2*:writes*$^-$, $\equiv$ $\rangle$ and express this correspondence in a reusable format (*e.g.*, EDOAL). The confidence assigned to a correspondence is the similarity score of the DL formula computed.

The instance matching phase (Step (4)) is based on existing *owl:sameAs*, *skos:closeMatch*, *skos:exactMatch*. In case these links are not available, and exact label matching is applied instead.

Finding a subgraph (Step (5) and (8)) for a pair of instances consists in finding a path between the two instances. The shortest paths are considered more accurate. Because finding the shortest path between two entities is a complex problem, paths of length below a threshold are sought. First, paths of length 1 are sought, then if no path of length 1 is found, paths of length 2 are sought, *etc.* If more than one path of the same length are found, all of them go through the following process. When a path is found, the types of the instances forming the path are retrieved. If the similarity of the most similar type to the CQA is above a threshold, this type is kept in the final subgraph.

For example, for a "*paper written by*" CQA with the answer (o_2*:paper1*,o_2*:person1*) in the target knowledge, a subgraph containing the following triples is found:

1. $\langle$ o_2*:person1* , o_2*:writes* , o_2*:paper1* $\rangle$
2. $\langle$ o_2*:paper1* , *rdf:type* , o_2*:Paper* $\rangle$
3. $\langle$ o_2*:paper1* , *rdf:type* , o_2*:Document* $\rangle$
4. $\langle$ o_2*:person1* , *rdf:type* , o_2*:Person* $\rangle$

The most similar type of o_2*:person1* is o_2*:Person*, which is below the similarity threshold. Triple 4 is then removed from the subgraph. The most similar type of o_2*:paper1* is o_2*:Paper*. Triple 3 is therefore removed from the subgraph. o_2*:Paper*'s similarity is above the similarity threshold: triple 2 stays in the subgraph. The translation of a subgraph into a SPARQL query is the same for binary and unary CQAs. Therefore, the subgraph will be transformed into a SPARQL query and saved as the following DL formula: *dom*(o_2*:Paper*) $\sqcap$ o_2*:writes*$^-$.

The similarity between the sets of labels L_s and L_t of Step (7) is the cartesian product of the string similarities between the labels of L_s and L_t (equation 1).

$$sim(L_s, L_t) = \sum_{l_s \in L_s} \sum_{l_t \in L_t} strSim(l_s, l_t) \tag{1}$$

$strSim$ is the string similarity between two labels l_s and l_t (equation 2). τ is the threshold for the similarity measure. In our experiments, we have empirically set up $\tau = 0.5$. $\tau = 0.5$ in our implementation.

$$strSim(l_s, l_t) = \begin{cases} \sigma \text{ if } \sigma > \tau, \text{ where } \sigma = 1 - \dfrac{levenshteinDist(l_s, l_t)}{\max(|l_s|, |l_t|)} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

The confidence value score of a correspondence (Step (9)) is calculated with the following equation, then truncated to 1:

$$confidence = labelSim + structuralSim \tag{3}$$

Label similarity *labelSim* is the sum of the label similarity of each entity of the formula with the CQA.

Structural similarity *structSim*. This similarity was introduced to enhance some structural aspects in a formula. In the implementation of the approach, this value is set to 0.5 when a path between the two instances of the answer, and 0 for a unary CQA subgraph. Indeed, if the label similarity of the path is 0, the structural similarity hints that the fact that a path was found is a clue in favour of the resulting DL formula.

1.3 Adaptations made for the evaluation

Automatic generation of CQAs OAEI tracks do not cover CQAs i.e., the CQAs can not be given as input in the evaluation. We extended last year's query generator so that it can output binary queries. The query generator now produces three types of SPARQL queries: *Classes*, *Properties* and *Property-Value pairs*.

Classes For each *owl:Class* populated with at least one instance, a SPARQL query is created to retrieve all the instances of this class. If `<o1#class1>` is a populated class of the source ontology, the following query is created:
`SELECT DISTINCT ?x WHERE {?x a <o1#class1>.}`

Properties For each *owl:ObjectProperty* or *owl:Dataproperty* with at least one instantiation in the source knowledge base, a SPARQL query is created to retrieve all instantiations of this property. If `<o1#property1>` is an instantiated property of the source ontology, the following query is created:
`SELECT DISTINCT ?x ?y WHERE {?x <o1#property1> ?y.}`

Property-Value pairs Inspired by the approaches of [1,2,5], we create SPARQL queries of the form

- `SELECT DISTINCT ?x WHERE {?x <o1#property1> <o1#Value1>.}`
- `SELECT DISTINCT ?x WHERE {<o1#Value1> <o1#property1> ?x.}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value".}`

These property-value pairs are computed as follow: for each property (object or data property), the number of distinct object and subject values are retrieved. If the ratio of these two numbers is over a threshold (arbitrarily set to 30) and the smallest number is smaller than a threshold (arbitrarily set to 20), a query is created for each of the less than 20 values. For example, if the property `<o1#property1>` has 300 different subject values and 3 different object values (`"Value1"`, `"Value2"`, `"Value3"`), the ratio $|subject|/|object| = 300/3 > 30$ and $|object| = 3 < 20$. The 3 following queries are created as CQAs:

- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value1".}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value2".}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value3".}`

The threshold on the smallest number ensures that the property-value pairs represent a category. The threshold on the ratio ensures that properties represent categories and not properties with few instantiations.

Implementation adaptations In the initial version of the system, Fuseki server endpoints are given as input. For the SEALS evaluation, we embedded a Fuseki server inside the matcher. The ontologies are downloaded from the SEALS repository, then uploaded in the embedded Fuseki server before the matching process can start. This downloading-uploading phase takes time, in particular when dealing with large files.

The CANARD system in the SEALS package is available at `http://doi.org/10.6084/m9.figshare.7159760.v2`. The generated alignments in EDOAL format are available at:

- **Populated Conference**: `http://oaei.ontologymatching.org/2019/results/complex/popconf/populated_conference_results.zip`
- **GeoLink**: `http://oaei.ontologymatching.org/2019/results/complex/geolink/geolink_results.zip`
- **Taxon**: `http://oaei.ontologymatching.org/2019/results/complex/taxon/results_taxon_2019.zip`

In this year's OAEI complex track, the Populated Conference, GeoLink and Taxon subtracks provide datasets with common instances. CANARD could generate alignments on these three datasets.

2 Results

2.1 Populated Conference

CANARD achieves this task with the longest runtime (96 min). The number of correspondences output by CANARD is detailed in Table 1. The results are detailed in Table 2.

CANARD achieves the highest the best query Fmeasure CQA Coverage score. AMLC achieves the best classical CQA Coverage, CANARD the second best. Both achieve CQA Coverage scores above ra1, but CANARD does not rely on an input alignment (in opposite to AMLC).

The classical Precision of CANARD is the lowest, its query Fmeasure precision above that of AMLC.

2.2 GeoLink

The number of correspondences output by CANARD is detailed in Table 3. The results are detailed in Table 4.

Relaxed precision and recall scores are calculated based on how the entities in the output correspondences are similar to those in the reference correspondences. All multiplied by a coefficient given the relation of the output correspondence and that of the reference one.

CANARD achieves the second best relaxed precision score, behind POMAP++ and the second best relaxed recall score behind AROA.

Table 1: Number of correspondences output by CANARD over the Populated Conference dataset

pair	(1:1)	(1:n)	(m:1)	(m:n)	Total
cmt-conference	19	100	0	5	124
cmt-confOf	18	17	0	6	41
cmt-edas	22	59	2	12	95
cmt-ekaw	11	111	0	12	134
conference-cmt	17	80	0	7	104
conference-confOf	28	13	3	0	44
conference-edas	17	38	0	8	63
conference-ekaw	31	120	2	3	156
confOf-cmt	15	37	0	0	52
confOf-conference	14	22	0	0	36
confOf-edas	15	36	0	0	51
confOf-ekaw	14	39	0	0	53
edas-cmt	20	50	0	4	74
edas-conference	16	49	0	2	67
edas-confOf	24	28	1	0	53
edas-ekaw	18	121	0	4	143
ekaw-cmt	15	71	0	0	86
ekaw-conference	31	80	0	0	111
ekaw-confOf	13	16	0	0	29
ekaw-edas	30	55	0	1	86
TOTAL	388	1142	8	64	1602

2.3 Taxon

CANARD has the longest runtime over the Taxon dataset (512 minutes $\sim$ 8h32). It is longer than last year's rutime (42 minutes) because the inclusion of binary queries in the process increases the number of input queries. Moreover the path finding algorithm consists in looking for all possible paths between two instances relies on SPARQL queries which take a long time to be executed.

The number of correspondences output by CANARD is detailed in Table 1. The results are detailed in Table 2.

Last year, CANARD had output 142 correspondences. This year it has output 791.

CANARD achieves the best CQA Coverage scores over the Taxon dataset. This year, the evaluation was oriented. For example, let's take a set of equivalent correspondences: Q=⟨SELECT ?x WHERE{ ?x a agtx:Taxon}, SELECT ?x WHERE{ ?x a dbo:Species}⟩. If an output alignment *agronomicTaxon-dbpedia* contains ⟨ *agtx:Taxon , dbo:Species ,* $\equiv$ ⟩ but the alignment *dbpedia-agronomicTaxon* does NOT contain ⟨ *dbo:Species , agtx:Taxon ,* $\equiv$ ⟩. The coverage score of Q for the pair *agronomicTaxon-dbpedia* is 1 but the coverage score of Q for *dbpedia-agronomicTaxon* is 0. Last year the evaluation was non-oriented, so the coverage score of Q would be the same (1.0) for both pairs. Taking that into consideration, we computed that if the evaluation was oriented this year, the classical

Table 2: Results of CANARD over the Populated Conference dataset

	Coverage		Precision		
pair	classical	query Fmeasure	classical	query Fmeasure	not disjoint
cmt-conference	0.28	0.53	0.15	0.48	0.90
cmt-confOf	0.50	0.50	0.22	0.60	0.98
cmt-edas	0.65	0.65	0.14	0.42	0.97
cmt-ekaw	0.35	0.59	0.07	0.39	0.97
conference-cmt	0.41	0.45	0.14	0.50	0.85
conference-confOf	0.30	0.35	0.25	0.55	0.73
conference-edas	0.36	0.38	0.19	0.41	0.79
conference-ekaw	0.38	0.47	0.19	0.45	0.78
confOf-cmt	0.50	0.71	0.19	0.76	1.00
confOf-conference	0.27	0.40	0.39	0.73	1.00
confOf-edas	0.23	0.28	0.14	0.45	0.67
confOf-ekaw	0.29	0.42	0.17	0.43	0.83
edas-cmt	0.59	0.67	0.27	0.54	0.97
edas-conference	0.39	0.53	0.37	0.62	0.97
edas-confOf	0.33	0.39	0.21	0.39	0.60
edas-ekaw	0.62	0.72	0.16	0.45	0.87
ekaw-cmt	0.43	0.58	0.30	0.58	0.92
ekaw-conference	0.30	0.50	0.30	0.62	0.93
ekaw-confOf	0.23	0.33	0.31	0.61	0.93
ekaw-edas	0.58	0.64	0.10	0.46	0.92
Average	0.40	0.51	0.21	0.52	0.88

Table 3: Number of correspondences output by CANARD over the GeoLink dataset

pair	(1:1)	(1:n)	(m:1)	(m:n)	Total
gbo-gmo	14	17	13	1	45
gmo-gbo	12	3	0	0	15

CQA Coverage of CANARD would have been 0.197, which shows significant improvement over last year's result: 0.13.

Some correspondences such as $\langle$ *agronomicTaxon:FamilyRank* , $\exists$ *dbo:family*$^{-}$*.wikidata:Q756* , $\equiv \rangle$ or $\langle$ *agronomicTaxon:GenusRank* , $\exists$ *dbo:genus*$^{-}$*.wikidata:Q756* , *(* $\rangle$wikidata:Q756 being the Plant class in wikidata) have more specific target members because of the Plant type restriction. Such correspondences entail higher precision-oriented CQA Coverage and Precision scores than classical ones.

3 General comments

CANARD relies on common instances between the ontologies to be aligned. Hence, when such instances are not available, as for the Hydrography datasets, the approach is not able to generated complex correspondences. Furthermore, CANARD is need-oriented and requires a set competency questions to guide the

Table 4: Results of CANARD over the Populated Conference dataset

Relaxed Precision	Relaxed Fmeasure	Relaxed Recall
0.85	0.59	0.46

Table 5: Number of correspondences output by CANARD over the Taxon dataset

pair	(1:1)	(1:n)	(m:1)	(m:n)	Total
agronomicTaxon-agrovoc	3	25	0	0	28
agronomicTaxon-dbpedia	10	38	0	0	48
agronomicTaxon-taxref	4	28	0	0	32
agrovoc-agronomicTaxon	0	6	4	23	33
agrovoc-dbpedia	3	33	2	21	59
agrovoc-taxref	0	0	0	0	0
dbpedia-agronomicTaxon	5	62	4	26	97
dbpedia-agrovoc	8	57	0	29	94
dbpedia-taxref	18	198	0	29	245
taxref-agronomicTaxon	9	26	0	13	48
taxref-agrovoc	2	17	0	5	24
taxref-dbpedia	5	50	5	23	83
TOTAL	67	540	15	169	791

matching process. Here, these "questions" have been automatically generated based on a set of patterns.

In comparison to last year's campaign, CANARD can now deal with binary CQAs in the form of SPARQL queries with two variables in the SELECT clause.

CANARD's runtime is extremely long. It depends (among other things) on the performance of the SPARQL endpoint it interrogates and the presence (or not) of equivalent links.

However, even with generated queries (instead of user input CQAs) it obtains some of the best coverage scores.

4 Conclusions

This paper presented the adapted version of the CANARD system and its preliminary results in the OAEI 2019 campaign. This year, we have been participated in the Taxon, Populated Conference and GeoLink track, in which ontologies are populated with common instances. CANARD was the only system to output complex correspondences on the Taxon track.

Acknowledgements

Elodie Thiéblin has been funded by Pôle Emploi for the redaction of this paper. The authors have also been partially supported by the CNRS Blanc project RegleX-LD.

Table 6: Results of CANARD over the Taxon dataset

	CQA Coverage				Precision			
pair	classical	rec.-or.	prec.-or.	overlap	classical	re.-or.	prec.-or.	overlap
agronomicTaxon-agrovoc	0	0.67	0.33	0.83	0.14	0.64	0.39	1.00
agronomicTaxon-dbpedia	0	0.42	0.58	0.83	0.06	0.40	0.42	0.98
agronomicTaxon-taxref	0.33	0.50	0.42	0.50	0.28	0.76	0.57	1.00
agrovoc-agronomicTaxon	0.17	0.17	0.17	0.17	0.12	0.79	0.50	0.91
agrovoc-dbpedia	0.17	0.17	0.17	0.17	0.07	0.27	0.22	0.58
agrovoc-taxref	0	0	0	0	NaN	NaN	NaN	NaN
dbpedia-agronomicTaxon	0.17	0.17	0.17	0.17	0.06	0.53	0.56	0.89
dbpedia-agrovoc	0.17	0.17	0.17	0.17	0.03	0.47	0.36	0.78
dbpedia-taxref	0.17	0.17	0.17	0.17	0.03	0.21	0.16	0.94
taxref-agronomicTaxon	0.33	0.50	0.42	0.50	0.04	0.31	0.24	1.00
taxref-agrovoc	0.17	0.42	0.42	0.50	0.04	0.33	0.28	1.00
taxref-dbpedia	0	0.08	0.17	0.33	0.04	0.30	0.30	0.99
Average	0.14	0.28	0.26	0.36	0.08	0.45	0.36	0.91

References

1. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: ISWC. pp. 598–614. Springer (2010)
2. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Discovering concept coverings in ontologies of linked data sources. In: ISWC. pp. 427–443. Springer (2012)
3. Thiéblin, É., Haemmerlé, O., Trojahn, C.: CANARD complex matching system: results of the 2018 OAEI evaluation campaign. In: Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 138–143 (2018), http://ceur-ws.org/Vol-2288/oaei18_paper4.pdf
4. Thiéblin, E., Haemmerlé, O., Trojahn, C.: Complex matching based on competency questions for alignment: a first sketch. In: Ontology Matching Workshop. p. 5 (2018)
5. Walshe, B., Brennan, R., O'Sullivan, D.: Bayes-recce: A bayesian model for detecting restriction class correspondences in linked open data knowledge bases. International Journal on Semantic Web and Information Systems 12(2), 25–52 (2016)