Ontology Alignment with FOAM++

AbdulHameed Haddad
Computer Science Dept.
Cairo University
College Station, Giza 12613,
Egypt

Akram Selah
Computer Science Dept.
Cairo University
College Station, Giza 12613,
Egypt

ABSTRACT

The rapid use of ontology in distributed systems as a knowledge representation mechanism, has led to a demand for ontology alignment process due to the heterogeneity arising between two or more ontology describing the same domain. Although many alignments tools have been proposed to reinforce the interoperability between different ontologies, most of them use fixed weights ,supplied by domain experts, in order to rate between similarity measures of two ontological entities. In this work we present FOAM++. It is a framework implemented as a java API, which aims to enhance the quality of ontology alignment process. in order to apply our method, we have extended FOAM API utilizing one of its prototypes(NOM). Our alignment method benefits from soft computing (Genetic Algorithms) methodologies in order to learn the used weights dynamically..

General Terms

ontology, ontology heterogeneity, ontology alignment, FOAM, FOAM++, NOM and OAEI.

1. INTRODUCTION

More recently the notion of ontology [1] is becoming widespread in fields such as intelligent information integration, information retrieval, web site management, information retrieval, electronic commerce, knowledge management and web-services discovery [2]. In such distributed systems, ontology's establish a common vocabulary for the community parties to communicate with each other. Tow flavor of using ontologies in the heterogeneous systems. (a) All the parties in the disturbed systems share the same ontology. (b) Every party has its own ontology. For a set of distributed parties (i.e. organizations), it is difficult to work with a standard ontology because of : (i) It is very difficult and expensive for many organizations to reach an agreement of a standard ontology. (ii) Even if they reach a standard ontology, this standard ontology often do not fit all the requirements of the different organizations. Also, if any party uses its own ontology, it will suffer from ontology heterogeneity when it need to contact with other parties. While ontology plays a big role in resolving heterogeneity between distributed systems, Ontology still be a source of heterogeneity because of: (1) Different businesses use different formats and modeling languages to represent their ontologies. (2) Ontologies using the same format differ in their structure and semantics of the terminology they use. (3) Ontologies are constructed by different experts with different goals. (4) Often different ontologies use different terminologies to describe the same thing (i.e. employee and worker reference to the same thing) . The semantic web researchers have investigated this issue much, because of the big role of ontology in the standardization and interoperability between the distributed systems. As result, semantic web researchers have proposed many of ontology alignment approaches to tackle this problem [3].

Ontology alignment can be defined as: "Given two ontologies, aligning one ontology with another one means that for each entity (concept, relation, or instance) in the first ontology, we try to find a corresponding entity, which has the same intended meaning, in the second ontology. For all aligned pairs, there is a similarity degree called alignment confidence. The following example consists of two simple ontologies that are to be aligned. Figure 1 shows two heterogeneous ontologies, O1 and O2. Both of them describes the same domain, but represented in different ways. Alignments are represented by dotted line, labeled by a number represents the confidence degree of aligning the meant pairs.

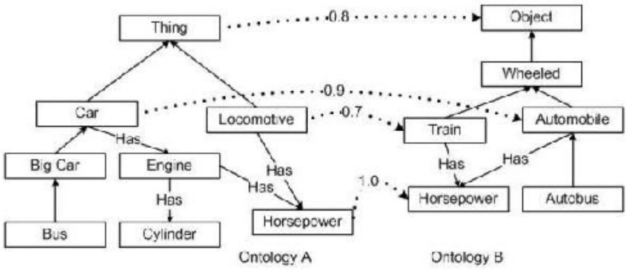


Fig. 1. Ontology A aligned with Ontology B

Semantic web researchers efforts in this domain have resulted into many of alignment methods. It is worth mentioning that the proposed methods follow a generic alignment process (Feature Engineering, Search Step Selection, Similarity Computation, Similarity Aggregation, Interpretation and Iteration) [4]. Regarding similarity aggregation, alignment process need heuristics in order to rate between the used similarities measures according to the importance of the corresponding measure. Most of the outstanding alignment methods use a fixed weights to rate the used similarities. In this paper we proposed a method that learn weights to use them during rating measures similarities. The remainder of this paper is organized as follow: Section 2 discusses our method objectives. Section 3 shows the alignment steps and presents a series of definitions used in our work. Section 4 presents our proposed method. Section 5 discusses applying genetics for ontology alignment. Section 6 evaluate our approach. Results discussion will be shown in section 7. Similar approaches are presented in section 8. Finally, future work and conclusion are shown in section 9.

2. PROBLEM STATEMENT

Ontology alignment has been studied more, so a lot of techniques have appeared for aligning ontologies. Two flavors of ontology alignment methods found listed below :

- Single method matchers: They use only a single method of matching items i.e. Linguistic or taxonomical matchers like (FCA-MERGE, and S-Match).
- More than one alignment method (composite matchers): (i.e. COMA++ and RiMOM) [5], [6].

The main idea of composite matchers is to aggregate similarity values calculated by multiple simple methods (i.e. linguistics, taxonomy, relations, and so on) to determine the degree of similarity between entities. But most of those matchers use a fixed weights to weight the different features importance [7]. Equation 1 shows similarity measures aggregation.

$$ConceptSim(CS) = \frac{w1 * syntaxSim + w2 * propertySim + w3 * classSim}{n} \quad (1)$$

In most of existing matchers, those weights supplied by domain expert. The most popular approaches that use a fixed weights are : COMA++ [5], QickMig [8], FOAM [9], S-Match [10], Falcon [11], ctxMatch [12], and RIMOM [6]. Our contribution is to propose a method that makes any alignment process (which aggregate the similarities linear) to learn weights dynamically. by doing this, we can prevent domain expert interference in assigning weights.

3. DEFINATIONS

3.1 Definition 1: (Ontology Alignment).

An align function (af) that aligns two ontology's O_1, O_2 can be defined as: for any entity (Concept, Relation, Individual) in O_1, it try to find its corresponding entity in O_2, by calculating the similarity degree between each two given entities i.e.:

$$af : O_1 \times O_2 \rightarrow A : \forall e \in O_1 \cdots \exists f \in O_2 \in A \Leftrightarrow sim(e,f) \geq \tau$$

Where

- A is an alignment matrix that contains the aligned pairs,
- $(e,f) \in (C_1, C_2) \cup (\Re_1, \Re_2) \cup (I_1, I_2)$.
- $(C_1, \Re_1, I_1)$ Are the Concepts, Relations, and Individuals set belongs to O_1 respectively, $(C_2, \Re_2, I_2)$ are the Concepts, Relations, and Individuals belongs to O_2 respectively.
- e, f : The same type ontological entities (Concept C ,Relation $\Re$,Individual I).
- $sim(e,f)$: is a similarity confidence between e, f based on a predefined set of features.
- τ Is a given threshold supplied by domain expert.

3.2 Definition 2 (Similarity Measure).

A similarity measure sim is a function that calculates the syntactic and semantic similarities between to given ontological entities, by exploiting a predefined ontological features related to the given entities. $sim(e,f) \rightarrow S$ Where S is a one dimensional array holds the similarity scores (sc) for a specific individual similarity such that each cell represents the sc of the feature have a number equal to the index of that cell. S can be represented as :

$$sim\ (e,f) = IndividualSim_i(e,f), 0 \prec i \leq featuresCnt : sim_i(e,f)$$

3.3 Definition 3: (Calculating Individual Similarities).

Our method relies on FOAM [9] to calculate the similarity between two entitles. FOAM has many approaches, with different heuristics, in order to calculate the similarity score between entities. NOM is one of these approaches, we have chosen NOM, because it fits with our method since it uses a linear weighting method to aggregate the different individual similarities for each entity pairs. NOM studies many of entity features during measuring entity similarity. Any entity feature may be a single item or many of items. For example entity label feature only check the syntactic similarity between entities labels, while super-concepts feature is a set of concepts that are super concepts of the given entity, such features we measure the similarity between two concepts sets. Individual similarity can be represented as:

$$individualSim_i(e,f) \in \{\ sim_{str}, sim_{obj}, sim_{set}\ \}$$

where:

- sim_{str} a syntactic similarity between two strings, sim_{obj} is a similarity between two objects based on some assertions like, and sim_{set} is a similarity value between two give entity sets. sim_{str}, sim_{obj}, sim_{set} are represented as follow:

- $$sim_{str}(s_1,s_2)=\left(0,\frac{\min(|s_1|,|s_2|)-ed(s_1,s_2)}{\min(|s_1|,|s_2|)}\right)$$

  where $ed(s_1,s_2)$ is the edit distance between the given strings.

- $$sim_{obj}(a,b)=\left\{\begin{matrix}1\ldots\ldots sim_{n-1}(a,b)\geq\tau\\ 0\cdots\cdots otherwize\end{matrix}\right\}$$

  where $sim_{n-1}(a,b)$ is the similarity score of a,b from the previous iteration.

- $$sim_{set}(E,F)=\frac{\sum_{e\in E}e}{|E|}.\frac{\sum_{f\in F}f}{|F|}$$

where E,F are sets of entities and $e=\left(\begin{matrix}sim(e,e_1),sim(e,e_2),sim(e,e_3),\cdots,sim(e,f_1),\\ sim(e,f_2),sim(e,f_3),\cdots\end{matrix}\right)$

and

$$f=\left(\begin{matrix}sim(f,f_1),sim(f,f_2),sim(f,f_3),\cdots,sim(f,e_1),\\ sim(f,e_2),sim(f,e_3),\cdots\end{matrix}\right)$$

3.4 Definition 4: (Weighting Individual Similarities).

$sim(e,f)$ calculates different similarities between the given entities using different types of features. Finlay it tries to translate all this similarities into one similarity called similarity score **SC**. Hence it needs to rate between all individual similarities according to the importance of each feature used with each individual similarity. As we mentioned above **FOAM++** tries to learn those weights dynamically, so the learned weights will be a one-dimensional array $w[featureCnt]$, produced by a genetic algorithms based method, such that $w[i]\in[0,1],0\leq i\leq featureCnt$

3.5 Definition 5: (Aggregation Individual Similarities).

Finally $sim(e,f)$ aggregates the individual similarities using this formula:

$$sim(e,f)=\frac{\sum_{k=1}^{k=n}w_k\frac{1}{1+e^{-a_k individualSim_k(e,f)}}}{\sum_{k=1}^{k=n}w_k}$$

3.6 Definition 6: (Alignment Evaluation).

An alignment evaluation method ae expresses how much any alignment process is suitable. This is done by calculating $fMeasure$ for $af(\mathrm{O}_1,\mathrm{O}_2)$ using a reference alignment $\mathrm{A}\Re$ [17]. A high $fMeasure$ means a good alignment method and vice versa. ae can be expressed as: $ae:\mathrm{A}\times\Re\rightarrow fMeasure$

Where

$$fMeasure=\frac{2\times prcision\times recall}{precision+recall}$$

and

$$precision=\frac{relevantMappings\cap retreivedMappings}{relevantMappings}$$

and

$$recall=\frac{relevantMappings\cap retreivedMappings}{retrievedMappings}$$

4. FOAM++

Our contribution in this paper is making FOAM++ to learn the used weights during alignment process using genetic algorithms heuristics. FOAM++ runs a genetic algorithms in order to learn the weights (dynamically), then it passes those weights to FOAM prototype (NOM) to do the final alignment process. Before Appling any genetic solution for any problem, we must to specify two important elements (a) chromosome (solution) representation, (b) fitness function [18]. With refer to definition 4, chromosome representation is observed from solution (chromosome) will be a one-dimensional array of length equals to $featureCnt$. Any gene of the chromosome takes a floating point number $\in[0,1]$. As fitness function FOAM++ relies on $fMeasure$ that judges how much any chromosome is suitable to be kept as a solution or passed to the next generation.

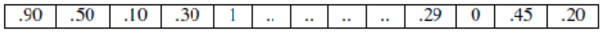
5. APPLYING GENETICS TO ONTOLOGY ALIGNMENT

Recall that the problem that we try to solve, is finding a list of weights to rate the used features in the alignment process. Every weight expresses the degree of the importance of its relevant feature. If the value of w1 is (.70), it means that the importance of the relevant feature is seventy percent. Actually when the alignment starts, we don't know how much a feature x is important? Suppose if we begin with a pool of random weights that express the importance of the used features. We will be in need to a powerful mechanism that tries to look forward to the right weights in this big search space. Genetic algorithms is one of those powerful techniques that tries to search about an optimum solution in such a huge search space. The main steps of using genetics in such problems are solution(chromosome) representation, fitness function, and setting up GA parameters [18].

5.1 Chromosome Representation

Let x_1 is the number of features used in the ontology alignment process. The goal of using GA in the proposed method is to find a solution represents a list of weights that rank the similarity values resulted from the similarity measure process. Since FOAM uses 22 features, so the solution will be represented by a one-dimensional array(chromosome) with 22 elements, each cell(Gene) is a weight value that refers to its corresponding feature importance. All cells (gene) in the chromosome solution may take a floating point value between 0 and 1. Table 1 shows the format representation of the used chromosomes.

TABLE 1. Chromosome Representation



.90	.50	.10	.30	1	..	..	..	..	.29	0	.45	.20

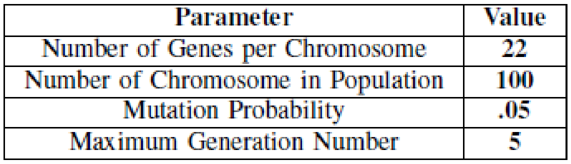
5.2 Fitness Function

The nature of GA methodology begins with random solutions. In order to distinguish between the nice solutions and bad solutions, we need some thing to do that; that what fitness function does exactly. Fitness function receives a solution and report how much this solution is better than the other solutions. FOAM++ begins with random weights and passes the control to the fitness function to weight them. In every generation, FOAM++ does the alignment using all possible solutions (weights). If we align using all the possible solutions in the current generation, there will be a corresponding *fmeasure* for all of those solutions(weights). Hence the fitness function will be maximizing f-measure function.

5.3 Setting up GA Parameters

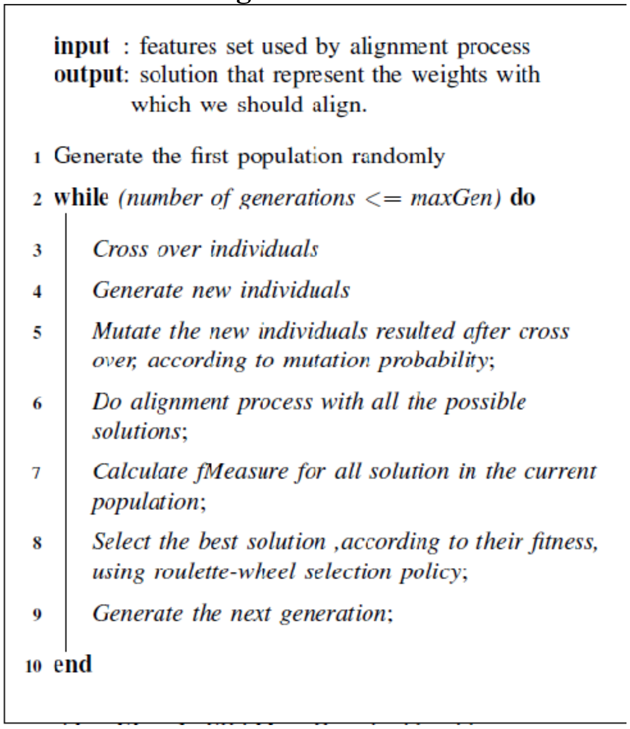
Any Genetic algorithm has its own parameters like(number of genes per chromosome, number of chromosome in population, mutation probability, and number of possible generations). Those parameters depend on the nature of the problem. The preliminary experiments showed us the following parameters listed in table 2 are enough to achieve good results in the alignment process.

Table2. The parameters of the genetic based alignment process



Parameter	Value
Number of Genes per Chromosome	22
Number of Chromosome in Population	100
Mutation Probability	.05
Maximum Generation Number	5

5.4 FOAM++ Algorithm



```
input  : features set used by alignment process
output: solution that represent the weights with
        which we should align.

Generate the first population randomly
while (number of generations <= maxGen) do
    Cross over individuals
    Generate new individuals
    Mutate the new individuals resulted after cross
    over, according to mutation probability;
    Do alignment process with all the possible
    solutions;
    Calculate fMeasure for all solution in the current
    population;
    Select the best solution ,according to their fitness,
    using roulette-wheel selection policy;
    Generate the next generation;
end
```

6. APPROACH EVALUATION

We have conducted our experiments on benchmark data sets supported by OAEI 2009 [19]. We have evaluated FOAM++ in terms of effectiveness , by comparing the resulted alignments with the reference alignments ,provided in OAEI 2009 benchmark data sets, using evaluation metrics: *fMeasure*, *precision, and recall*. Tables 3 shows excerpts of the experimental results of the alignment process using FOAM(fixed weights) and FOAM++(learned weights) from the prospective of evaluation metrics: *Precision, Recall, and fMeasure*. We will discuss the results in the next section.

7. RESULTS DISCUSIOIN

Figure 2 shows the average values of precision, recall, *fmeasure* respectively for both FOAM and FOAM++. In figure 2 we see FOAM++ shows better recall and f-measure average than FOAM, while FOAM shows a better precision than FOAM++. The variation of the returend results is due to the following reasons:

- FOAM precision is better because it relies on a high threshold , so it excludes any aligned pairs has a similarity confidence smaller than the threshold value. It is (.90). This value of threshold considered very high, that makes the possibilities of occurrence false aligns very low, so a high threshold considered a guarantee that produce good precision values.

- FOAM++ recall and f-measure are better because FOAM++ is based on a genetic algorithm that make it learns the used weights for rating the similarity values of its corresponding features. If we take the test cases *201-2*, *201-4*, and *201-6* as examples. First 201-2 test case means that we align the reference ontology with an ontology having .20 of its entity names are scrambled. Similarly *202-4* and *202-6* have .40 and .60 of scrambled names respectively. Entity names, as similarity feature, in *201-2* should have a weight greater than the weight that takes it entity name in *201-4* ontology, that exactly what FOAM++ do. It assigns weights to the used feature according to its importance in the aligning process.
- Average of the precision of (FOAM++) is not better than FOAM precision, because FOAM++ doesn't relies on a fixed threshold. As the approach learns the weights, it learns the threshold too. It seems that sometimes foam++ uses a low threshold that may be a reason for many of false aligned pairs , that produce poor precision. If we make the approach rely on a high fixed threshold, it may enhance the precision and automatically enhance f-measure too.

in my opinion, since f-measure represents the harmonic mean of precision and recall, and it is the main measure to evaluate quality of any ontology alignment tools, FOAM++ is considered better than FOAM, because it showed f-measure average better than FOAM.

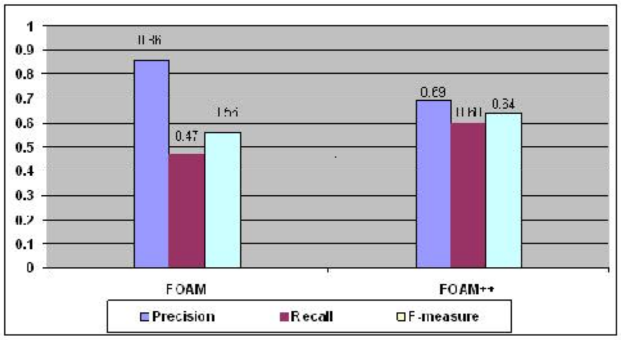


Fig. 2. FOAM++ & FOAM results comparison

8. RELATED WORKS

As we mentioned in section 2, ontology alignment approaches are either individual matchers (which use one method like FCA-MERGE, and S-Match), or combined matchers (which synthesize more than one matcher like COMA++ [5], FOAM [9], CtxMatch, and RiMOM [6]). But those approaches use fixed parameters (weights, threshold, etc) supplied by domain experts. Other approaches have appeared to tackle this problem and prevent domain expert interfere. Some of these approaches benefit from user feed back by exploring user validation of initial alignments for optimizing automatically the configuration parameters of some of the matching strategies of the system, e.g. weights, and thresholds, for the given matching task [7]. In [20], Lee et al. have introduced an alignment approach based on the idea of the exhaustive search, in order to optimize automatically the parameters related to matching task. in [21], Huang et al. proposed an approach based on machine learning techniques like neural networks and genetic algorithms. This approach help the researcher to assign the matching parameters automatically. As a genetic algorithms based approaches, the most outstanding approaches are GOAM [22], and GOAL [7]. Both of them apply genetic algorithms techniques for solving the ontology alignment problem. The difference between them and our approach is the way of utilizing the genetic algorithms procedure to do the alignment process.

9. CONCLUSION AND FUTURE WORK

In this paper, we have presented a genetic algorithms based approach that enhance the quality of ontology alignment, by using a learnt weights to rate between the used features or alignment process. In order to apply our method, we have utilized an open source framework for ontology alignment(FOAM). the preliminary experiments have shown us a promising results. We plan to test our method with others alignment approaches that have a high rank in the domain of ontology alignment. Also, our approach need some heuristics and techniques that improve its performance.

10. REFERENCES

[1] T. Gruber, "A translation approach to portable ontology specification,"Knowledge Acquisition, vol. 5, no. 2, pp. 199–220, 1993.

[2] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic matching of web service capabilities," ISWC. Springer, pp. 333–347, 2002.

[3] E. Jrme and S. Pavel, "Ontology matching," Springer-Verlag, Heildelberg(DE), 2007.

[4] A. Valarakos, V. Spiliopoulos, K. Kotis, and G. Vouros, "Automs-f: A java framework for synthesizing ontology mapping method," Springer-Verlag, Heildelberg (DE), 2007.

[5] S. Massmann, D. Engmann, and E. Rahm, "Coma++: Results for the ontology alignment contest oaei," 2009.

[6] X. Zhang1, Q. Zhong1, J. Li, J. Tang, and G. a. L. Xie, "Rimom results for oaei," 2009.

[7] J. Martinez-Gil, E. Alba, and J. Aldana-Montes, "Optimizing ontology alignments by using genetic algorithms," Knowledge and Information Systems, 2009.

[8] M. Drumm, C.d an Schmitt, H. H. Do, and E. Rahm, "Quickmig -automatic schema matching for data migration projects," Proc. ACM CIKM07., Portugal, 2007.

[9] M. Ehrig, "Ontology alignment, bridging the semantic gap," Springer Science+Business Media, LLC, vol. 5, pp. 146 – 149, New York, 2007.

[10] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, "S-match: an algorithm and an implementation of semantic matching," Proceedings of the European Semantic Web Symposium, Springer, 2004.

[11] W. Hu, G. Cheng, D. Zheng, X. Zhong, and Y. Qu, "The results of falcon-ao in the oaei 2009 campaign. ontology matching," 2009.

[12] S. Niedbala, "Owl-ctxmatch in the oaei 2006 alignment contest. Ontology matching," 2006.

[13] [E. Kaufmann, C. Burki, and M. Klein, "How similar is it? towards personalized similarity measures in ontologies.," In Wirtschaftsinformatik. eEconomy, eGovernment, eSociety, Siebte Internationale, 2005.

[14] A. Maedche and S. Staab, "Measuring similarity between ontologies.," In Proceedings of the European Conference on Knowledge Acquisition and Management(EKAW), 2002.

[15] I. Levenshtein, Binary Codes Capable of Correcting Deletions,Insertions, and Reversals. Doklady Akademii Nauk SSSR, 1965.

[16] T. Cox and M. Cox, "Multidimensional scaling.," 1994.

[17] O. Svab, V. Svatek, and H. Stuckenschmidt, "A study in empirical and 'casuistic' analysis of ontology mapping results.," In Proceedings of the 4th European conference on The Semantic Web (ESWC-07), pp. 655–669, 2007.

[18] J. Koza, "On theprogramming of computers by means of natural selection (complex adaptive systems)," December, 1992.

[19] "Ontology alignment evaluation initiative (oaei).," 2009.

[20] Y. Lee, M. Sayyadian, A. Doan, and A. Rosenthal, "etuner: tun-ing schema matching software using synthetic scenarios," VLDB J, vol. 16, no. 1, pp. 97–122, 2007.

[21] D. J. M. V. J. Huang, J. and M. N. Huhns, "ontology matching using an artificial neural network to learn weights.," IJCAI Workshop on Semantic Web for Collaborative Knowledge Acquisition, 2007.

[22] C. J. J. Wang, Z. Ding, "Goam: Genetic algorithm based ontology matching," in IEEE Asia-Pacific Conference on Services Computing.

Appendix

Table3. FOAM & FOAM++ RESULTS COMPARISON

	FOAM			FOAM++		
Data Sets	**precision**	**recall**	**fMeasure**	**Precision**	**Recall**	**fMeasure**
101	1	1	1	1	1	1
103	0.98	0.98	0.98	0.98	0.98	0.98
104	0.98	0.98	0.98	0.98	0.98	0.98
201	1	0.27	0.42	0.65	0.55	0.59
201-2	0.98	0.81	0.89	1	0.84	0.91
201-4	0.97	0.7	0.81	0.91	0.89	0.9
201-6	0.98	0.58	0.73	0.9	0.87	0.88
201-8	0.98	0.41	0.58	0.73	0.68	0.71
202	1	0.01	0.02	0.39	0.23	0.29
202-2	1	0.75	0.86	1	0.93	0.96
202-4	1	0.56	0.72	0.95	0.74	0.83
202-6	1	0.38	0.55	0.82	0.77	0.79
202-8	1	0.22	0.36	0.91	0.44	0.6
204	1	0.81	0.9	0.96	0.95	0.95
205	0.95	0.36	0.52	0.76	0.7	0.73
206	0.98	0.4	0.57	0.83	0.77	0.8
207	0.92	0.38	0.54	0.84	0.76	0.8
208	1	0.63	0.77	0.96	0.82	0.89
209	1	0.11	0.2	0.54	0.36	0.43
210	1	0.19	0.31	0.71	0.48	0.58
224	1	0.98	0.99	0.97	0.97	0.97
225	0.99	0.98	0.98	0.98	0.98	0.98
228	1	1	1	0.92	1	0.96
232	0.99	0.97	0.98	1	0.99	0.99
233	1	1	1	0.92	1	0.96
236	1	1	1	0.92	1	0.96
237	0.98	0.97	0.97	1	0.99	0.99
238	0.98	0.96	0.97	0.96	0.99	0.97
239	1	1	1	0.91	1	0.95
240	0.97	0.97	0.97	1	1	1
241	1	1	1	0.92	1	0.96
246	1	1	1	1	1	1
247	0.97	0.97	0.97	0.94	0.97	0.96
248	1	0.01	0.02	0.18	0.14	0.16