DSSim-ontology mapping with uncertainty

Miklos Nagy, Maria Vargas-Vera, Enrico Motta

Knowledge Media Institute (Kmi)
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
mn2336@student.open.ac.uk;{m.vargas-vera,e.motta}@open.ac.uk

Abstract. This paper introduces an ontology mapping system that is used with a multi agent ontology mapping framework in the context of question answering. Our mapping algorithm incorporates the Dempster Shafer theory of evidence into the mapping process in order to improve the correctness of the mapping. Our main objective was to assess how applying the belief function can improve correctness of the ontology mapping through combining the similarities which were originally created by both syntactic and semantic similarity algorithms. We carried out experiments with the data sets of the Ontology Alignment Evaluation Initiative 2006 which served as a test bed to assess both the strong and weak points of our system. The experiments confirm that our algorithm performs well with both concept and property names.

1. Presentation of the system

1.1 State, purpose, general statement

In the context of the Semantic Web, AQUA [1,2] an ontology based question answering system offers the possibility to answer user queries from heterogeneous data sources described by their own domain specific ontologies. In order to produce coherent answer to the users' query in this distributed environment the AQUA system need to create ontology mappings between both the concepts and properties of the different domains and the query terms posed by the user. However, in the context of question answering like the AQUA system the dynamic nature of the source information (e.g. web enabled databases) does not always make it possible to create ontology mapping a-priory by the help of a domain expert, but mappings need to be created on the fly. Considering the dynamic nature of this environment an important aspect is how the incomplete and uncertain results of the different similarity algorithms can be interpreted during the mapping process. We believe that proper utilization of uncertainty can considerably improve the mapping precision. However, uncertain data handling and combining uncertain data obtained from different sources in general is computationally expensive operation therefore we use multi agent architecture to address performance related issues.

1.2 Specific techniques used

Creating the particular ontology mappings is an iterative process where ideally the users are involved in the loop as well. In a real case scenario the users pose different questions that contain both concepts and properties of a particular domain. This information then can be used to query the different ontologies, create mapping between its concepts and properties that can be used to answer the particular query. For the Ontology Alignment Contest we have implemented an iterative closed loop which creates the mapping without any human interaction and works as follows:

1. We take a concept (or property) from ontology 1 and consider (refer to it from now) it as the query fragment that would normally be posed by a user. From the query fragment we build up a graph which contains the close context of the query fragment such as the concept and its properties.
2. We take syntactically similar concepts and properties and its synonyms to the query graph from ontology 2 and build a graph that contains both concepts (properties) and its synonyms.

3. Different similarity algorithms (considered as different experts in evidence theory) are used to assess quantitative similarity values (converted into belief mass function) between the nodes of the query and ontology fragment which is considered as an uncertain and objective assessment. Then the information produced by the different algorithms is combined using the Dempster's rule of combination.
4. Based on the combined evidences we assess semantic similarity between the query and ontology graph fragment structures and select those in which we calculate the highest belief function.
5. The selected concepts are added into the alignment.

The overview of the mapping process is depicted on figure 1.

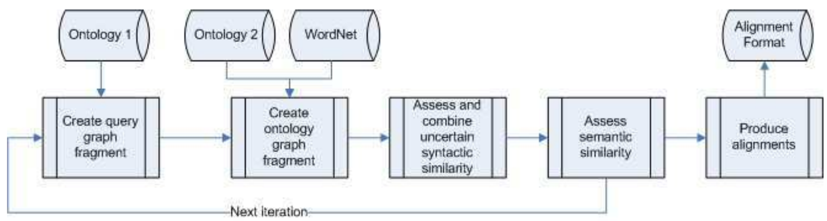

Fig. 1. The iterative mapping process

In order to avoid a complex graph of relationships in the query and the ontology fragments we need to define a reasonable limit on the number of synonyms, which are extracted from the WordNet. To define such a limit is also desirable when we carry out the belief combination since all extracted terms represent a variable where each similarity value needs to be combined with the Dempster's rule of combination. The combination rule implies that the problem space increases exponentially with the number of variables therefore the proper definition of this limit can considerably affect the scalability of our system.

1.2.1 Syntactic similarity

To assess syntactic similarity between ontology entities we use different string-based techniques to match names and name descriptions. These distance functions map a pair of strings to a real number, which indicates a qualitative similarity between the strings. To achieve more reliable assessment we combine different string matching techniques such as edit distance like functions e.g. Monger-Elkan [3] to the token based distance functions e.g. Jaccard [4] similarity. To combine different similarity measures we use Dempster's rule of combination. Several reasonable similarity measures exist however, each being appropriate to certain situations. To maximize our system's accuracy we employ a variety of similarity measures. At this stage of the similarity mapping our algorithm takes one entity from Ontology 1 and tries to find similar entity in extended query. The similarity mapping process is carried out on the following entities:

- Concept-name similarity
- Property name and set similarity

The use of string distances described here is the first step towards identifying matching entities between query and the ontology or between ontologies with little prior knowledge, or ill structured data. However, string similarity alone is not sufficient to capture the subtle differences between classes with similar names but different meanings. So we work with WordNet in order to exploit synonymy at the lexical-level. Once our query sting is extended with lexically synonym entities we calculate the string similarity measures between the query and the ontologies. In order to increase the correctness of our similarity measures the obtained similarity coefficients need to be combined. Establishing this combination method was our primary objective that had been included into the system. Further once the combined similarities have been calculated we developed a simple methodology to derive the belief mass function that is the fundamental property of Demster-Shafer framework.

1.2.2 Semantic similarity

For semantic similarity between concept, relations and the properties we use graph-based techniques. We take the extended query and the ontology input as labeled graphs. The semantic matching is viewed as graph-like structures containing terms and their inter-relationships. The similarity comparison between a pair of nodes from two ontologies is based on the analysis of their positions within the graphs. Our assumption is that if two nodes from two ontologies are similar, their neighbours might also be somehow similar. We consider semantic similarity between nodes of the graphs based on similarity of leaf nodes. That is, two non-leaf schema elements are semantically similar if their leaf sets are highly similar, even if their immediate children are not. The main reason why semantic heterogeneity occurs in the different ontology structures is because different institutions develop their data sets individually, which as a result contain many overlapping concepts. Assessing the above-mentioned similarities in our system we adapted and extended the SimilarityBase and SimilarityTop algorithms [5] used in the current AQUA system for multiple ontologies. Our aim is that the similarity algorithms (experts in terms of evidence theory) would mimic the way a human designer would describe a domain based on a well-established dictionary. What also needs to be considered when the two graph structures are obtained from both the user query fragment and the representation of the subset of the source ontology is that there can be a generalization or specialization of a specific concepts present in the graph which was obtained from the local source and this needs to be handled correctly. In our system we adapted and extended the before mentioned SimilarityBase and SimilarityTop algorithms, which has been proved effective in the current AQUA system for multiple ontologies.

1.2.3 Uncertainty

In our system we use the Dempster-Shafer theory of evidence [6], which provides a mechanism for modeling and reasoning uncertain information in a numerical way particularly when it is not possible to assign a belief to a single element of a set of values. Consequently the theory allows the user to represent uncertainty for knowledge representation, because the interval between support and plausibility can be easily assessed for a set of hypothesizes. Missing data also could be modeled by Dempster-Shafer approach and additionally evidences from two or more sources can be combined using Dempster's rule of combination. The combined support, plausibility, disbelief and uncertainty can each be separately evaluated. The main advantage of the Dempster-Shafer theory over the classical probabilistic theories is the evidence of different levels of abstraction can be represented in a way, which allows clear distinction to be made between uncertainty and ignorance. Further advantage is that the theory provides a method for combining the effect of different learned evidences to establish a new belief by using Dempster's combination rule. The following elements have been used in our system in order to model uncertainty:
Belief mass function (m): is a finite amount of support assigned to the subset of Θ. It represents the strength of some evidence and

$$\sum_{A\subseteq\Theta} m(A) = 1 \qquad \textbf{(1)}$$

where m(A) is our exact belief in a proposition represented by A. The similarity algorithms itself produce these assignment based on the above mentioned (see in section similarity) similarities. As an example consider the query fragment that contains the concept "book". Based on the WordNet we identify that the concept "volume" is one synonym of the "book" so after similarity assessment our variables will have the following belief mass value:

- m(Ontology1_{book}, Ontology2_{volume}) = 0.89
- m(Ontology1_{book}, Ontology2_{book}) = 1.0

In practice we would assess up to 8 synonym similarities with different algorithms (considered as experts) which can be combined based on the combination rule in order to create a more reliable mapping. Once the combined belief mass functions have been assigned the following additional measures can be derived from the available information.
Belief: amount of justified support to A that is the lower probability function of Dempster, which accounts for all evidence E_k that supports the given proposition A.

$$belief_i(A) = \sum_{E_k\subseteq A} m_i(E_k) \qquad \textbf{(2)}$$

An important aspect of the mapping is how one can make a decision over how different similarity measures can be combined and which nodes should be retained as best possible candidates for the match. To combine the qualitative similarity measures that have been converted into belief mass functions we use the Dempster's rule of combination and we retain the node which belief function has the highest value.

Dempster's rule of combination:
Suppose we have two mass functions $m_i(E_k)$ and $m_j(E_k')$ and we want to combine them into a global $m_{ij}(A)$. Following Dempster's combination rule

$$m_{ij}(A) = m_i \oplus m_j = \sum_{E_k \cap E_{k'}} m_i(E_k) * m_j(E_{k'}) \qquad \textbf{(3)}$$

1.2.4 Trust and conflicting beliefs

Based on our experiments with the benchmarks we have investigated why in some cases the belief combination produced incorrect result even thought before the combination a correct mapping could have been derived for the particular case based on individual beliefs. The problem occurs when the different agents' similarity assessment produces conflicting beliefs over the correctness of a particular mapping. A conflict between two beliefs in DS theory can be interpreted qualitatively as one source strongly supports one hypothesis and the other strongly supports another hypothesis, and the two hypotheses are not compatible. In this scenario applying Dempster's combination rule to conflicting beliefs can lead to an almost impossible choice with a very low degree of belief which due to the normalisation will result in the most possible outcome with a very high degree of belief [7, 8]. This combination rule strongly emphasizes the agreement between multiple sources and ignores all the conflicting evidence through a normalization factor. Imagine the following scenario where Ω frame of discernment has three elements $\{e_1,e_2,e_3\}$ and the assigned belief masses on the correctness of the particular mappings are as described on table 1.

	Before normalisation	After normalisation
Agent 1	$m_1(e_1)=0; m_2(e_2)=0.01; m_3(e_3)=0$	$m_1(e_1)=0; m_2(e_2)=1; m_3(e_3)=0$
Agent 2	$m_1(e_1)=0.74; m_2(e_2)=0.35; m_3(e_3)=0.24$	$m_1(e_1)=0.55; m_2(e_2)=0.26; m_3(e_3)=0.19$
Agent 3	$m_1(e_1)=0.69; m_2(e_2)=0.3; m_3(e_3)=0.21$	$m_1(e_1)=0.57; m_2(e_2)=0.25; m_3(e_3)=0.18$

Table 1. Conflicting belief masses

In this scenario the belief of "Agent 1" is in conflict with the other agents' belief and due to the normalization of the hypothesis set a week possibility is transformed into strong support which would result in an incorrect mapping. In our ontology mapping framework the belief functions are considered as a method to model an agent's beliefs, therefore the belief function defined by an agent can also be viewed as a way of expressing the agent's preferences over choices, with respect to masses assigned to different hypotheses. The larger the mass assigned to a hypothesis is the more preferred the hypothesis will be. In this context the problem is how do we handle the agent's conflicting individual preferences that need to be aggregated in order to form a collective preference. We have utilized the degree of trust based on reputation model [9] between the individual agents' belief over the correctness of the mapping. In our scenario the reputation model is particularly appealing because it can be defined as the collective opinion or view about the mapping where this view can be mainly be derived from an aggregation of individual preferences. In our ontology mapping framework we assess trust between the agent's beliefs and determine which agent's belief cannot be trusted ignoring the one which contradicts with the majority of the beliefs which are similar to each other.

1.3 Adaptations made for the evaluation

Our mapping algorithm which is originally based on multi agent architecture has been re-implemented as a standalone mapping process which uses the common WordNet dictionary which is considered more general knowledge then originally we assume in or architecture. Originally our mapping process receives query fragments from the AQUA system where the query fragments contain several concept names and their properties. For the evaluation we modified our mapping process so we consider the individual concept or property names as query fragments which contain less information about the possible mapping then the query fragments that we originally receive from the AQUA system.

1.4 Link to the system and parameters file

http://kmi.open.ac.uk/people/miklos/OAEI2006/DSSemanticSimilarity.zip

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

http://kmi.open.ac.uk/people/miklos/OAEI2006/benchmarks.zip

2. Results

All the tests have been carried out on a commercially available notebook with windows operating system. The mapping algorithm has been implemented in Java and been integrated with the Alignment api. The comments are made on the tests that have been grouped as follows:

2.1 Tests 101-104

The ontologies include (see figure 2) the reference alignment and irrelevant ontology a language generalization and a language restriction. Our results (see result matrix) show that our mapping algorithm creates the mapping with high precision for this tests.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
101	1	0.98	0	0.99	0.98
102	0	NaN	1	NaN	NaN

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
103	1	0.98	0	0.99	0.98
104	1	0.98	0	0.99	0.98
H-mean	1	0.98	0	0.99	0.98

Fig. 2. Results from test 101-104

2.2 Tests 201-210

The ontology 201 that does not contain names and 202 which neither contain names nor comments were not mapped at all by our algorithm. Our algorithms considers only class and property IDs as identified by the “rdf:ID” tag therefore the only information that can be used to create these mappings the “rdfs:comment” but our algorithm does not make use of it. Ontologies 203 and 204 are without comments and certain naming conventions were also mapped with high precision by our algorithm. Ontology 205 which contains synonyms were mapped with high precision but with really weak recall what can be explained by the fact that our algorithm looks for WordNet synonyms based on the full terms from the ontologies so e.g. MastersThesis or MScThesis as one word does not have WordNet synonym but MSc and Thesis separately do. Ontologies 206 to 210 are the French translations of the original ontology and since our algorithm does not look at the comments therefore our mapping has a low recall rate. The results of the mappings for this group are depicted on figure 3.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
201	NaN	0	NaN	NaN	NaN
202	NaN	0	NaN	NaN	NaN
203	1	0.98	0	0.99	0.98
204	0.99	0.68	0.01	0.8	0.67
205	0.88	0.23	0.12	0.36	0.2

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
206	0.91	0.21	0.05	0.34	0.2
207	0.91	0.21	0.09	0.34	0.19
208	0.99	0.68	0.01	0.8	0.67
209	0.88	0.23	0.12	0.36	0.2
210	0.91	0.21	0.09	0.34	0.19
H-mean	0.95	0.34	0.04	0.5	0.33

Fig. 3. Results from test 201-210

2.3 Tests 221-247

Ontologies from 221 to 247 (see figure 4) contain no specialization, flatenned hierarchy, expanded hierarchy, no instance, no restrictions, no datatypes, unit difference, no properties, class vs instances, flattened classes and expanded classes have been mapped with a very high recall and precision rate. We can conclude that on this group of tests our algorithm performs well which can be contributed to the fact that we carry out both syntactic and semantic similarity assessment.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
221	1	0.98	0	0.99	0.98
222	1	0.98	0	0.99	0.98
223	1	0.98	0	0.99	0.98
224	1	0.98	0	0.99	0.98
225	1	0.98	0	0.99	0.98
228	1	1	0	1	1
230	0.99	0.97	0.01	0.98	0.96
231	1	0.98	0	0.99	0.98
232	1	0.98	0	0.99	0.98

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
233	1	1	0	1	1
236	1	1	0	1	1
237	1	0.98	0.01	0.98	0.97
238	1	0.98	0	0.99	0.98
239	1	1	0.03	0.98	0.97
240	1	1	0	1	1
241	1	1	0	1	1
246	1	1	0.03	0.98	0.97
247	1	1	0	1	1
H-mean	0.99	0.98	0	0.99	0.98

Fig. 4. Results from test 221-247

2.4 Tests 248-266

Again since our algorithm considers only class and property IDs as identified by the "rdf:ID" tag therefore these tests have not produced any mapping. In a future implementation we will considerer labels. Then, our similarity algorithm will be able to handle effectively these cases.

2.5 Tests 301-304

For the real word ontologies (see figure 5) our algorithm produced relatively good mappings with good recall and high precision. We believe that the real word ontlogies and the reference ontology were not so different semantically in terms of concept and property hierarchies or structure so the syntactic similarity was dominated the results.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
301	0.87	0.79	0.13	0.83	0.67
302	0.93	0.58	0.07	0.72	0.54

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
303	0.84	0.78	0.16	0.81	0.63
304	0.94	0.89	0.06	0.92	0.84
H-mean	0.9	0.78	0.1	0.83	0.69

Fig. 5. Results from test 301-304

3. General comments

3.1 Comments on the results (strength and weaknesses)

We consider the results successful when we reach a high precision rate since our main objective is to increase ontology mapping precision with incorporating uncertainty into the mapping process. Most of the benchmark tests proved that when different similarity assessments have to be combined handling uncertainty can lead to a high precision rate which is a definite strength of our system. Another strength of our system is that the produced mappings are not very dependent on the structure and hierarchy of the concepts and properties in the ontology (see tests 221-247). Since the multi agent architecture has been replaced with the single process the execution time has increased considerably. Additionally the agents's "specific knowledge" has been replaced with the general WordNet synonyms that negatively influenced the system. Further our algorithm always considers the ID tag in the ontologies therefore any additional information like comments or the language element is omitted. Not considering the language element can be considered as a weakness. However, we believe that comments in ontologies can work well when the ontologies originate from a well controlled environment with strong academic background like universities or research institutions. From the another side if we consider the nature of the semantic web where any private company can place its ontology to the web to support its own web enabled data it can lead to really different comments even for the same concepts or properties.

3.2 Discussions on the way to improve the proposed system

Based on the results we have identified the following improvement possibilities that can further improve our system:

1. We need to split it up the concept and property IDs in the ontologies which are the combination of two or more different terms e.g. MScThesis into unique terms and the WordNet synonyms can be retrieved on the combination of the separated terms. This can lead to a definite improvement of recall number of the particular mapping.
2. Wherever possible or present considering the language tag as primary information. It is important that we create mapping based on the same language. Failing to do so can lead to incorrect mappings that cannot be detected based on qualitative measures.

3.3 Comments on the OAEI procedure

The OAEI procedure and the provided alignment api works well for the benchmarks. However we experienced difficulties with the anatomy ontology. We have tried on several computers but we have always got **OutOfMemoryError** due to the large size of the FMA ontology. Our investigation showed that when the GroupAlign class of the alignment api parses the source and target ontologies into a **org.semanticweb.owl.model.OWLOntology** object the memory usage of the JVM process increases to nearly 1GB. Once the similarity mapping process starts, any manipulation of the original ontology object leads to OutOfMemoryError and causes the process to stop. We have also tried to increase the stack size of the JVM but it did not solve the problem.

3.4 Comments on the OAEI test cases

We have found that most of the benchmark tests can be used effectively to test various aspects of an ontology mapping system since it provides both real word and generated/modified ontologies. The ontologies in the benchmark are conceived in a way that allows anyone to clearly identify system strengths and weaknesses which is an important advantage when future improvements have to be identified. However, our system did not perform as well as we first expected probably due to the fact that most of the classes and properties in the ontologies are organized in a rather flat hierarchy so in our system the semantic similarity component did not influence the overall mappings considerably. Unfortunately, we could not make use of a large group of tests (248-266) since our system does not consider individuals or instances of the classes. Concerning the anatomy data sets we planned to produce alignment as well however, we were unable to successfully run the process using the alignment api due to the reasons described in the section 3.3. The external and blind evaluations are certainly valuable exercises however we plan to utilize them in the future due to technical limitations of our system.

3.4 Comments on the OAEI measures

For our system the precision measure was the most important of all because this gives us the possibility to draw constructive conclusions on how the uncertainty handling can influence the precision of the system. The additional measures like recall and fallout can be used effectively for identifying where do we need to make further improvements in our system.

3.5 Proposed new measures

Besides the traditional measures it would be useful as well to introduce a measure that expresses the difficulty to create the particular mapping. E.g. there is a considerable difference in the level of difficulty between creating mapping with the reference ontology itself (101 to 101) and real word ontology (101 to 304). This measure then could be used to assess the how the particular system can handle mappings that involves complex comparison operations.

4. Conclusions

The increasing popularity of the Semantic Web poses new challenges for ontology mapping. If we accept that mapping ontologies can provide a better knowledge management of the heterogeneous sources on the Semantic Web, then issues of inconsistency and incompleteness need to be addressed. Therefore ontology mapping systems that operate in this environment should have the appropriate mechanisms to cope with these issues. In this complex environment different scientific disciplines need to be utilized together to achieve better results for answering user queries within an acceptable response times. We think that in our implementation we have made an encouraging step towards a theoretical solution but the different key system components such as similarity measure or the scalability of uncertainty handling part needs to be investigated further. In our future research we will investigate how different optimalisation methods for belief combination can be adapted and applied in our scenario with a dynamic multi agent environment where each agent has partial knowledge of the domain. Participating in the Ontology Alignment Evaluation Initiative is an excellent opportunity to test and compare our system with other solutions and helped a great deal identifying the future possibilities that needs to be investigated further.

References

[1] Vargas-Vera, M. and Motta, E. 2004. AQUA - Ontology-based Question Answering System. In Proceedings of Third International Mexican Conference on Artificial Intelligence (MICAI-2004), Mexico City, Mexico.

[2] Vargas-Vera, M.; Motta, E.; and Domingue, J. 2003. AQUA: An Ontology-Driven Question Answering System. AAAI Spring Symposium, New Directions in Question Answering, Stanford, USA.:AAAI Press.

[3] Monge, A. E.; Elkan, C. P. 1996. The field-matching problem: algorithm and applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, US.

[4] Cohen, W.; Ravikumar, P.; and Fienberg, S. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks, In Proceedings of Information Integration on the Web (IIWeb 2003), Accapulco, Mexico.

[5] Vargas-Vera, M.; and Motta, E. 2004. An Ontology-driven Similarity Algorithm. KMI-TR-151, Knowledge Media Institute, The Open University, UK.

[6] Shafer, G. 1976. A Mathematical Theory of Evidence.: Princeton University Press.

[7] Smets, Ph. 1988. Belief functions, In Non-Standard Logics for Automated Reasoning: Academic Press, pp. 253–286.

[8] Zadeh, L. 1986. A simple view of the Dempster–Shafer theory of evidence and its implication for the rule of combination, AI Magazine 7, 85–90.

[9] Ramchurn, S. D., Huynh, T. D. and Jennings, N. R. 2004. The Knowledge Engineering Review, Volume 19: 1-25.

Appendix

Matrix format

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
101	1	0.98	0	0.99	0.98
102	0	NaN	1	NaN	NaN
103	1	0.98	0	0.99	0.98
104	1	0.98	0	0.99	0.98
201	NaN	0	NaN	NaN	NaN
202	NaN	0	NaN	NaN	NaN
203	1	0.98	0	0.99	0.98
204	0.99	0.68	0.01	0.8	0.67
205	0.88	0.23	0.12	0.36	0.2
206	0.91	0.21	0.05	0.34	0.2
207	0.91	0.21	0.09	0.34	0.19
208	0.99	0.68	0.01	0.8	0.67
209	0.88	0.23	0.12	0.36	0.2
210	0.91	0.21	0.09	0.34	0.19
221	1	0.98	0	0.99	0.98
222	1	0.98	0	0.99	0.98
223	1	0.98	0	0.99	0.98
224	1	0.98	0	0.99	0.98
225	1	0.98	0	0.99	0.98
228	1	1	0	1	1
230	0.99	0.97	0.01	0.98	0.96
231	1	0.98	0	0.99	0.98
232	1	0.98	0	0.99	0.98
233	1	1	0	1	1
236	1	1	0	1	1
237	1	0.98	0.01	0.98	0.97

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
238	1	0.98	0	0.99	0.98
239	1	1	0.03	0.98	0.97
240	1	1	0	1	1
241	1	1	0	1	1
246	1	1	0.03	0.98	0.97
247	1	1	0	1	1
248	NaN	0	NaN	NaN	NaN
249	NaN	0	NaN	NaN	NaN
250	NaN	0	NaN	NaN	NaN
251	NaN	0	NaN	NaN	NaN
252	NaN	0	NaN	NaN	NaN
253	NaN	0	NaN	NaN	NaN
254	NaN	0	NaN	NaN	NaN
257	NaN	0	NaN	NaN	NaN
258	NaN	0	NaN	NaN	NaN
259	NaN	0	NaN	NaN	NaN
260	NaN	0	NaN	NaN	NaN
261	NaN	0	NaN	NaN	NaN
262	NaN	0	NaN	NaN	NaN
265	NaN	0	NaN	NaN	NaN
266	NaN	0	NaN	NaN	NaN
301	0.87	0.79	0.13	0.83	0.67
302	0.93	0.58	0.07	0.72	0.54
303	0.84	0.78	0.16	0.81	0.63
304	0.94	0.89	0.06	0.92	0.84
H-mean	0.98	0.55	0.02	0.7	0.53