GMap Results for OAEI 2021*

Weizhuo Li[1,4], Shiqi Zhou[2], Qiu Ji[1(✉)], and Bingjie Lu[3]

[1] School of Modern Posts, Nanjing University of Posts and Telecommunications, China.
[2] School of Computer Science and Engineering, Southeast University, Nanjing, China
[3] Zhejiang Lab, HangZhou, China.
[4] State Key Laboratory for Novel Software Technology, Nanjing University, China
liweizhuo@amss.ac.cn qiuji@njupt.edu.cn

Abstract. GMap is an alternative probabilistic scheme for ontology matching, which combines the sum-product network and the noisy-or model. More precisely, we employ the sum-product network to encode the similarities based on the set of individuals and disjointness axioms. The noisy-or model is utilized to encode the probabilistic matching rules, which describe the influences among entity pairs across ontologies. This paper briefly introduces GMap and its results of two tracks (i.e., Conference, Anatomy) on OAEI 2021.

1 Presentation of the system

1.1 State, purpose, general statement

The state of the art approaches have utilized probabilistic graphical models [1] for ontology matching such as OMEN [2], iMatch [3], CODI [4] and MORW [5]. However, few of them could not guarantee inference tractable and ensure no loss in inference accuracy. Therefore, these matching systems had to adopt approximate inference, so the quality of alignments was influenced to some extend. In this paper, we propose an alternative probabilistic scheme, called GMap, combining the sum-product network (SPN) and the noisy-or model [6]. Except for the tractable inference, these two graphical models have some inherent advantages for ontology matching. For SPN, even if the knowledge (e.g., individuals or disjointness axioms) is missing, SPN can also calculate their contributions by the maximum a posterior (MAP) inference. For the noisy-or model, it is reasonable to incorporate probabilistic matching rules to describe the influences among entity pairs.

Figure 1 shows the matching framework of GMap. Given two ontologies O_1 and O_2, we first calculate the lexical similarity based on edit-distance, external lexicons and TF-IDF [7] with the max strategy. Then, we employ SPN to encode the similarities based on individuals and disjointness axioms and calculate the contribution through Maximum A Posteriori (MAP) inference [1]. After that, we utilize the noisy-or model to encode the probabilistic matching rules and the value calculated by SPN. With one-to-one constraint and crisscross strategy in the refine module, GMap obtains initial alignments.

* This work was partially sponsored by the Natural Science Foundation of China grants (No. 62006125, 61602259), the NUPTSF grant (No. NY220171, NY220176), the Foundation of Jiangsu Provincial Double-Innovation Doctor Program (JSSCBS20210532), and the State Key Laboratory for Novel Software Technology at Nanjing University.

The whole matching procedure of GMap is iterative. If there is no external mapping identified, the process of matching will be terminated.

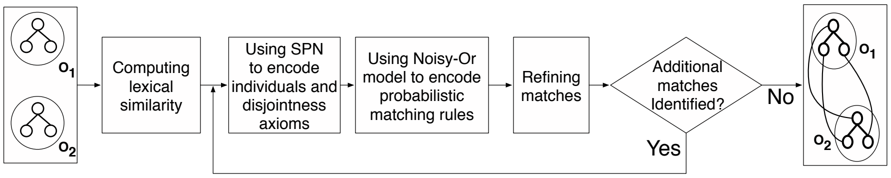

O1
O2
Computing lexical similarity
Using SPN to encode individuals and disjointness axioms
Using Noisy-Or model to encode probabilistic matching rules
Refining matches
Additional matches Identified?
No
Yes
O1
O2

Fig. 1: Matching process in GMap

1.2 Specific techniques used

The similarities based on individuals and disjointness axioms IIn open-world assumption, individuals or disjointness axioms are missing at times. Therefore, we define a special assignment—"*Unknown*" of the similarities based on the set of individuals and disjointness axioms.

For individuals, we employ the string equivalent to judge the equality of them. When we calculate the similarity of concepts based on individuals across ontologies, we regard individuals of each concept as a set and use Ochiai coefficient[5] to measure the value. We use a boundary t to divide the value into three assignments (i.e., 1, 0 and *Unknown*). Assignment 1 (or 0) means that the pair matches (or mismatches). If the value ranges between 0 and t or the individuals of one concept are missing, then the assignment is set to *Unknown*.

For disjointness axioms, we utilize these axioms and subsumption relations within ontologies and define some rules to determine assignments of similarity. For example, x_1, y_1 and x_2 are concepts that come from O_1 and O_2. If x_1 matches x_2 and x_1 is disjoint with y_1, then y_1 is disjoint with x_2 as well as their descendants. The similarity also has three assignments. Assignment 1 (or 0) means the pair mismatches (or overlaps). If all the rules are not satisfied, the assignment is *Unknown*.

Using SPN to encode the similarities based on individuals and disjointness axioms Sum-Product Network is a directed acyclic graph with weighted edges, where variables are leaves and internal nodes are sums and products [8]. As shown in Figure 2, we designed a sum-product network denoted by S to encode above similarities and calculate the contributions. All the leaves in S, called indicators, are binary-value. M represents the contribution of individuals and disjointness axioms and indicators M_1, M_2, M_3 comprise all the assignments of it. $M_1 = 1$ (or $M_2 = 1$) means that the contribution is positive (or negative). If $M_3 = 1$, the contribution is *Unknown*. Similarly, Indicators D_0, D_1, I_1, I_2, I_3 correspond to assignments of the similarities based on the set of individuals and disjointness axioms. The concrete assignment metrics are listed in Table 1–2 and the assignment metric of M is similar to the metric of similarity D.

[5] https://en.wikipedia.org/wiki/Cosine_similarity

Table 1: Metric for Similarity D

Assignments	Indicators
$D = 1$	$D_0 = 0, D_1 = 1$
$D = 0$	$D_0 = 1, D_1 = 0$
$D =$ *Unknown*	$D_0 = 1, D_1 = 1$

Table 2: Metric for Similarity I

Assignments	Indicators
$I = 1$	$I_1 = 1, I_2 = 0, I_3 = 0$
$I = 0$	$I_1 = 0, I_2 = 1, I_3 = 0$
$I =$ *Unknown*	$I_1 = 0, I_2 = 0, I_3 = 1$

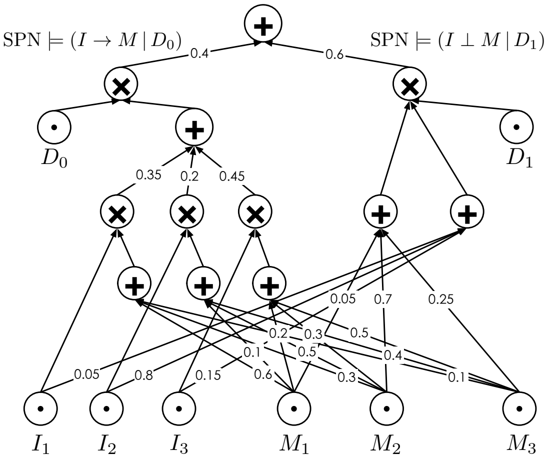


Fig. 2: The designed SPN : When $D_0 = 1, D_1 = 0$, it means that the distribution of M depends on the distribution of I; When $D_0 = 0, D_1 = 1$, the distributions of M and I are independent.

With the MAP inference in SPN [8], we can obtain the indicators' value of contribution M. Precisely, the MAP inference in SPN contains three steps. Firstly, replace sum nodes with max nodes. Secondly, with the bottom-up method, each max node can get a maximum weighted value. Finally, the downward pass starts from the root node and recursively selects the highest-value child of each max node, then the indicators' value of M are obtained. Moreover, even if the set of individuals or disjointness axioms are missing at times, we can also calculate the contribution M by MAP inference. Assumed $I = 1$, $D =$ *Unknown* for one pair, we can obtain $I_1 = 1, I_2 = 0, I_3 = 0, D_0 = 1, D_1 = 1$ with defined similarities and assignment metrics of SPN. As contribution M is not given, so we need to set $M_1 = 1, M_2 = 1, M_3 = 1$. After MAP inference, we observe $M_1 = 1$. It means that the contribution is positive. Besides, it can also infer $D_0 = 1$, which means the individuals of matching pair overlap.

As the network S is complete and decomposable, the inference in S can be computed in time linear in the number of edges [9]. Therefore, MAP inference is tractable.

Combining the lexical similarity and the contribution calculated by SPN Considering the range of lexical similarity, we introduce a scaling factor α to limit the contribution of lexical similarity. It can help us to analyze the sources from different contributions. The SPN-based similarity denoted by S_0 is defined in Eqs 1, which is calculated according to the indicators' value of M and D.

$$S_0(x_1, x_2) = \begin{cases} 0 & M_2 = 1, D_1 = 1 \\ \alpha * lexSim(x_1, x_2) + \lambda & M_1 = 1, D_0 = 1 \\ \alpha * lexSim(x_1, x_2) - \lambda & M_2 = 1, D_0 = 1 \\ \alpha * lexSim(x_1, x_2) & M_3 = 1, D_0 = 1 \end{cases} \tag{1}$$

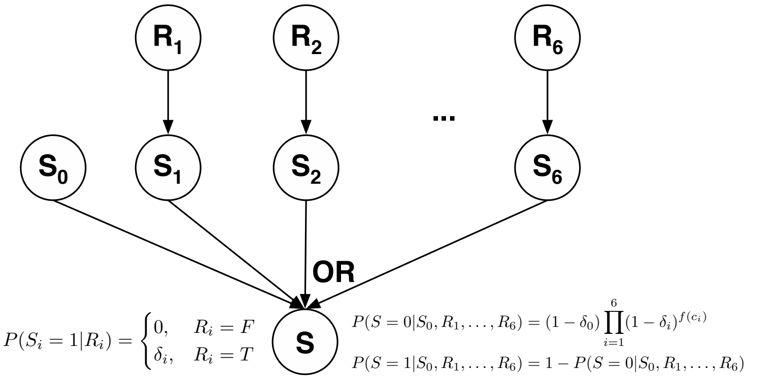
where λ is a contribution factor that represents the contribution based on disjointness axioms and the set of individuals. If contribution is positive (or negative) and pair overlaps, the SPN-based similarity is equal to the scaled lexical similarity adding (subtracting) λ. If the contribution is *Unknown* and pair overlaps, the SPN-based similarity is equal to the scaled lexical similarity. If the pair mismatches, then the inferred contribution is negative and the SPN-based similarity is set to 0.0.

Using Noisy-Or model to encode probabilistic matching rules As listed in Table 3, we utilize probabilistic matching rules to describe the influences among the related pairs across ontologies.

Table 3: The probabilistic matching rules between entity pairs

ID	Category	Probabilistic matching rules
R_1	Class	two classes probably match if their fathers match
R_2	Class	two classes probably match if their children match
R_3	Class	two classes probably match if their siblings match
R_4	Class	two classes asserted in domain probably match if related objectproperties match and range of these property match
R_5	Class	two classes asserted in range probably match if related objectproperties match and domain of these properties match
R_6	Class	two classes asserted in domain probably match if related dataproperties match and the type of these properties are the same

Considering the matching probability of one pair, we observe that the condition of each rule has two values (i.e., T or F) and all the matching rules are conditional independent when the value of this pair is given. Moreover, all the matching rules are conducive to improving the matching probability of this pair. Therefore, we utilize the noisy-or model [1] to encode them.



$$P(S_i = 1|R_i) = \begin{cases} 0, & R_i = F \\ \delta_i, & R_i = T \end{cases}$$

$$P(S = 0|S_0, R_1, \ldots, R_6) = (1 - \delta_0) \prod_{i=1}^{6} (1 - \delta_i)^{f(c_i)}$$

$$P(S = 1|S_0, R_1, \ldots, R_6) = 1 - P(S = 0|S_0, R_1, \ldots, R_6)$$

Fig. 3: The network structure of noisy-or model designed in GMap

Figure 3 shows the designed noisy-or model applied in concept pairs and the extension to property pairs is straight-forward, where R_i corresponds to the ith rule and S_i is the conditional probability depended on the condition of R_i. S_0 represents the SPN-based similarity which is a leak probability [1]. We can easily calculate the matching probability of each pair, $P(S = 1|S_0, R_1, ..., R_6)$, according to the formulas listed in Figure 3, where c_i is the count of satisfied R_i and the sigmoid function $f(c_i)$ is used to limit the upper bound of contribution of each R_i.

As the inference in the noisy-or model can be computed in time linear in the size of nodes [1], so GMap can keep inference tractable in the whole matching process.

1.3 Adaptations made for the evaluation

There are two kinds of parameters that need to be set. One mainly comes from networks, and it is set manually based on some considerations [10]. The others are adapted by I3CON data set[6] such as scaling factor (α), contribution factor (λ) in Eqs 1 and threshold (θ). Nevertheless, we do not make any specific adaptation for OAEI 2021 evaluation campaign, and all parameters are the same for all the tracks.

1.4 Link to the system and parameters file

The latest version of GMap in maven project can be downloaded by google drive [7], which is implemented by MELT platform [11]. In addition, GMap is also an open source project in https://github.com/liweizhuo001/GMap1.1.

2 Results

In this section, we present the results of GMap achieved on OAEI 2021. Our system mainly focuses on Conference, Anatomy.

[6] http://www.atl.external.lmco.com/projects/ontology/i3con.html

[7] https://drive.google.com/file/d/1kq8ntRVQclFF-TZOb6AesqJZ7e0P-h7F/view?usp=sharingand

2.1 Conference

Conference track contains sixteen ontologies from the conference organization domain. According to the crisp evaluation based on the main (blind) reference alignment in Conference Track, the results of GMap and other top-ranked matching systems are listed in Table 4.

Table 4: Results of GMap and other top-ranked matching systems in Conference track

Matcher	Precision	Recall	F.5-Measure	F1-Measure	F2-Measure
AML	0.78	0.62	0.74	0.69	0.65
LogMap	0.76	0.56	0.71	0.64	0.59
GMap	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**
ATMatcher	0.69	0.51	0.64	0.59	0.54
Wiktionary	0.66	0.53	0.63	0.59	0.55

Overall, GMap ranked 3rd of the 14 participants in terms of F1-Measure, which outperforms others in recall except for AML, but its precision is lower than theirs. There are mainly two reasons. One is the lexical similarity which combines the similarities based on edit-distance, external lexicons and TF-IDF with the max strategy. The other is the noisy-or model, which is hard to describe the negative effect on pairs matching [1]. Both of them would retain some false positive matches after the matching finish. Especially in property pairs, even though their domains and ranges mismatch, GMap can not describe this negative impact. Therefore, employing alignment debugging techniques [12–18] are comparatively ideal solutions to deal with this issue.

2.2 Anatomy

The goal of the Anatomy track is to find an alignment between the Adult Mouse Anatomy (2744 classes) and a part of the NCI Thesaurus (3304 classes). The results of GMap and other top-ranked matching systems are listed in Table 5.

Table 5: Results of GMap and other top-ranked matching systems in Anatomy Track

Matcher	Runtime (s)	Size	Precision	F1-Measure	Recall	Recall+	Coherent
AML	32	1471	0.956	0.941	0.927	0.81	+
LogMapBio	1043	1586	0.874	0.894	0.914	0.773	+
LogMap	7	1402	0.917	0.881	0.848	0.602	+
TOM	15068	1313	0.933	0.866	0.808	0.525	−
GMap	**2362**	**1344**	**0.916**	**0.861**	**0.812**	**0.534**	−
Fine-TOM	2647	1315	0.916	0.851	0.794	0.490	−
Wiktionary	493	1194	0.956	0.843	0.753	0.347	−

As a result, GMap ranked 5th of the 13 participants in terms of F1-Measure. We analyze that lexical-based module and simplified combination strategy may become the main bottlenecks of GMap. Benefited from the thesauruses (e.g., UMLS) and optimized combination strategy, most top-ranked systems can obtain better performances in ontology matching tasks. As mentioned above, most systems (e.g., AML, LogMap) employ the techniques of mapping validation, which is helpful to improve the quality of alignment further. Relatively, these techniques are not employed in the current version, and we leave this issue for future work.

3 General comments

3.1 Comments on the results

GMap [6] achieved qualified results in its second participation in OAEI, which is competitive with several top-ranked systems in main OAEI tracks. Benefited from SPN and the noisy-or model, the quality of alignment can be improved further compared with the original lexical similarity. However, some weaknesses still remain. For example, the alignment incoherence of GMap is still unsolved, which influences the performances of GMap. In addition, it is important for us to consider the efficiency of GMap, such as running time and memory usage tailored for large-scale ontologies.

3.2 Discussions on the way to improve the proposed system

GMap still has a lot of room for improvement. Employing mapping validation techniques is helpful to solve the alignment incoherent and reduce some false positive matches in final alignments. In addition, seeking available data sets to learn parameters of the sum-product network and the noisy-or model is also one direction of our future works.

4 Conclusion

In this paper, we have presented GMap and its results of two tracks (i.e., Conference, Anatomy) on OAEI 2021. The results indicate that GMap is competitive with the top-ranked systems by means of combining some special graphical models (i.e.,SPN, Noisy-or model). On the other hand, for those disadvantages exposed, we discuss the possible solutions. In the future, we would like to participate in more tracks (e.g., Multifarm, Complex, Interactive matching) and hope to solve the efficiency of large ontology matching tasks.

References

1. Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
2. Prasenjit Mitra, Natasha F Noy, and Anuj Rattan Jaiswal. OMEN: A Probabilistic Ontology Mapping Tool. In *Proceedings of the 4th International Semantic Web Conference*, pages 537–547. Springer, 2005.
3. Sivan Albagli, Rachel Ben-Eliyahu-Zohary, and Solomon E Shimony. Markov network based ontology matching. *Journal of Computer and System Sciences*, 78(1):105–118, 2012.
4. Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A Probabilistic-Logical Framework for Ontology Matching. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. AAAI Press, 2010.

5. Chuncheng Xiang, Baobao Chang, and Zhifang Sui. An Ontology Matching Approach Based on Affinity-Preserving Random Walks. In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the 24th International Join Conference on Artificial Intelligence*, pages 1471–1478. AAAI Press, 2015.
6. Weizhuo Li. Combining sum-product network and noisy-or model for ontology matching. In *Proceedings of the 10th International Workshop on Ontology Matching*, pages 35–39. CEUR-WS.org, 2015.
7. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Science & Business Media, 2013.
8. Hoifung Poon and Pedro M. Domingos. Sum-Product Networks: A New Deep Architecture. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 337–346. AUAI Press, 2011.
9. Robert Gens and Domingos Pedro. Learning the Structure of Sum-Product Networks. In *Proceedings of The 30th International Conference on Machine Learning*, pages 873–880. JMLR.org, 2013.
10. Li Ding and Tim Finin. Characterizing the semantic web on the web. In *Proceedings of the 5th International Semantic Web Conference*, pages 242–257. Springer, 2006.
11. Sven Hertling, Jan Portisch, and Heiko Paulheim. MELT - matching evaluation toolkit. In *Semantic Systems. The Power of AI and Knowledge Graphs - 15th International Conference, SEMANTiCS 2019*, pages 231–245, 2019.
12. Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, Universitätsbibliothek Mannheim, 2011.
13. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-Based and Scalable Ontology Matching. In *Proceedings of the 10th International Semantic Web Conference*, pages 273–288. Springer, 2011.
14. Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco M Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PloS one*, 10(12):1–19, 2015.
15. Weizhuo Li, Songmao Zhang, and Guilin Qi. A graph-based approach for resolving incoherent ontology mappings. *Journal of Web Intelligence*, 16(1):15–35, 2018.
16. Silvana Castano, Alfio Ferrara, Davide Lorusso, Tobias Henrik Näth, and Ralf Möller. Mapping validation by probabilistic reasoning. In *Proceedings of the 5th European Semantic Web Conference*, pages 170–184. Springer, 2008.
17. Jan Noessner and Mathias Niepert. ELOG: A probabilistic reasoner for OWL EL. In *Proceedings of the 5th International Conference on Web Reasoning and Rule Systems*, pages 281–286. Springer, 2011.
18. Weizhuo Li and Songmao Zhang. Repairing mappings across biomedical ontologies by probabilistic reasoning and belief revision. *Knowledge-Based Systems*, 209:106436, 2020.