Results of CANARD in OAEI 2024

Guilherme Sousa[1], Rinaldo Lima[2] and Cassia Trojahn[3]

[1]IRIT: Institut de Recherche en Informatique de Toulouse, France
[2]Universidade Federal Rural de Pernambuco, Recife, Brazil
[3]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

Abstract

This paper presents the 2024 results of an enhanced version of the CANARD system, which integrates Large Language Models (LLMs) to address the challenges of complex alignments. By leveraging LLM-based embeddings, the system better captures semantic and contextual relationships, improving both precision and coverage. Four architectural settings – Label Embedding Similarity (LES), Embeddings of SPARQL Query (ESQ), Subgraph Embeddings (SE) and Instance Embeddings (IE) – were explored to improve the alignment quality. Experiments on the Populated Conference dataset from the OAEI Complex Track demonstrate improvements over baseline approaches, with an increase in F-measure up to 45% in some cases. However, challenges such as runtime overhead in IE and noise in SE components were identified, where future work can explore better aggregation techniques or fine-tuned LLMs.

Keywords

Complex Ontology Matching, Embeddings, LLM.

1. Presentation of the System

Due to the textual composition of ontologies, the capacity for language understanding is an important feature for matching ontologies. Large Language Models (LLMs) show an increasing capacity for textual comprehension in understanding and generation tasks. In the simple ontology matching task, these models are widely explored and show increasing performance [1]. However, few approaches have explored LLMs or embeddings in complex ontology matching. The system here is an extension of a previous version of CANARD (Complex Alignment Need and A-box-based Relation Discovery) [2]. This new version replaces the main similarity computation modules of CANARD, which are mainly lexical similarity computations, with embedding similarity generated by LLMs. With this replacement, the matcher is supposed to have an increasing capacity to retrieve better complex correspondences while being able to filter incorrect ones.

The base implementation of CANARD uses CQAs (Competency Questions for Alignment) to reduce the search space and receives a source CQA as input together with the two ontologies to be matched. CQAs are SPARQL queries used to retrieve instances in the source KG and check for `owl:sameAs`, `skos:closeMatch`, or `skos:exactMatch` predicates relating to those instances in the target KG. In case no predicate is found, an exact string matching is performed. With those instances in target KG, the subgraphs related to those instances are retrieved along with the entities' labels and descriptions. After that, a cartesian lexical similarity is computed between the labels of the entities present in the CQA and the ones retrieved from the subgraphs with a threshold filter that ignores the similarities below a threshold.

The new architecture of CANARD is composed of 9 steps (Figure 1). In step 1, a DL formula is extracted from the source SPARQL CQA. Then in step 2, the labels from the entities present in the source SPARQL CQA are extracted. In Step 3, the instances are retrieved by querying the source KG

OM-2024: The 19th International Workshop on Ontology Matching collocated with the 23rd International Semantic Web Conference (ISWC 2024), November 11th, Baltimore, USA.
*Corresponding author.
†First author has implemented and evaluated the system.
guilherme.santos-sousa@irit.fr (G. Sousa); rinaldo.jose@ufrpe.br (R. Lima); cassia.trojahn@irit.fr (C. Trojahn)
0000-0002-2896-2362 (G. Sousa); 0000-0002-1388-4824 (R. Lima); 0000-0003-2840-005X (C. Trojahn)

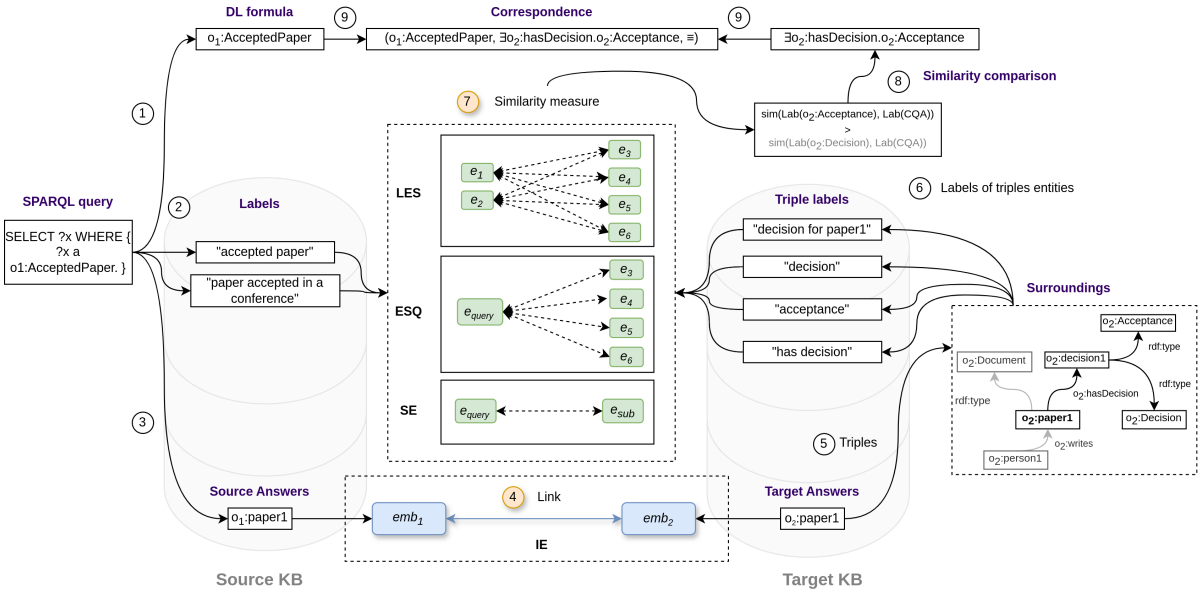

Figure 1: Overall steps of the system.

with the source CQA. In step 4, similar instances in the target KG are retrieved for those retrieved in step 3. In step 5, the subgraph of the target instances is retrieved. In step 6 the labels of the entities in the subgraphs are retrieved and in step 7 their similarity is measured against the labels retrieved in step 2. In step 8 the correspondences with summed similarity higher than a threshold are kept and in step 9 the final alignment is written in EDOAL format. Two main improvements have been implemented, corresponding to the two similarity evaluations in the instance matching step and in the subgraph similarity measure. By using embeddings to compute those similarities, the matcher improves its capacity to retrieve the relevant entities to each case. Also, since LLMs have increased language understanding, using embeddings generated by LLMs improves similarity computation.

1.1. Embeddings Generation

Multiple embedding applications are possible in the proposed architecture. Since subgraphs are involved in the similarity computation to the CQA entities (step 7 of Figure 1), some aggregation techniques are applied. For the similarity computation with embeddings three aggregation strategies were considered:

Label Embedding Similarity (LES) Entity labels from the source and target KGs are processed through pre-trained LLMs. Those LLMs have an associated tokenizer that splits the text into multiple tokens. Those tokens are then input to the LLM that generates one embedding for each token. The final embedding for each label is derived by averaging all embedding from the output of the model's last hidden layer. All embeddings from the CQA side are cross-compared with all tokens in the subgraph labels resulting in n:m comparisons. The similarities lower than a threshold are filtered out and the resulting ones are added to the final similarity.

Embeddings of SPARQL query (ESQ) In this setting the embeddings generated for all entities in the CQA (similar to the LES step) are averaged resulting in a single embedding for the CQA. Then this embedding is compared with the ones from the subgraph entities resulting in 1:m comparisons.

Subgraph embeddings (SE) For this setting, the embedding for the CQA is computed as in the ESQ setting, and the embeddings for the subgraph are aggregated. Two types of aggregation are considered depending on the type of the subgraph. If the CQA is unary (one variable in the SPARQL query) the corresponding subgraph is a triple composed of subject, predicate, object, and also the subject type and object type. For binary CQAs (two variables) the corresponding subgraph is a path that connects the entities retrieved in the variables.

In the instance matching step (step 4 Figure 1), an embedding is generated for all entities in the datasets

without the BNodes. Then, for any given entity the most similar ones by computing their embedding similarity. This setup is named **Instance Embeddings** (IE) and can be applied simultaneously with the CQA embedding settings LES, ESQ, and SE.

1.2. Adaptations Made for the Evaluation

Some adaptations were made to evaluate the system in the OAEI complex track. The matcher was evaluated in the Populated Conference dataset as it contains CQAs and instances as required by CANARD to execute and later by the evaluator to compare the results.

The Populated Conference dataset contains 5 populated ontologies and all pairs were evaluated with the proposed approach. For each pair, a range of similarity thresholds was evaluated to determine the optimal value for filtering alignments. Two distinct threshold values are needed in this approach, one for the CQA-related similarity and the other for the instance matching step. For the similarity in step 7 (Figure 1), values ranging between 0.5 and 1.0 incrementing by 0.1 were considered. For the instance matching step (step 4 in Figure 1), thresholds of 0.8, 0.85, and 0.9 were employed to verify the impact of the threshold in this step. Only a few thresholds were considered in the step 4 to reduce the number of combinations to be tested.

The embedding generation (before the main execution), involves processing all KG labels and instances through LLMs and is computationally intensive. To address this, embeddings were precomputed and cached for reuse during the matching process. The computation using LLMs was computed using GPU acceleration and the usage in the evaluation was done only with CPU. The tested models are GritLM-7B [3], sfr-mistral [4], Glove [5], and Stella-base [1].

The evaluation was automatically performed using the evaluator proposed in [6] that was used to evaluate the alignments of matchers in the complex track for the OAEI campaign. Two metrics from those available in this evaluator were selected. Coverage (query F-measure based on CQAs) and precision. Both metrics consider the comparison of instance sets.

Considering all parameter variations of input ontologies, embedding models, and thresholds, 1800 combinations of parameters were tested to identify the most promising configurations. This was followed by additional evaluations incorporating embeddings into the linking step for the top-performing settings. These adaptations allowed the system to demonstrate substantial improvements over baseline methods, achieving higher precision and coverage scores on the tested dataset.

1.2.1. Link to the System and Parameters File

The baseline approach can be found in https://framagit.org/IRIT_UT2J/ComplexAlignmentGenerator. The embedding variation used in this paper can be found at https://gitlab.irit.fr/melodi/ontology-matching/complex/canarde.

2. Results

In this section, the results of the evaluation of the Populated Conference dataset are presented. In the query-oriented evaluation, the GritLM-7B with the ESQ setting was the one with the highest query-oriented f-measure and precision. In the precision-oriented evaluation, the Stella-base model with ESQ setting and the instance embeddings IE setting with 0.85 threshold reaches higher results in all precision-oriented metrics. The results of this evaluation are presented in Table 1.

As shown in Table 1, LES and ESQ achieve the highest performance when LLMs are utilized. These configurations involve fewer embedding aggregations than SE. Notably, as the number of aggregations increases, such as in the SE configurations, the performance of all models decreases. This degradation can be attributed to the loss of semantic information and increased noise caused by combining embeddings without a weight transformation mechanism or filtering, such as those used in Graph Neural Networks

[1] https://huggingface.co/infgrad/stella-base-en-v2

	query oriented					precision oriented				
model	cls cqa	rec.	prec.	ovlp	f-m.	cls prec.	rec.	prec.	ovlp	f-m.
base (levenshtein)	0.35	**0.36**	0.47	0.35	0.47	0.21	0.26	0.26	0.28	0.26
GritLM-7B (LES)	**0.37**	0.32	0.68	**0.36**	0.68	0.36	0.39	0.38	0.40	0.39
sfr-mistral (i-LES)	0.37	0.32	0.67	0.35	0.67	0.36	0.39	0.38	0.39	0.39
GritLM-7B (ESQ)	0.37	0.32	**0.68**	0.35	**0.68**	0.36	0.39	0.38	0.40	0.39
sfr-mistral (i-ESQ)	0.37	0.32	0.67	0.35	0.67	0.36	0.39	0.38	0.39	0.39
glove (SE)	0.20	0.25	0.39	0.17	0.39	0.18	0.24	0.25	0.28	0.22
glove (i-SE)	0.21	0.25	0.40	0.18	0.40	0.18	0.23	0.25	0.28	0.22
stella-base (ESQ+IE 0.9)	0.30	0.30	0.64	0.25	0.64	0.38	0.41	0.40	0.41	0.40
stella-base (ESQ+IE 0.85)	0.30	0.29	0.62	0.24	0.62	**0.39**	**0.42**	**0.41**	**0.42**	**0.41**
GritLM-7B (ESQ+IE 0.9)	0.33	0.27	0.63	0.30	0.63	0.37	0.41	0.40	0.42	0.40

Table 1
Results for best models in each setting. *i* refers to ignore case version. The values near IE are the threshold in the link step. The columns were abbreviated for shortness where cls stands for classical precision and ovlp stands for overlap (comparison of instance sets).

[7]. Also, increasing the model size consistently improves performance across all configurations. However, the results among the LLMs don't diverge much. Another observation is that the IE setting enhances precision-oriented metrics across all models but in some cases, the results in the query-oriented evaluation are reduced.

Also, the improved architecture was compared with other matchers in the same dataset. The results of this comparison are presented in Table 2.

Matcher	Prec.	Coverage
Matcha-DL	-	-
AMLC	0.230	0.260
CANARD 2018	0.212	0.471
CANARD 2024 (Stella-base IE 0.85)	**0.389**	0.623
CANARD 2024 (GritLM-7B ESQ)	0.359	**0.679**

Table 2
Comparison of the proposed approach with other matchers. Precision in this table stands for classical precision and Coverage to classical - query F-measure coverage.

3. General Comments

The results demonstrate that the integration of Large Language Models (LLMs) enhances the performance of the CANARD framework in the Populated Conference dataset. The usage of embeddings increased the precision and F-measure by up to 45% over the baseline, showing their effectiveness in capturing semantic nuances. The LES and ESQ configurations were the most effective, and applying the instance embeddings also increased the performance of the matcher in the precision-oriented that is instance-based evaluation.

However, some weaknesses were still present for example the Instance Embeddings (IE) setting incurred significant computational overhead due to the exhaustive comparison of embeddings, particularly for large datasets. Also, the aggregation of subgraph embeddings used in the setting SE occasionally introduced noise, leading to reduced performance in comparison to other configurations.

Several improvements can address the identified weaknesses. The IE step can be optimized using approximate similarity measures or clustering techniques to reduce the search space. Techniques like weighted aggregation or Graph Neural Networks (GNNs) could improve the quality of subgraph embeddings by better preserving semantic relationships. Also, using domain-specific fine-tuning of LLMs on ontology-related tasks could further enhance the relevance of embeddings. Is also possible

to combine embedding-based methods with symbolic reasoning approaches to enhance the ability to capture complex logical relationships.

4. Conclusions

This paper presented an enhanced ontology matching system that integrates Large Language Models (LLMs) into the CANARD framework to deal with complex alignment tasks. The use of LLM-based embeddings improved performance in the conducted evaluation of the Populated Conference dataset. While the system excelled in capturing semantic relationships, challenges such as high runtime in certain configurations and reduced performance in subgraph embeddings highlight areas for future optimization. Proposed improvements include efficiency enhancements, fine-tuning of LLMs, and advanced aggregation techniques.

References

[1] S. Hertling, H. Paulheim, Olala: Ontology matching with large language models, in: K. B. Venable, D. Garijo, B. Jalaian (Eds.), Proceedings of the 12th Knowledge Capture Conference 2023, K-CAP 2023, Pensacola, FL, USA, December 5-7, 2023, ACM, 2023, pp. 131–139. URL: https://doi.org/10.1145/3587259.3627571. doi:10.1145/3587259.3627571.

[2] É. Thiéblin, O. Haemmerlé, C. Trojahn, Results of CANARD in OAEI 2020, in: P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh, C. Trojahn (Eds.), Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020, volume 2788 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 176–180. URL: https://ceur-ws.org/Vol-2788/oaei20_paper6.pdf.

[3] N. Muennighoff, H. Su, L. Wang, N. Yang, F. Wei, T. Yu, A. Singh, D. Kiela, Generative representational instruction tuning, CoRR abs/2402.09906 (2024). URL: https://doi.org/10.48550/arXiv.2402.09906. doi:10.48550/ARXIV.2402.09906. arXiv:2402.09906.

[4] R. Meng, Y. LIU, S. R. JOTY, C. XIONG, Y. ZHOU, S. YAVUZ, Sfr-embedding-mistral: Enhance text retrieval with transfer learning [salesforce ai research blog]. 2024, Available also from: https://blog.salesf orceairesearch. com/sfr-embedded-mistral (????).

[5] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL, 2014, pp. 1532–1543. URL: https://doi.org/10.3115/v1/d14-1162. doi:10.3115/V1/D14-1162.

[6] É. Thiéblin, O. Haemmerlé, C. Trojahn, Automatic evaluation of complex alignments: An instance-based approach, Semantic Web 12 (2021) 767–787. URL: https://doi.org/10.3233/SW-210437. doi:10.3233/SW-210437.

[7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Networks Learn. Syst. 32 (2021) 4–24. URL: https://doi.org/10.1109/TNNLS.2020.2978386. doi:10.1109/TNNLS.2020.2978386.