Enhanced Ontology Matching for Big Data Integration

Nesma Mahmoud¹,² and Hatem M Abdikader¹,³

¹ Information Systems Department, Menoufia University, Shebin Elkom, Egypt
² nesma.1115@ci.menofia.edu.eg, ³ hatem6803@yahoo.com

Abstract – ontology matching (OM) is a critical process for many disciplines. It aims at identifying the semantic correspondences among different ontologies that are merged for data integration. Unfortunately, OM still faces challenges, especially, in the big data integration (BDI) area. The high degree of semantic heterogeneity problem prevents the integration of relevant data and increased with large-scale ontologies of BDI. The quality of OM still needs more improvements to cope with BDI applications. So, this paper proposes a semantic OM approach called semantic matcher. It achieves the goals of semantic heterogeneity resolving and the quality improvement. It exploits the semantic similarity based on the word embedding model. The word embedding model provides efficient distribution semantic representation of domain words as vectors based on their context. The applicability of the proposed semantic matcher is evaluated through an experiment. The conference and anatomy gold standard datasets are evaluating the experimental results. Accuracy evaluates the quality through the precision, recall, and F-measure measures. Based on the experimental results evaluation, the proposed semantic matcher is promising and efficient in the semantic OM for BDI.

1. INTRODUCTION

OM finds correspondences (also called mappings) between different ontologies [1]. It results in a set of semantically related alignments that enable the integration and interoperability between those ontologies [2]. OM is the base and challenging process in BDI and other applications [3].

BDI is the core and the challenging task in unlocking big data values [4, 5]. Because, it provides unified view and access from heterogeneous data resides in many sources. Analysing the integrated data leads to more effective results of meaningful insights. Decision makers mainly depend on these meaningful insights in the best driving of their businesses. The semantic web usage increasing leads to the increasing of ontologies appearance in many different domains [5, 6, 7]. Ontology is a graph schema for representing information and knowledge of a certain domain. Big data characteristics, especially the volume, lead to the appeal of large-scale ontologies.

Ontologies still suffer from a high degree of semantic heterogeneity because they depend on the subjective view in building [1, 5, 8]. Also, they built with different methodologies by different expert people even in the same domain. OM is the effective way to resolves semantic heterogeneity. But OM still needs more improvement to overcome the semantic heterogeneity and to cope with BDI application [2, 8, 9].

Word embedding model provides distributed semantic representation of words from text based on their context [11]. It learns from the large text corpora of big data and provides a high-quality distributed representation of words as vectors. These representations were proved their success in many applications like recommendation system, NLP-based applications, spam filtering, etc. Word2Vec (W2V) model is one of the most used forms of word embedding that is exploited in this paper. W2V comprises efficient implementation of two models, skip-gram and Continuous Bag of Words (CBOW) [11, 12]. Both skip-gram and CBOW models have own algorithms and parameters that are tuned during the experiments.

So, in this paper, a semantic matcher is proposed for tackling OM challenges with BDI. It exploits the semantic similarity based on the word embedding model. The best semantic similarity results come from the dependency on the semantic representation of words based on their context that is provided by the word embedding model. Also, a strategy for dealing with matched single words and
sentences is also proposed in the semantic matcher. The semantic matcher targets the followings. Firstly, resolving the semantic heterogeneity of ontologies, especially, large-scale ontologies. Secondly, improving the matching quality for the schema integration application of BDI that require the matching result alignments to be correct and complete as much as they can be. In the evaluation, the gold standard datasets and reference alignments from the Ontology Alignment Evaluation Initiative (OAEI) contest are exploited.

The rest of the paper is organized as follows. Section 2 gives a review about the most related works achieved in improving the OM quality. Section 3 illustrates in detail the proposed semantic matcher. Section 4 demonstrates the experiment implementation details and discusses the results with their evaluation. Finally, the paper conclusion with future works are introduced in section 5 and after that the list of references.

2. RELATED WORK

The literature review focuses on the achieved works in the quality improvement of OM [3, 7, 9, 12, 13]. Also, it focuses on how new techniques, especially, deep learning techniques, were utilized in OM.

The work [7] was utilized the word embedding model, from deep learning, in the OM area. It represents the first utilizing of the word embedding model in OM, to the best of our knowledge and based on the work [12]. It was exploited the semantic similarity based on the word embedding model for improving matching quality. Also, the semantic similarity was combined with the edit distance-based similarity in a hybrid model. The proposed work was evaluated on the OAEI benchmark and conference tracks, and real-world ontologies. The results shown promises in matching quality improvement. Authors recommended the proposing of different strategies in applying the semantic similarity for more accuracy improvements of the matching. Also, there is a limitation in using the general-purpose dataset in building the word embedding models rather than the domain specific datasets.

The PoMap is an OM system that was proposed in [9]. It was carried out the matching process based on the syntactic matching that covered the element and structural levels of matching. Several string similarity measures and thresholds were evaluated, as preliminary step, to select the optimal values of them. The element and structural levels of matching were combined in sequential. This work had some drawbacks summarized as follows. First, the quality still needs more improvements. Secondly, the work depended on only the string and linguistic similarities and didn’t evaluate the semantic similarity effectiveness on the matching quality improvement. Also, to the best of the improvement, the target application should be specified to guide the matching process improvement.

The work [12] proposed an unsupervised entity alignment method called DeepAlignment. The DeepAlignment method focused on refining the pre-trained word vectors that describe ontological entities in unsupervised way. Then, these refined word vectors were exploited in the OM process. This work evaluated the importance of the representation learning techniques in OM. But it still needs more improvement in the quality of matching. Also, it didn’t evaluate the effect of matching large-scale ontologies on the proposed approach.

3. PROPOSED SEMANTIC MATCHER

The semantic matching process finds the correspondences between two different ontologies. The two ontologies are the source O1 and the target O2 respectively. O1 and O2 consist of a set of entities E1 and E2 respectively that need to be semantically matched. The entities E1 and E2 are classes or concepts, properties, and individual. All the matched entities types are denoted by words.

The proposed semantic matcher is illustrated in algorithm 1. It takes in the two entity sets E1 and E2 of O1 and O2 respectively, the word embedding model, and the threshold. Word2vec (W2V) is the word embedding implementation that built and fed into the semantic matcher. The threshold role is the refinement of the result alignments A of algorithm 1. It excludes the alignments which similarity value less than the specified threshold value. Algorithm 1 works as follow. Firstly, it accepts two entities each time to be matched from ontology O1 and O2 respectively (line 2 and 13). Secondly, the semantic matcher checks each input word to determine if it is single or consists of several words i.e., sentence (from line 4 to 12 and from line 14 to 22). Single words and sentences are treated differently in the semantic matcher.
Algorithm 1: Proposed semantic matcher

Input: O1 entities E1, O2 entities E2, word2vec model w2v, and threshold
Output: set of alignments A

1. Begin:
2. For each e1 in E1 do
3.     sim = 0
4.     If e1 is a single word then
5.         e1Vector = w2v(e1) // get e1 semantic vector representation
6.     Else
7.         e1array = split(e1) // split e1 into the array e1array of words
8.         For each e1c in e1array do
9.             e1cVector = w2v(e1c)
10.            e1Vector = sumVector(e1vector, e1cVector)
11.     End for
12.     End if
13.     For each e2 in E2 do
14.         If e2 is a single word then
15.             e2Vector = w2v(e2) // get e2 semantic vector representation
16.         Else
17.             e2array = split(e2) // split e2 into the array e2array of words
18.             For each e2c in e2array do
19.                 e2cVector = w2v(e2c)
20.                e2Vector = sumVector(e2vector, e2cVector)
21.         End for
22.         End if
23.     sim = cosineSimilarity(e1Vector, e2Vector)
24.     If sim >= threshold then
25.         add e1, e2, and sim to A
26.     End if
27.     End for
28. End for
29. Return A
30. End

For single words, the semantic vector representation is retrieved through the W2V model. For sentences or compound words, the proposed matcher tokenizes them into single words. Then the semantic vector representation for each tokenized word is retrieved through the W2V model and summed into one vector representation. Thirdly, the semantic vectors representations of each single and compound word, from the previous step, are passed to the cosine similarity for similarity calculation (line 23). Finally, the alignment refinement stage where the result alignment is added to the list of refined alignments A if its similarity is equal to or more than the specified threshold value (lines 24-25). The alignment is a triple consist of the two words and their similarity value <word1, word2, simVal>. The list of refined alignments A is the final alignments.

4. RESULTS AND DISCUSSION

The development environment is the Java programming. The experiment device specifications are: windows 10 platform, 6 GB RAM, core i3 processor, and 150 GB of the disk. Deeplearning4J (DL4J)\(^1\) is a java library that provides implementation of most machine and deep learning algorithms. It also provides basic natural language processing (NLP) tasks implementation. It is utilized for the W2V model implementation and the pre-processing of the matching ontologies. The pre-processing includes unifying the case of the matching words, removing symbols like hashes, underscores, etc. Based on the pre-specified target of improving the matching quality, skip-gram of the W2V word

\(^1\) https://deeplearning4j.org/
embedding model is the best fit based on the experiments and results. Its parameters values that best fit the paper needs are: window size = 5, vector dimensionality = 100, number of iterations during the learning = 3.

The experiment is evaluated on gold standard datasets and reference alignments from the OAEI\textsuperscript{2} contest. Two tracks for different domains are selected from the OAEI, the conference\textsuperscript{3} and the anatomy\textsuperscript{4}. The conference track is dedicated for the conference organizations domain and includes 16 ontologies. The anatomy track is dedicated for the biomedical domain and consists of two ontologies. The proposed semantic matcher is evaluated against the quality. Accuracy is measured the quality in terms of precision, recall, and F-measure measures. Precision and recall measure the correctness and the completeness respectively of the proposed approach. Equations 1, 2 and 3 represent the precision, the recall, and the F-measure respectively where “A” is the extracted refined alignments and R\textsubscript{f} is the reference alignments. |A ∩ R\textsubscript{f}| is the number of true matched refined alignments. The threshold refines the extracted alignments. It has indirect effect on the precision and recall by controlling the number of extracted alignments. The experiment is evaluated at the thresholds 1.0 and 0.9.

\[
\text{Precision } (A, R) = \frac{|A \cap R_f|}{|A|} \quad (1)
\]
\[
\text{Recall } (A, R) = \frac{|A \cap R_f|}{|R_f|} \quad (2)
\]
\[
\text{F-measure } = \frac{(2 \times \text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \quad (3)
\]

Table 1 shows the accuracy of the proposed semantic matcher on the conference dataset at the threshold 0.9. The proposed semantic matcher results are compared to the previous work [7], StringEquiv [12], CroMatch [14], and DeepAlignment [12]. The proposed approach outperforms the work [7] and the StringEquiv in precision, recall, and F-measure. It only has a good precision value than the CroMatch and DeepAlignment works. Also, it differs from the CroMatch and the DeepAlignment in the F-measure values with 0.044 and 0.074 respectively.

Table 2 shows the comparative quality analysis on the conference dataset. The proposed semantic matcher results are compared to the previous work [7], StringEquiv [12], CroMatch [14], and DeepAlignment [12]. The result alignments are refined at the threshold 1.0. Our approach outperforms the work [7] and the StringEquiv in precision, recall, and F-measure. It has only good precision than the CroMatch and DeepAlignment. The proposed approach differs from the CroMatch and DeepAlignment in the F-measure values with 0.069 and 0.099 respectively.

Based on the conference dataset evaluation at the tables 1 and 2, with the increase of the threshold value from 0.9 to 1.0, the precision, recall, and F-measure are affected. The precision is decreased from 0.909 to 0.896 with a difference 0.013 due to the reverse relationship between the threshold and the precision. The recall is decreased from 0.579 to 0.541 with a difference 0.038 due to the proportional relationship between the number of extracted alignment and the number of refined alignments.

\textsuperscript{2} http://oaei.ontologymatching.org/2013/
\textsuperscript{3} http://oaei.ontologymatching.org/2013/conference
\textsuperscript{4} http://oaei.ontologymatching.org/2013/anatomy
Table 1. Conference dataset accuracy evaluation at threshold 0.9

|                  | Precision | Recall | F-Measure |
|------------------|-----------|--------|-----------|
| Previous Work [7]| 0.872     | 0.469  | 0.61      |
| StringEquiv      | 0.83      | 0.5    | 0.62      |
| CroMatch         | 0.76      | 0.69   | 0.72      |
| DeepAlignment    | 0.71      | 0.8    | 0.75      |
| Proposed work    | **0.909** | **0.579** | **0.676** |

Table 2. Conference dataset accuracy evaluation at threshold 1.0

|                  | Precision | Recall | F-Measure |
|------------------|-----------|--------|-----------|
| Previous Work [7]| 0.872     | 0.469  | 0.61      |
| StringEquiv      | 0.83      | 0.5    | 0.62      |
| CroMatch         | 0.76      | 0.69   | 0.72      |
| DeepAlignment    | 0.71      | 0.8    | 0.75      |
| Proposed work    | **0.896** | **0.541** | **0.651** |

Figure 1. Proposed semantic matcher accuracy at threshold 0.9

Figure 2. Proposed semantic matcher accuracy at threshold 1.0
Figures 1 and 2 are shown the accuracy of the proposed semantic matcher on the anatomy dataset at threshold 0.9 and 1.0 respectively. The proposed semantic matcher results are compared to the StringEquiv and ALIN works [15]. At figure 1, it has the best recall and F-measure values than StringEquiv and ALIN. It differs from StringEquiv and ALIN in the precision with 0.35 at figure 2, the proposed semantic matcher outperforms the StringEquiv and ALIN in the recall and F-measure. It differs from StringEquiv and ALIN in the precision with 0.196 and 0.195 respectively. The increasing from 0.9 to 1.0 in the threshold value results in the decreasing of the precision values among our approach and StringEquiv and ALIN from 0.35 to 0.196 and 0.195 respectively.

Based on the anatomy dataset evaluation at figures 1 and 2, the threshold value increasing from 0.9 to 1.0 affects the results. The precision is increased from 0.646 to 0.801 due to the decrease of the number of extracted refined alignments. The recall is decreased from 0.956 to 0.753 due to the decrease in the number of extracted refined alignments that result in the decrease of the true matched refined alignments numbers. The F-measure value weights the precision and the recall values. It is increased from 0.771 to 0.776.

5. CONCLUSIONS

This paper is proposed semantic matcher approach for OM for BDI and hence schema integration application. The proposed semantic matcher exploits the semantic similarity based on the word embedding model. From the results and evaluation, it provides the following advantages. The semantic similarity improves the OM accuracy than the string similarity. Its dependency on semantic word representation provided by the word embedding model provides good accuracy results. The semantic similarity strategy that deals with single and compound words matching provides promising results in improving the accuracy. In the future, more domains will be evaluated. The enlarging of datasets was utilized in the building of the W2V word embedding model for increasing the accuracy improvement.

REFERENCES

[1] Euzenat J, Shvaiko P. Ontology Matching. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013.
[2] Saruladha K, Aghila G, Sathiya B. LOMPT: An efficient and Scalable Ontology Matching Algorithm. Procedia Engineering. 2012; 38:2272-2287.
[3] Salah Kettouch M, Luca C, Hobbs M, Dascalu S. Using semantic similarity for schema matching of semi-structured and linked data. 017 Internet Technologies and Applications (ITA), Wrexham. 2017; 128-133.
[4] Pooja Mansukhalal G, Malathy C. Semantic Integration with Ontology Based Approach. 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICCEEOT). 2016; 2257-2260.
[5] Eine B, Matthias J, and Werner Q. “Ontology-Based Big Data Management.” Systems Approaches and Tools for Managing Complexity 5, no. 3 (2017).
[6] Mountasser I, Brahim O, and Bouchra F. “Hybrid Large-Scale Ontology Matching Strategy on Big Data Environment.” 18th International Conference on Information Integration and Web-Based Applications and Services, 2016.
[7] Zhang Y, Xuepeng W, Siwei L, Shizhu H, Kang L, and Jun Z. “Ontology Matching with Word Embeddings.” Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. NLP-NABD 2014, CCL 2014. Lecture Notes in Computer Science 8801 (2014): pp.34–45.
[8] Ochieng P, and SWAIB K. “Large Scale Ontology Matching: State of the Art Analysis.” ACM Computing Surveys; 51, May (2018): pp 1–35.
[9] Laadhar, A, F Ghozzi, I Megdiche, F Ravat, O Teste, and F Gargouri. “POMap: An Effective Pairwise Ontology Matching System.” 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD 2017), 2017, 161–168.
[10] Mikolov T, Kai C, Greg C, and Jeffrey D. “Distributed Representations of Words and Phrases and Their Compositionality.” Advances in Neural Information Processing Systems 26 (NIPS 2013), 2013, pp 3111–3119.
[11] Mikolov T, Greg C, Kai C, and Jeffrey D. “Efficient Estimation of Word Representations in Vector Space.” *ArXiv Preprint ArXiv:1301.3781*, 2013, pp 1–12.

[12] Kolyvakis, Prodromos, Alexandros Kalousis, and Dimitris Kiritsis. “DeepAlignment: Unsupervised Ontology Matching With Refined Word Vectors.” *Proceedings of NAACL-HLT*, 2018, 787–798.

[13] Xiang C, Tingsong J, Baobao C, and Zhifang S. “ERSOM: A Structural Ontology Matching Approach Using Automatically Learned Entity Representation.” *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal*, no. September (2015): 2419–2429.

[14] Vrdoljak B, Marko B, B Vrdoljak, and M Banek. “CroMatcher: An Ontology Matching System Based on Automated Weighted Aggregation and Iterative Final Alignment.” *Journal of Web Semantics*, 2016.

[15] Achichi M, Michelle C, Zlatan D, Daniel F, Alfio F, Giorgos F, Irini F, et al. “Results of the Ontology Alignment Evaluation Initiative 2017.” *Proceedings of 16th International Semantic Web Conference (ISWC 2017)*, 2017.