OACAS
Ontologies Alignment using Composition and Aggregation of Similarities

Sami Zghal
CRIL CNRS FRE 2499, Artois University, IUT of Lens, Rue de l'Université - S.P. 16, 62307 Lens Cedex, France

Marouen Kachroudi, Sadok Ben Yahia
Computer Science Department, Faculty of Sciences of Tunis, Campus Universitaire, 1060 Tunis, Tunisia

Engelbert Mephu Nguifo
LIMOS CNRS UMR 6158, Complexe scientifique des Cézeaux, BP 125, 63173 Aubiere Cedex, France

Keywords: Ontology alignment, Similarity measures, Composition of similarities.

Abstract: Ontologies are the kernel of semantic Web. They allow the explicitation of the semantic purpose for structuring different fields of interest. In order to harmonize them and to guarantee the interoperability between these resources, the topic of alignment of ontologies has emerged as an important process to reduce their heterogeneity and improve their exploitation. The paper introduces a new method of alignment of OWL-DL ontologies, using a combination and aggregation of similarity measures. Both ontologies are transformed into a graph which describes their information. The proposed method operates in two steps: local (linguistic similarity composition and neighborhood similarity) step and the aggregation one.

1 INTRODUCTION

An ontology is defined as *"an explicit specification of a conceptualization"* (Gruber, 1993). Indeed, an ontology is a set of concepts, relations and possibly axioms representing a knowledge field. Consequently, the diversity of the reality is at the same time a source of richness of and heterogeneity. This heterogeneity unfortunately reduces the interoperability levels (Euzenat, 2001). Thus, the process of ontologies alignment aims to lower the conflict between them.

The ontology alignment issues grasped the interest of the community as witnesses the wealthy number of approaches, *e.g.*, FALCON-AO (Hu et al., 2007), ONTODNA (Kiu and Lee, 2007) and RIMOM (Li et al., 2007), to cite but a few. The FALCON-AO system, contains five modules integrating a graphic interface. The process begins with a "*parsing*" stage to extract a graph model representing the characteristics of the ontology. The following stage consists in choosing the strategy of alignment, through a library of aligners. Their role is the exploitation of the properties of the ontology. The resultant alignment is presented under the format RDF / XML, having considered the linguistic aspect as well as the structural aspect of an ontology. The second system, ONTODNA uses techniques of data mining to tackle the issue of semantic heterogeneity. This system operates in four stages. The algorithm takes in entry two ontologies to be aligned. Then, it launches a process of linguistic and lexical treatments. A stage of Clustering is performed according to the best obtained correspondences. A final linguistic treatment is applied in the sake of restoring the semantic relations between the various ontological entities. The RIMOM system is an alignement tool which contains six strategies. Every strategy is defined according to the type of information that an ontology can contain. Indeed, the system offers seven different alignment methods, which are afterward organized through a linear interpolation function. The system also exploits the structural aspect of an ontology, by the propagation of similarity through its hierarchy. The final alignment is obtained after a sequence of refinements, through heuristic rules to keep the best possible alignment.

The new alignment method, OACAS (Ontologies Alignment using Composition and Aggregation of Similarities), introduces a new alignment algorithm of OWL-DL (Ontology Web Language Description Logic) ontologies. The main thrust of this method is

Zghal S., Kachroudi M., Yahia S. and Nguifo E. (2009).
OACAS - Ontologies Alignment using Composition and Aggregation of Similarities.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 233-238
DOI: 10.5220/0002301802330238

the application of the most suitable similarity measure depending of the category of the node in the ontology. In addition, the OACAS method explores a wider neighborhood than do the pioneering methods of the literature. Carried out experiments showed that OACAS presents very encouraging values of the commonly used evaluation metrics for the assessment of ontologies alignement, especially for the real ontologies.

The paper is organized as follows. Section 2 introduces the new ontology alignment method and thoroughly presents the underlying fundamentals. Section 3 presents an evaluation of the new method that includes a description of the evaluation metrics used, experimental tests and a comparative study. The conclusion and future issues are presented in section 4.

2 THE OACAS METHOD

The proposed new ontology alignment method, OACAS, aligns two ontologies. Both ontologies are described in the OWL-DL language. Both ontologies are transformed in two graphs O-GRAPHS. The obtained graphs are parsed in order to produce the alignment process out. The process of building the graphs allows to map the considered ontologies to be aligned in two graphs, *called* O-GRAPHS. An O-GRAPH describes all the information included in an OWL-DL ontology: classes, relations and instances. Both classes and instances represent the nodes of the graph. The relations between these different entities are induced by the links of an O-GRAPH. Each entity of the ontology is formalized through an associated notion to the RDF formalism. OWL-DL ontology entities are described thanks to OWL language constructors. These constructors are represented through RDF triplets: <subject, predicate, object>. In an OWL-DL ontology, a class or a relation description is an RDF triplet. The subject corresponds to the class or to the relation. Predicates are OWL primitives, which are OWL and RDF properties. Each property, used in a triplet, sketches a knowledge of the described entity. The arrangement of tho se knowledge constitues the entity definition. The representation of an OWL-DL ontology through an O-GRAPH permits to load the ontology in main memory only once. An O-GRAPH, stored in main memory, statistically reduces the time required to access initial OWL-DL ontology disk resident file.

2.1 The Alignment Method

The introduced OACAS method lays on a composition of similarity computation based model. The method starts by exploring the O-GRAPH structure. It determines the nodes of both ontologies to be aligned and gets out the similarity measures. For each node of the same category (or cluster), the alignment model computes similarity mesures between descriptors by using appropriate functions. An aggregation function combines the similarity measures and the node's structures of the nodes to be aligned. Thus, this function considers all the descriptive information of this couple (name, comment and label) as well as its neighborhood structure. The algorithm implementing the OACAS method takes as input two OWL-DL ontologies to be aligned and produces an RDF file containing the aligned nodes as well as their similarity measures. The OACAS method operates into two successive steps. The first one computes the local similarity, whereas the second one computes the aggregation similarity.

2.1.1 The Local Similarity Computation

The local similarity computation is performed into two successive steps. The first step computes many linguistic similarity measures and aggregates them for each couple of nodes belonging to the same category (or type). The second step computes neighborhood similarities by exploiting the structures of the nodes to be aligned.

The Linguistic Similarity Composition. The linguistic similarity computation is carried out once for each node of the same cluster (node of the same type) in the beginning of the alignment process. The linguistic similarity measures of couples of entities of the same type (class, property and instance) are computed using the LINGUISTIC function. The names of properties and instances are used to compute linguistic similarities. For class category, the computation of the linguistic similarity considers both the comments and labels. The computation of linguistic similarities uses different similarity measures. Those measures are adapted to different descriptors (names, comments and labels) of the entities to be aligned. Different similarity values obtained, for the descriptors, are composed. This composition assigns weights to each similarity measure of descriptors. The sum of the assigned weights to different similarity values is equal to 1. This unit sum guarantees that the composition of the similarity produces a normalized value (between 0 and 1). The LEVENSHTEIN similarity measure (Euzenat and Shvaiko, 2007) is used to compute the similarity value between the names of ontological entities. The Q-GRAM similarity measure (Ukkonen, 1992) computes the similarity value between the comments of the ontological entities. The JARO-WINKLER similarity measure (Euzenat and Shvaiko, 2007) computes the similarity value between the labels of ontological entities. The LINGUISTIC function computes composed linguistic similarity of couples of nodes of both ontologies to be aligned, *i.e.*, O_1 and O_2. It takes as input ***(i)*** both ontologies sketched by two corresponding O-GRAPHS; ***(ii)*** linguistics similarity functions (*i.e.*, $Funct$); and ***(iii)*** weighted attributed to the descriptors nodes (*i.e.*, Π_D). As a result, it produces a composed linguistic similarity vector, V_{CLS}, for each couple of nodes. The similarity function $Funct$ considers two nodes, N_1 and N_2, and returns the linguistic similarity value of the descriptor, Sim_{LD}. LEVENSHTEIN or Q-GRAM or JARO-WINKLER implements the similarity function, $Funct$, depending of the type of the nodes. Composed linguistic similarity, Sim_{CL}, is computed depending of the descriptors of nodes to be aligned and associate weights to each descriptor, Π_D. Both nodes (N_1 and N_2) and the associated composed linguistic similarity (Sim_{CL}) are added to the composed linguistic similarity vector (V_{CLS}). The composed linguistic similarity of different couples of entities will be used to compute the neighborhood similarity as sketched in the following.

The Neighborhood Similarity Computation. The NEIGHBORHOOD function considers both ontologies to be aligned (*i.e.*, O_1 and O_2), the composed similarity vector (V_{CLS}), the weights assigned to each category (Π_C) and the weights associated to the neighbor level (Π_L). Therefore, it produces the neighborhood similarity vector, V_{NS}. The neighborhood similarity computation needs composed linguistic similarity of the couple of nodes to be aligned and the nodes structures. Neighborhood nodes are organized by category, node having the same type. The neighborhood similarity computation propagates similarity into two successive neighborhood levels. The first level (level 1) includes direct neighbors of the nodes to be aligned whereas second one (level 2) contains indirect neighbors. Direct neighbors of the first level represent nodes having direct relationship with the node under consideration. Neighbors of the second level represent nodes having relationship with the nodes of the first one. The neighbors entities of the first level are clustered into three categories (classes, instances or properties). Each category (or cluster) includes ontological entities having the same type. After the step of clustering, the neighborhood similarity is computed between those categories. The neighborhood nodes of the level 2 are treated in the same manner as the neighbors of the first one. The neighborhood similarity by group *MSim* takes nodes from vectors VN_1 and VN_2 regrouped by category (where VN_1 and VN_2 denote a vector nodes of O_1 and O_2). The process computation uses the "*Match-Based similarity*" (Valtchev, 1999) as follows:

$$MSim(E,E') = \frac{\sum_{(i,i')\in Pairs(E,E')} Sim_{CLS}(i,i')}{Max(|E|,|E'|)}. \quad (1)$$

Both sets E and E' represent nodes of the same cluster belonging respectively to vectors VN_1 and VN_2. The neighborhood similarity, Sim_N, is computed using Equation 2:

$$Sim_N = \sum_{i\in(1,2)} (\Pi_{Vi}(\sum_{(E,E')} \Pi_{(E,E')} MSim(E,E'))), \quad (2)$$

where i stands for the level (*i.e.*, 1 or 2). The neighborhood similarity, Sim_N is a normalized value, since the sum of weights assigned to different neighbors is equal to 1, ($\Pi_{V1} + \Pi_{V2} = 1$). Direct neighbors (level 1) have more important relationships than those of indirect one (level 2). Thus, nodes of level 1 have an important impact on the produced alignment. For this reason, the weight assigned to the first level, $\Pi_{V1} = 0.8$, is more important than the one assigned to the second level, $\Pi_{V2} = 0.2$. In addition, the sum of weights assigned to the category of nodes is equal to 1 ($\sum(\Pi_C) = 1$). Those weights are uniformly assigned between the different categories. The neighborhood similarity is computed thanks to an iterative process, level by level. The obtained values of the composed linguistic similarity, *i.e.* V_{CLS}, and neighbors similarity, *i.e.* V_{NS}, are combined in order to compute aggregation similarity.

2.1.2 The Aggregated Similarity Computation

The aggregation similarity is a combined similarity between the local similarities (the composed linguistic similarity and the neighborhood similarity). Function AGGREGATION needs to have in input both ontologies to be aligned, O_1 and O_2, the two similarity vectors, V_{CLS} and V_{NS}, and the weights attributed to the both kind of similarities, Π_{CL} and Π_N. It produces the aggregated similarity vector, V_{AS}. For each couple of entities, N_1 and N_2, of the same category of the both ontologies to be aligned, O_1 and O_2, the aggregated similarity is computed as follows:

$$Sim_A(e_1,e_2) = \Pi_{CL} Sim_{CL}(e_1,e_2) + \Pi_N Sim_N(e_1,e_2). \quad (3)$$

Note that the sum of the weights, attributed to each kind of similarity, is equal to 1 in order to have a normalized aggregation (between 0 and 1). In addition, the sum of weights is equal to 1 ($\Pi_{CL} + \Pi_N = 1$). In the next section, we focus on the experimental evaluation of OACAS.

3 EXPERIMENTAL EVALUATION

The carried out experimental evaluation uses the tests provided and distributed by the OAEI[1] in order to promote activities within the ontology alignment community. The goal of these *benchmark* bases is to identify the strength and weakness areas in each alignment algorithm. The battery of tests is based on one particular ontology. This ontology is dedicated to the very narrow domain of bibliography. From this ontology (*i.e.*, Test 101) derives a number of alternative ontologies of the same domain for which alignments are provided. The *benchmark* test library is composed of a set of 51 pairs of ontologies. Each ontology is to be aligned with the reference ontology (*i.e.*, Test 101). The Test 101 contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. The alignment method should supply for each test an alignment. The obtained alignment is compared to the available reference one. Both, obtained and reference alignments, are used to compute evaluation metrics in order to assess the quality of the aligner algorithm.

3.1 Evaluation Metrics

Precision and *recall* are respectively the most used metrics to evaluate the quality of an alignment method (Euzenat et al., 2006). The OAEI uses these measures to assess the quality of the obtained alignment. The main goal of these measures is to assess the automatization of the comparing process of the alignment methods. The first step, within the process of evaluation of the quality of alignment, consists in resolving the problem manually. The manually obtained result is considered as the reference alignment. The comparison between the reference alignment and that obtained by the alignment method produces three sets: N_{found}, $N_{expected}$ and $N_{correct}$. The N_{found} set represents the set of the couples aligned by an alignment method. The $N_{expected}$ set gathers the couples aligned in the reference alignment. The $N_{correct}$ set is the intersection of both N_{found} and $N_{expected}$ sets. It represents the couples that concurrently belong to the reference alignment and to the output of the alignment method. The *precision* is the ratio of the number of pertinent found couples, *i.e.*, "$N_{correct}$", by the number of total couples, *i.e.*, "N_{found}". Thus, it represents the part of the true correspondences between those found. The *precision* function is defined as:

$$precision = \frac{\mid N_{correct} \mid}{\mid N_{found} \mid}. \quad (4)$$

The *recall* is the ratio of found pertinent couples, "$N_{correct}$", by the total number of pertinent couples, "$N_{expected}$". It specifies, the part of the true found correspondences. The *recall* function is defined as:

$$recall = \frac{\mid N_{correct} \mid}{\mid N_{expected} \mid}. \quad (5)$$

Precision and *recall* metrics are used to perform the evaluation of OACAS method.

3.2 Experimentation and Results

The main objective of the experimentations with the OACAS method is to find the best combination of linguistic measures. In the experimental study, various measures have been used. The goal is to experiment different measures in order to find the more appropriate measure associated to the node descriptors. In order to achieve the objective, 27 arrangements of tests have been experimented. Each test uses a particular combination of similarity measures to compute linguistic similarities between the descriptors of entities to be aligned. During the process of the carried out tests, different weights were assigned to the descriptors (names, comments and labels). The nodes to be aligned can have different descriptors. Depending on the descriptors of the nodes, different weights are attributed. In the case where the nodes are described by three descriptors, the weights are 0.8, 0.1 and 0.1 associated respectively to the names, comments and labels. Whereas the nodes contain only names and comments descriptors, the weights are respectively 0.85 and 0.15. The weights 0.85 and 0.15 are assigned to the names and labels where those the entities are described by them. The experimental results obtained are developed in the next subsection.

The combination using three different linguistic similarities (LEVENSHTEIN, Q-GRAM and JARO-WINKLER) is the best one. In fact, the LEVENSHTEIN measure is more appropriate for computing linguistic similarity between the names of entities to be aligned. Whereas, the Q-GRAM measure is more indicated to compute linguistic similarity between comments of ontological entities. JARO-WINKLER measure is more appropriated for computing linguistic similarity between the labels of entities to be aligned. Indeed, names and labels of ontological entities are short

[1]Ontology Alignment Evaluation Initiative - OAEI-2007 Campaign, http://oaei.ontologymatching.org/2007/

strings. For this type of strings, LEVENSHTEIN and JARO-WINKLER measures are more adapted to compute the linguistic similarity. Comments are strings composed with many words. For this type strings, the Q-GRAM measure gives the best linguistic similarity values. Columns 2-3 of Table 1 of the Appendix show precision and recall values for the best combination. The next subsection will compare OACAS *vs* other related methods.

3.3 Comparative Study

In order to evaluate the results obtained by OACAS, Tables 1 and 2 of the Appendix show the obtained precision and recall values. They also represent values respectively obtained by FALCON-AO, ONTODNA and RIMOM methods. With respect to Tables 1 and 2, the OACAS method produces better results than the three others methods, in particular in the 30x family tests. Theses tests represent real ontologies. The OACAS method gives worse results in the 26x family tests. In this family of tests, ontological entities do not have properties (names and comments). Interestingly enough, these ontological components are the main factors in computing the alignment score using the OACAS method. The experimental results also highlight that the performances of the OACAS method are highly linked to the different characteristics of ontological components (names, comments and labels). In the case where the descriptors of the entities are dropped, the quality (precision and recall values) of the alignment is degraded. For example, in the tests 257 and 260 the ontological entities are not described neither by names nor by comments. In addition, relations and properties are absent. Whenever the name and comments descriptors are not present, the values of linguistic similarity measures are lowered and consequently the value of the composed linguistic similarity will follow the same tendency. Moreover, when the considered ontologies to be aligned do not contain relations nor properties, the values of neighborhood similarities decrease. The local (linguistic similarity composition and neighborhood similarity) similarity computation in the OACAS method reduces the aggregation similarity value. For this reason, precision and recall values are degraded.

4 CONCLUSIONS

In this paper, we introduced a new alignment method of OWL-DL ontologies. The new proposed method OACAS, allows to exploit at most the informative present within in an ontology described in OWL-DL. The process of alignement in the OACAS method, contains two phases: a local phase and a phase of aggregation. The local phase allows to calculate the linguistic similarity consisted as well as the neighborhood similarity. This two similarities are combined during the second phase to determine the aggregation similarity. The results obtained by the method OACAS are very encouraging, compared with the results obtained by the methods FALCON-AO, ONTODNA and RiMOM. The method OACAS shows more successful results compared to the other methods in particular on the real ontologies (the family of the tests 30x). In order to improve the OACAS method, some improvements can be brought. Indeed, the method OACAS have to deal with ontologies of huge sizes. The integration of the API WORDNET (Miller, 1995) is necessary, to improve the values of the measures of the linguistic similarity.

REFERENCES

Euzenat, J. (2001). Towards a principled approach to semantic interoperability. In *Proceedings of the IJ-CAI Workshop on Ontology and Information Sharing*, pages 19–25, Seattle, US.

Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W. R., and Yatskevich, M. (2006). Results of the ontology alignment evaluation initiative 2006. In *Proceedings of the First ESWC 2006 international workshop on ontology matching*.

Euzenat, J. and Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag, Heidelberg (DE).

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Hu, W., Zhao, Y., Li, D., Cheng, G., Wu, H., and Qu, Y. (2007). Falcon-AO: results for OAEI 2007. In *Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007)*, pages 170–178, Busan, Korea.

Kiu, C. and Lee, C. (2007). OntoDNA: ontology alignment results for OAEI 2007. In *Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007)*, pages 227–235, Busan, Korea.

Li, Y., Zhong, Q., Li, J., and Tang, J. (2007). Result of ontology alignment with RiMOM at OAEI'07. In *Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007)*, pages 196–205, Busan, Korea.

Miller, G. A. (1995). WORDNET: a Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Ukkonen, E. (1992). Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92(1):191–211.

Valtchev, P. (1999). *Construction automatique de taxonomies pour l'aide la représentation de connaissance par objets*. Thèse de doctorat, Université de Grenoble 1, France.

APPENDIX

Table 1: Precision and recall values for OACAS, FALCON-AO, ONTODNA and RIMOM methods (Part 1).

	OACAS		ONTODNA	
Tests	**Pre.**	**Rec.**	**Pre.**	**Rec.**
101	1.00	1.00	0.94	1.00
103	1.00	1.00	0.94	1.00
104	1.00	1.00	0.94	1.00
201	0.66	0.15	0.11	0.01
202	0.78	0.15	0.11	0.11
203	1.00	1.00	0.94	1.00
204	1.00	0.95	0.93	0.84
205	0.91	0.40	0.57	0.12
206	0.00	0.00	0.69	0.23
207	0.00	0.00	0.69	0.23
208	1.00	0.95	0.93	0.84
209	0.00	0.00	0.57	0.12
210	0.00	0.00	0.69	0.23
221	1.00	1.00	0.93	0.76
222	1.00	1.00	0.94	1.00
223	1.00	1.00	0.94	1.00
224	1.00	1.00	0.94	1.00
225	1.00	1.00	0.94	1.00
228	1.00	1.00	0.53	0.27
230	1.00	1.00	0.91	1.00
231	1.00	1.00	0.94	1.00
232	1.00	1.00	0.93	0.76
233	1.00	1.00	0.53	0.27
236	1.00	1.00	0.53	0.27
237	1.00	1.00	0.94	1.00
238	1.00	1.00	0.94	1.00
239	0.91	1.00	0.50	0.31
240	1.00	1.00	0.50	0.27
241	1.00	1.00	0.53	0.27
246	1.00	1.00	0.50	0.31
247	1.00	1.00	0.50	0.27
248	1.00	0.10	0.11	0.01
249	1.00	0.10	0.11	0.01
250	0.00	0.00	0.00	0.00
251	0.00	0.00	0.11	0.01
252	1.00	0.16	0.11	0.01
253	1.00	0.08	0.11	0.01
254	0.00	0.00	0.00	0.00
257	0.00	0.00	0.00	0.00
258	1.00	0.08	0.11	0.01
259	1.00	0.08	0.11	0.01
260	0.00	0.00	0.00	0.00
261	0.00	0.00	0.00	0.00
262	0.00	0.00	0.00	0.00
265	0.00	0.00	0.00	0.00
266	0.00	0.00	0.00	0.00
301	0.95	0.83	0.88	0.69
302	0.96	0.88	0.90	0.40
303	0.96	0.85	0.90	0.78
304	0.96	0.95	0.92	0.88

Table 2: Precision and recall values for OACAS, FALCON-AO, ONTODNA and RIMOM methods (Part 2).

	FALCON-AO		RIMOM	
Tests	**Pre.**	**Rec.**	**Pre.**	**Rec.**
101	1.00	1.00	1.00	1.00
103	1.00	1.00	1.00	1.00
104	1.00	1.00	1.00	1.00
201	1.00	0.95	1.00	1.00
202	0.87	0.87	1.00	0.80
203	1.00	1.00	1.00	0.88
204	0.98	0.98	1.00	1.00
205	1.00	0.98	1.00	0.99
206	1.00	0.93	1.00	0.99
207	0.98	0.91	1.00	0.99
208	1.00	1.00	0.98	0.86
209	0.79	0.78	1.00	0.84
210	0.81	0.80	0.99	0.85
221	1.00	1.00	1.00	1.00
222	1.00	1.00	1.00	1.00
223	1.00	1.00	1.00	1.00
224	1.00	0.99	1.00	0.99
225	1.00	1.00	1.00	1.00
228	1.00	1.00	1.00	1.00
230	0.94	1.00	0.94	1.00
231	1.00	1.00	1.00	1.00
232	1.00	0.99	1.00	0.99
233	1.00	1.00	1.00	1.00
236	1.00	1.00	1.00	1.00
237	1.00	0.99	1.00	0.99
238	1.00	0.99	1.00	0.99
239	1.00	1.00	1.00	1.00
240	1.00	1.00	1.00	1.00
241	1.00	1.00	1.00	1.00
246	1.00	1.00	1.00	1.00
247	1.00	1.00	1.00	1.00
248	0.85	0.84	0.99	0.78
249	0.87	0.87	1.00	0.79
250	1.00	0.27	1.00	0.55
251	0.56	0.56	0.76	0.58
252	0.71	0.71	0.85	0.70
253	0.85	0.84	0.99	0.77
254	1.00	0.27	1.00	0.27
257	1.00	0.27	1.00	0.55
258	0.54	0.54	0.76	0.57
259	0.70	0.70	0.85	0.69
260	1.00	0.31	0.93	0.45
261	0.89	0.24	1.00	0.27
262	1.00	0.27	1.00	0.27
265	1.00	0.31	0.93	0.45
266	0.89	0.24	1.00	0.27
301	0.91	0.82	0.75	0.67
302	0.90	0.58	0.72	0.65
303	0.77	0.76	0.45	0.86
304	0.96	0.93	0.90	0.97