GOMMA Results for OAEI 2012

Anika Groß, Michael Hartung, Toralf Kirsten, and Erhard Rahm

Department of Computer Science, University of Leipzig, Germany
{gross,hartung,tkirsten,rahm}@informatik.uni-leipzig.de

Abstract. We present the OAEI 2012 evaluation results for the matching system GOMMA developed at the University of Leipzig. The original application focus of GOMMA has been the life science domain but as a generic tool it can also match ontologies from other areas. It could thus participate in all OAEI tracks running on the SEALS platform. GOMMA supports several methods for efficiently matching large ontologies in particular parallel matching on multiple cores or machines, reducing the search space as well as reusing and composing previous mappings to related ontologies.

1 Presentation of the system

1.1 State, purpose, general statement

GOMMA (**G**eneric **O**ntology **M**atching and Mapping **Ma**nagement) [6] is a comprehensive infrastructure to manage and analyze the evolution of life science ontologies and mappings [4]. It includes a generic component to semantically align (match) ontologies. GOMMA is able to match very large ontologies as common in the life sciences. To deal with large ontologies GOMMA provides several scalable match techniques:

1. Parallel ontology matching on multiple computing nodes and CPU cores [2],
2. Indirect computation of ontology mappings by reusing and composing previously determined ontology mappings via intermediate ontologies [3], and
3. A newly introduced blocking approach to reduce the search space by restricting matching to overlapping ontology parts.

These techniques all support efficiency, in particular reduced computation times. The latter two approaches can also improve match quality. While the original focus of GOMMA has been in the life science domain, the match component is generic. We could thus participate in all 2012 match problems of the Ontology Alignment Evaluation Initiative (OAEI)[1] running on the SEALS platform.

1.2 GOMMA Matching Workflow

The GOMMA matching workflow used for OAEI 2012 is displayed in Fig. 1. In the following, we describe its three main phases, namely the initial phase (including the new blocking strategy), the matching phase as well as a set of postprocessing steps.

[1] http://oaei.ontologymatching.org

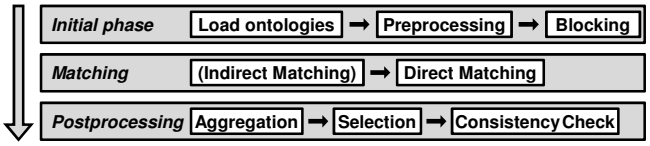

Fig. 1. GOMMA matching workflow for OAEI 2012

Generally, the input of the matching are two ontologies, source O_1 and target O_2, each consisting of concepts (classes, properties) as well as a structure (relationships between concepts, e.g., is_a, part_of). Internally, ontologies are represented as rooted, acyclic graphs. A concept has different attributes such as its name or a set of synonyms. The output of the matching workflow is a mapping M consisting of a set of correspondences whereby each correspondence has a similarity value denoting the strength of the connection between two concepts c_1 and c_2: $M = \{(c_1, c_2, sim) \,|c1 \in O_1, c2 \in O_2\}$.

Initial Phase and Blocking In the initial phase we first parse and *load the ontologies*. In this step, we assign all information relevant for matching to concepts, in particular name, synonyms, comments and instances. Note, that some attributes are multi-valued, e.g., there can be several synonyms or instances per concept. The information is stored within text attributes and used for string-based match comparisons.

During *preprocessing* we also check the language of attribute values (using xml:lang of rdfs:label). In case it is different from English we translate the term to English and add it as a new synonym to the concept. We used a free translation API[2] to automatically translate non-English terms. Using this facility, we iteratively established a dictionary to store the retrieved synonyms. All concept attribute values are further *normalized*, i.e., we remove delimiters and stop words, and normalize strings to lower case.

In the initial phase, we further apply a *blocking* strategy to reduce the number of comparisons for large ontologies. There have been various approaches to reduce the search space for large scale matching (see [7] for a recent survey). Our current approach is different and focuses on "asymmetric" match problems where a specific ontology is matched to a broader ontology from which only a part is relevant. An example for such an asymmetric match problem is the alignment of a pure anatomy ontology such as the Foundational Model of Anatomy (FMA) against a broad biomedical ontology such as NCI Thesaurus covering anatomy in one part. Another scenario for linked data is to match a domain-specific ontology, e.g. from the geographical domain, to the broad DBpedia ontology.

To deal with such match problems we aim at automatically identifying the relevant part of the broader ontology and to match only this part with the more specific, and typically smaller ontology. This blocking strategy is expected to (1) dramatically improve efficiency in applicable cases and (2) improve match quality (in particular precision) due to fewer false positive correspondences. The blocking strategy is based on an initial mapping and works in the following steps:

[2] http://mymemory.translated.net/

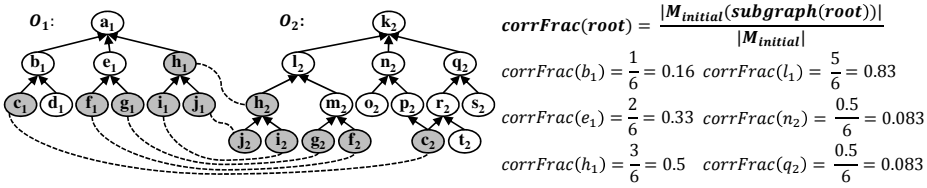

Fig. 2. Blocking ontology subgraphs

1. Determine an initial mapping $M_{initial}$ using a very efficient match method, e.g., exact name matching with hashed attribute values (applied in OAEI 2012) or the reuse of precomputed mappings.
2. Identify a set of subgraph *roots* below the top root. Determine the number of correspondences from $M_{initial}$ per subgraph root, $|M_{initial}\,(subgraph\,(root))|$, by propagating the correspondence counts from the leaf level upwards. In case of multiple inheritance, the correspondence count is partially propagated upwards the ontology structure (for the example in Fig. 2 this is done for O_2 concept c_2).
3. For each $root$ compute a correspondence fraction $corrFrac(root)$ that is the number of correspondences assigned to the root $|M_{initial}\,(subgraph\,(root))|$ divided by the overall size of the initial mapping $|M_{initial}|$ (see Fig. 2).
4. Select the most valuable root(s) with a $corrFrac$ above a given threshold. All concepts in the subgraph of this root will be used for matching, other concepts will not be compared. If no root exceeds the threshold, blocking is not applied, i.e., the whole ontology needs to be matched since no dominating part is found.

Fig. 2 illustrates the approach for two ontologies and a set of predetermined correspondences. To choose a promising subgraph for matching, we consider roots on the second ontology level (b_1,e_1,h_1 for O_1 and l_2,n_2,p_2 for O_2). Applying a $corrFrac$ threshold of 0.7 means that a subgraph must cover at least 70% of all initial correspondences. This is only the case for root l_2 in O_2, i.e., in the example only O_2 can be partitioned so that the whole O_1 will be matched with the l_2-subgraph of O_2.

Matching GOMMA's matching component allows for *direct* and *indirect matching* of ontologies. Direct match strategies involve internal ontology knowledge like concept-associated or structural information. By contrast, our indirect matching is based on the composition of existing mappings to intermediate (background) ontologies. To efficiently match especially large ontologies, we further parallelize the direct matching process. In the following we describe the match strategies used for OAEI 2012.

To directly match two ontologies we combine up to three different matchers. We always apply a name/synonym matcher that determines the maximal string similarity for the names and multi-valued synonyms per concept pair. In case the necessary information is available, we also apply a comment matcher and instance matcher. GOMMA supports further matchers such as structural matchers [6] but we found them less effective for life science ontologies. We thus did not include them in our default strategy used for all OAEI tasks.

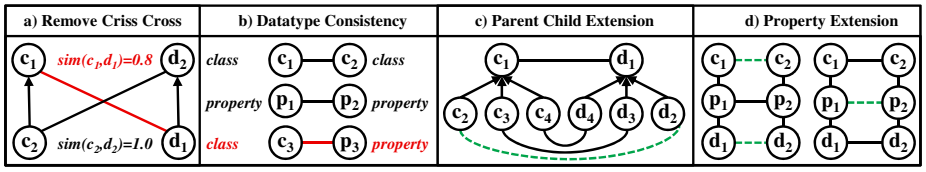

Fig. 3. Consistency checking. Red continuous (green dotted) line = remove (add) correspondence

To efficiently match large ontologies we apply intra-matcher parallelization [2]. For this purpose, we uniformly partition the input ontologies into smaller fragments with the same number of concepts and we solve the fragment-level match tasks in parallel. This parallelization is made easy since for the applied matchers all information used for matching is directly associated to the concepts.

To improve match quality we further apply an indirect composition-based match approach [3]. This approach allows the reuse of existing high quality mappings to efficiently match two so far unmatched ontologies. For example, anatomy ontologies O_1 and O_2 can be matched by composing two mappings O_1 - H and H - O_2 with an intermediate "hub" ontology H, e.g. UMLS. For OAEI we used our direct match strategy to precompute several mappings from the source and target ontology via different intermediate ontologies and combine these composed mappings. Since the resulting mapping may still be incomplete, we identify the unmatched source and target concepts and match them directly to extend the result mapping.

Postprocessing The main task of this phase is the combination or aggregation of the directly and indirectly determined mappings and to select the most likely correspondences from the combined mapping. Before this, we first *filter* out all correspondences per mapping with a similarity below a specified threshold. To combine several mappings we take their union and *average* the similarity values per correspondence. We then apply a *maxDelta selection* [1] for the remaining correspondences. This approach returns for each concept only those correspondences with the maximal similarity value or those within a small delta distance to the maximal value, i.e., we only keep the best correspondences for each source and target concept.

We further apply techniques to improve the consistency of mappings by removing presumably wrong and by adding presumably missing correspondences. We currently check four simple constraints; additional checks may be added in the future to further improve consistency. Fig. 3 shows small exemplary scenarios for each *consistency checker*. The first two conditions check situations that may result in a removal of correspondences (to improve precision), similar as in systems like ASMOV [5]. The two other conditions can lead to the addition of correspondences (to improve recall).

First, correspondences must meet a so-called Criss Cross condition (Fig. 3a), i.e., we eliminate conflicting correspondences (c_1, d_1) and (c_2, d_2) where c_2, is a child of c_1, but d_1 a child of d_2 (or vice versa). One can either remove both correspondences or only remove the one with the lower similarity value. Second, we check the datatype consistency (Fig. 3b). In particular, we remove correspondences between properties and classes, i.e., only class-class / property-property correspondences are allowed.

The first rule to extend the mapping checks whether two concepts match but only a subset of their children (Fig.3c). Here, we add a correspondence for the most similar, unmatched pair of children. Finally, in case of matching properties we add correspondence(s) for the domain/range classes if they have no corresponding class, or we conclude a property match if both, domain and range class, have correspondences (Fig.3d).

1.3 Adaptations made for the evaluation

GOMMA's modular structure helped us to adapt the system to work for the OAEI tasks. One major effort was the adaptation of the ontology import mechanism. We implemented a new SAX-based ontology parser which can be used to load multiple ontologies in parallel via threading. Usually, parallel execution of match workflows in GOMMA requires multiple compute nodes. To better utilize the single machine used for the evaluation, we adapted parallel matching to the use of threading to distribute several match jobs on all available CPU cores on only one machine.

1.4 Link to the system and parameters file

GOMMA is available at `http://dbs.uni-leipzig.de/GOMMA`.

2 Results

We now present and discuss the evaluation results of GOMMA in the OAEI 2012 campaign. We participated in six tracks: Anatomy, Large Biomedical Ontologies, Benchmarks, Library, Conference and Multifarm. Detailed results and descriptions about the used computation environments are provided on the OAEI 2012 result page[3].

2.1 Anatomy and Large Biomedical Ontologies

Anatomy Results For the Anatomy Track two real-world anatomy ontologies namely the Mouse Anatomy (2,744 concepts) and the anatomy part of the NCI Thesaurus (3,304 concepts) should be matched. GOMMA achieves a good F-Measure value of ≈87% in a short amount of time (17 sec.) (Fig.4). In a separate configuration using background knowledge (GOMMA-bk) we apply indirect (composition-based) matching [3] using mappings to three intermediate ontologies (UMLS, Uberon or FMA). By doing so we could increase F-Measure to 92.2% in a reduced execution time (15 sec.).
Large Biomedical Ontologies Results This track was extended w.r.t. its first evaluation in OAEI2011.5. In addition to matching FMA and NCI, two new tasks namely FMA–SNOMED and SNOMED–NCI were introduced. All tasks are divided into three subtasks where *small* and *large* ontology fragments or *whole* ontologies need to be matched. In this track GOMMA's approaches (*composition-based matching*, *parallel matching* and *blocking*) helped to achieve high quality match results with relatively low execution times. Since all ontologies consist of more than 4,000 (and up to 120,000)

[3] OAEI 2012 campaign: http://oaei.ontologymatching.org/2012/results/

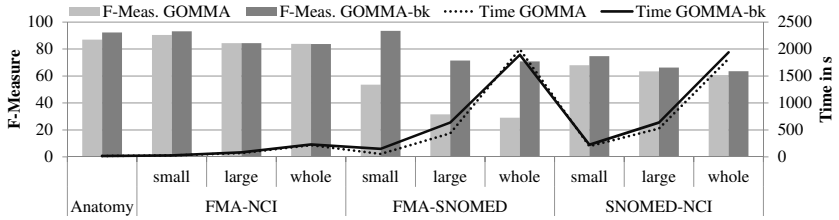

Fig. 4. Evaluation results for the Anatomy and Large Biomedical Ontologies tracks (FMA–NCI, FMA–SNOMED, SNOMED–NCI).

concepts, we apply our blocking strategy (Sec. 1.2) to reduce the overall runtime. Blocking leads to the selection of subgraphs for NCI (FMA–NCI task) and SNOMED (FMA–SNOMED, SNOMED–NCI) thereby reducing the search space by factor 2–6.

The results are summarized in Figure 4. The shown F-Measure values are based on the UMLS reference mapping. There are further results based on refined reference mappings available[4]. As for the Anatomy task, using background knowledge increases the match quality substantially with still acceptable runtime. The best results with 93-94% F-Measure are achieved for the *small* FMA-related subtasks for GOMMA-bk (mappings to UMLS, Uberon). The *small* SNOMED–NCI task seems to be more challenging (≈75% F-Measure with bk). Comparing GOMMA and GOMMA-bk for FMA-SNOMED, we observe a very strong improvement of ≈40% F-Measure when applying *composition-based matching*. For the *whole* FMA-SNOMED (SNOMED-NCI) task we achieve a good F-Measure of 71% (64%) thereby consuming ≈ 30 min computation time. Overall GOMMA-bk takes slightly longer than GOMMA except for the *whole* FMA-SNOMED task. In this case the result of composition-based matching might already cover a higher part of the input ontologies and we do not need to execute a direct matching on whole ontologies.

2.2 Benchmarks and Library

Benchmarks Track Results This track is subdivided into five sub-tracks namely Biblio, Finance and Benchmark 2–4. There are multiple match tasks per sub-track where one source ontology is compared with a number of systematically modified target ontologies. Overall, GOMMA achieved F-Measure values in the range between 60–70% with favoring precision over recall. The recall results are slightly better than in the 2011.5 campaign due to new postprocessors to extend the mapping as described in Sec. 1.2. Using our new thread-based parser, we solved each of the problems in less than one minute.

Library Results In this new, real-world match task the two ontologies STW and TheSoz consisting of about 6,500 and 8,500 concepts need to be aligned. Both ontologies provide a lightweight vocabulary for economic/social science topics and are used in libraries for indexing and search. GOMMA achieved a high recall of ≈91%, however

[4] oaei.ontologymatching.org/2012/results/largeBioMed/

the precision was low (54%). The resulting F-Measure of 67% is comparable to the Benchmark results. Since the vocabularies provide a huge number of labels and synonyms (≈5 per concept), our name/synonym matcher had to evaluate 40,000x32,000 ≈1.3 billion comparisons leading to a runtime of ≈13min. on a 2 core machine.

2.3 Conference and Multifarm

Conference Results The Conference track consists of 16 small ontologies from the domain of conference organization. Each ontology must be matched against each other. In summary, we required about 91 seconds to solve the complete task. The match quality was evaluated against an original (ra1) as well as entailed reference alignment (ra2). For both evaluations we achieved F-Measure values better than the Baseline2 results (61% for ra1 and 56% for ra2). Compared to the 2011.5 campaign, we were able to increase match quality by about 3% in terms of F-Measure (for ra2). In particular, we improved recall by applying the postprocessing methods described in Sec. 1.2.
Multifarm Results The Multifarm task is an extension of the Conference task since conference ontologies in nine different languages (e.g. English, Russian, Chinese) should be matched among each other (36 language pairs). We performed a translation approach (see Sec. 1.2) as a preprocessing step to translate non-English labels into English ones, so that we can afterwards match the translated ontologies with each other. Overall GOMMA required 35 minutes to solve all 36 match problems, i.e. less than one minute per language-pair. The average F-Measure is 35% with an average recall (precision) of 31% (45%). The best results emerge for language pairs where one language is English or for pairs with similar languages, e.g., Spanish to Portuguese with 47% F-Measure.

3 General comments

3.1 Comments on the results and future improvements

The evaluation confirmed that GOMMA has the following strengths:

- Scalable matching of ontologies of different size by performing blocking, parallel matching and mapping composition. A high efficiency and effectiveness is especially achieved in the Anatomy and Large Biomedical Ontologies tracks.
- Substantial improvement of match quality by using domain knowledge, in particular by composing mappings with domain-specific hub ontologies or by applying multi-language translation services for improved synonyms.

We plan to further improve the consistency of the result mapping by applying additional checks during postprocessing. Moreover, we like to apply a more general blocking method to boost both the runtime and match quality (precision) for additional match problems.

3.2 Comments on the OAEI 2012 procedure

Measuring the overall runtimes per match task and system is useful but insufficient to identify and analyze underlying bottlenecks. For example, it would be helpful to see

the time requirements for major phases such as import vs. match. When evaluating scalability (e.g., between a 1-core and a 4-core CPU) the import time might be constant whereas the real match time is reduced with good speed-up. Moreover, it might be interesting to compare the runtime of tools over different years. For each participating tool, available older versions might be re-executed on the currently used machine such that execution times are comparable.

Tools developed by co-organizers of OAEI tracks should not be considered in the official evaluation. This is to avoid the possible suspicion that the design of the match tasks might be tailored to the co-organizers' tools or that the configuration of these tools might be favored by the co-organizers' access to critical data that is unknown for other participants (e.g., Library track gold standard).

4 Conclusion

The participation in six tracks of OAEI 2012 showed that GOMMA is able to efficiently and effectively match ontologies of different size. Especially in the Anatomy and Large Biomedical Ontologies tracks GOMMA's techniques such as *composition-based matching*, *parallel matching* and *blocking* showed to be valuable for a scalable ontology matching. We envision further improvements of GOMMA, e.g. by applying a more general blocking strategy or by additional consistency checks for result mappings.

5 Acknowledgement

Funding: This work is supported by the German Research Foundation (DFG), grant RA 497/18-1 ("Evolution of Ontologies and Mappings").

References

1. Do, H., Rahm, E.: COMA: a system for flexible combination of schema matching approaches. In: Proc. of the 28th Intl. Conf. on Very Large Data Bases (VLDB). pp. 610–621 (2002)
2. Gross, A., Hartung, M., Kirsten, T., Rahm, E.: On matching large life science ontologies in parallel. In: Data Integration in the Life Sciences. pp. 35–49. Springer (2010)
3. Gross, A., Hartung, M., Kirsten, T., Rahm, E.: Mapping composition for matching large life science ontologies. In: Proc. of the 2nd Intl. Conf. on Biomedical Ontology (ICBO), CEUR Workshop Proceedings, CEUR-WS.org/Vol-833/ (2011)
4. Hartung, M., Kirsten, T., Rahm, E.: Analyzing the evolution of life science ontologies and mappings. In: Data Integration in the Life Sciences (DILS). pp. 11–27. Springer (2008)
5. Jean-Mary, Y., Shironoshita, E., Kabuka, M.: Ontology matching with semantic verification. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 235–251 (2009)
6. Kirsten, T., Gross, A., Hartung, M., Rahm, E.: Gomma: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. Journal of Biomedical Semantics 2, 6 (2011)
7. Rahm, E.: Towards large-scale schema and ontology matching. Schema matching and mapping pp. 3–27 (2011)