

Property-based Entity Type Graph Matching

Fausto Giunchiglia^[0000-0002-5903-6150] and Daqian Shi^[0000-0003-2183-1957]

Department of Information Engineering and Computer Science (DISI),
University of Trento, Italy
{fausto.giunchiglia, daqian.shi}@unitn.it

Abstract. We are interested in dealing with the heterogeneity of Knowledge bases (KBs), e.g., ontologies and schemas, modeled as sets of entity types (etypes), e.g., person, where each etype is associated with a set of properties, e.g., age or height, via an inheritance hierarchy. A huge literature exists on this topic. A common approach is to model KBs as graphs decorated with labels and reduce the problem of KB matching to that of matching these two elements, *viz.*, labels and structure of the graph. However, labels of etypes are often misplaced, e.g., they are more general or specific than the correct etype, as defined by its properties. Structure-based matching may also lead to wrong conclusions as the properties assigned to an etype in an inheritance hierarchy do not depend on the order by which they are assigned and, therefore, on the specific structure of the graph. In this paper, we propose a novel etype graph matching approach, dealing with the two problems highlighted above, based on two key ideas. The first is to implement matching as a classification task where etypes are characterized by the associated properties. The second is we propose two *property-based* etype similarity metrics, which model the roles that properties have in the definition of an etype. The experimental results show the effectiveness of the algorithm, in particular for those etype graphs with a high number of properties.

Keywords: Etype graph matching · Machine learning · Entity type similarity · Knowledge reuse

1 Introduction

We are interested in dealing with the heterogeneity of Knowledge bases (KBs), e.g., ontologies and schemas, modeled as sets of entity types (etypes), e.g., person, where each etype is associated with a set of properties, e.g., age or height, via an inheritance hierarchy. A huge literature exists on this topic, e.g., [23, 24, 33]. Most etype graph matching approaches exploit label-based methods [6, 36], such as character similarity metrics and synonym analysis, and structure-based methods [18], implementing various forms of graph matching. However, labels of etypes may suggest a wrong etype [19, 34]. For example, an eagle can be labelled as *Bird* in a general-purpose ontology and *Eagle* in a domain-specific ontology. Structure-based matching may also lead to wrong conclusions as the properties assigned to an etype in an inheritance hierarchy are cumulative and depend only

on the nodes in the path from the root and, therefore, do not depend on the order by which they are assigned. For example, the super-class of etype *Eagle* can be *Animal* in one etype graph and *Bird* in another etype graph.

As a solution to the above problems, the main intuition underlying the work described in this paper is to match etypes on the basis of the properties which are used to define them. It is, in fact, the properties that are used to intensionally define an etype which define it independently of its specific name and also independently of its hierarchy [17]. Furthermore, it is fact that in most relevant ontologies, etypes are associated with sufficient properties, like DBpedia [1] and OpenCyc [13]. And the reason for this is quite obvious, being the purpose of any data or knowledge integration task exactly that of extending the number of properties associated to an etype.

In this paper, we implement the above intuition based on main contributions:

- We introduce two *property-based* etype similarity metrics, namely the *horizontal similarity* ES_h and the *vertical similarity* ES_v which characterise the role that properties have in the definition of given etypes. These similarity metrics capture the main idea that for any two etypes, the properties which distinguish one etype from the other should not occur in the other etype. Since different properties contribute differently for matching etypes, we introduce ES_h which focuses on measuring the properties with different shareability, and ES_v measures properties based on their specificity.
- We implement the etype graph matching as a classification task where the matching of etypes is based on their associated properties. In this paper, we propose and evaluate a machine learning (ML)-based etype graph matching approach.

The paper is organized as follows. Section 2 introduces our own specific formalization for etype graphs and relevant terminology. Section 3 presents two property-based etype similarity metrics. Section 4 introduces the overall etype graph matching algorithm. The evaluation details and results in Section 5, where the experiments are based on a selected test cases from the Ontology Alignment Evaluation Initiative (OAEI) [11]. Finally, we present the related work in Section 6 and the conclusions in Section 7.

2 Etype Graphs as FCA contexts

We formalize etype graphs as formal concept analysis [17] (FCA) contexts. Specifically, we define an etype graph ETG as $ETG = \langle E, P, T \rangle$, with $E = \{e_1 \dots e_n\}$ being the set of etypes from the etype graph, $P = \{p_1 \dots p_n\}$ being the set of properties, $T = \{e \in E | \langle e, T(e) \rangle\}$ being the set of correspondences between etypes and properties, where function $T(e)$ returns properties of e . We consider the property p is used to describe an etype e when the property belongs to set $T(e)$. Two observations:

1. E is a set of etypes but not a set of entities. Similar to what happens in general FCA, which assumes that an entity is described by a set of property

values, an etype is considered to be described by a set of properties $T(e)$. Since in our method we focus on the correlations between etypes and properties, we organize an etype graph as etype-property correlation map as an FCA context without containing additional information.

2. Etype characterization exploits not only the properties associated with it but also the others, namely those which are not used in its definition. Thus, we introduce the non-associated properties into our FCA context and distinguish two more different cases for better presenting the FCA context.

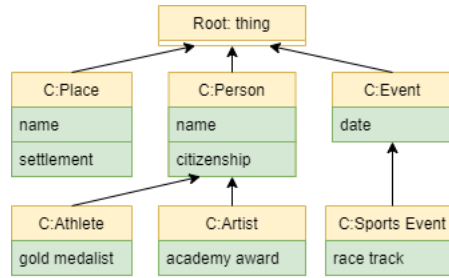


Fig. 1. An example the hierarchy of etype graph

As an example, Figure 1 presents the hierarchy of an etype graph, extracted from DBpedia [1]. In each box, etypes are presented in yellow and their properties in green. We formalize the etype graph in Figure 1, into an FCA context as from below.

| FCA Context | | Properties | | | | | | |
|-------------|--------------|------------|------|-------------|------------|---------------|---------------|------------|
| | | name | date | citizenship | settlement | academy award | gold medalist | race track |
| Etypes | Person | O | X | O | X | UN | UN | X |
| | Event | X | O | X | X | X | X | UN |
| | Place | O | X | X | O | X | X | X |
| | Artist | O | X | O | X | O | X | X |
| | Athlete | O | X | O | X | X | O | X |
| | Sports Event | X | O | X | X | X | X | O |

Fig. 2. An example of formalizing etype graph into FCA contexts

In Figure 2 we adopt the following conventions. The value box with a circle represents the fact the property is associated with the etype, e.g., *citizenship* is associated with *Person*. The value box with a cross means the property is not associated with the etype, e.g., *date* is not used to describe etype *Person*. The value “UN” represents the fact that the property is not associated with the etype but associated with at least one of its subclasses, namely undefined. The intuition is that the property might or might not be used to describe the current etype, e.g., *academy award* is used to describe *Artist* and it might be

used to describe *Person* since *Artist* is a subclass of *Person*. We encode these three correlations as the parameter w_p . Since the correlation of “associated with” is positive for a property describing an etype, the correlation of “not associated with” is negative and the correlation of “undefined” is neutral, we take w_p to be defined as $w_p \in \{1, 0, -1\}$.

$$w_p = \begin{cases} 1, & \text{if } p \in \text{prop}(E) \\ 0, & \text{if } p \notin \text{prop}(E) \& p \in \text{prop}(E.\text{subclass}) \\ -1, & \text{if } p \notin \text{prop}(E) \& p \notin \text{prop}(E.\text{subclass}) \end{cases} \quad (1)$$

In the above equation, we take p as the target property and $\text{prop}(E)$ as the properties associated with E . Thus, the circles, UNs and crosses in Figure 2 are set to 1, 0 and -1, respectively.

3 Property-based similarity

The similarity metrics are inspired to the work in [16, 19] in considering properties as one of the most important features to describe an etype and to the formalization of the “get-specific” heuristic provided in [20]. These provide us the intuition that a more specific property provides more information to identify an etype. Let us introduce our two etype similarity metrics in detail.

3.1 Horizontal Similarity

When measuring the specificity of a property, a possible idea is to horizontally compare the number of etypes that are described by a specific property, namely the shareability of the property [19]. If a property is used for describing diverse etypes, it means that the property is not highly characterizing. Thus, for instance, in figure 2, the property *name* is used to describe *Person*, *Place*, *Athlete*. Dually, if a property is used for describing a few etypes or the property is associated with only one etype, this means this property can be regarded as highly characterizing, e.g., in Figure 2, property *settlement* is specific for etype *Place*. Based on this intuition, we consider the specificity of a property is related to its shareability. Therefore, we propose SP as the metric for measuring property specificity. More precisely, SP aims to minimize the number of etypes that are associated with the target property in a specific etype graph. We model the metric SP as:

$$SP_{ETG}(p) = w_p * e^{\lambda(1-n(p))} \in [-1, 1] \quad (2)$$

where p is the input property and $n(p)$ is the number of etypes that are described by the input property in a specific entity graph ETG , thus $n(p) \geq 0$; e refers to the natural mathematical constant [15]; λ is a constraint factor whose aim is to produce a gentle curve. Assume that A and B are two etype graphs. Then we model ES_h as follows:

$$ES_h(E_a, E_b) = \frac{1}{2} \sum_{i=1}^k \left(\frac{SP_A(p_i)}{|\text{prop}(E_a)|} + \frac{SP_B(p_i)}{|\text{prop}(E_b)|} \right) \in [0, 1] \quad (3)$$

where we take E_a, E_b as the candidate etypes from A and B respectively. Thus $E_a \in A$ and $E_b \in B$; $prop(E)$ refers to the properties associated with the specific etype and $|prop(E)|$ refers to the number of $prop(E)$. k is the number of matched properties which are associated with both etype E_a and E_b . $SP_A(p_i)$ and $SP_B(p_i)$ refer to the specificity of the aligned property p_i in A and B , respectively. Notice that we have $ES_h(E_a, E_b) = ES_h(E_b, E_a)$. Notice also that we apply z-score normalization [29] to ES_h at the end of calculation, and that the range of ES_h is between 0 to 1.

3.2 Vertical Similarity

Etype graphs are organized as classification hierarchies such that upper-layer etypes represent more abstract or more general concepts, whereas lower-layer etypes represent more concrete or more specific concepts [20, 31]. Correspondingly, properties of upper-layer etypes are more general since they are used to describe general concepts, vice versa, properties of lower-layer etypes are more specific since they are used to describe specific concepts. We assume that specific properties will contribute more to the identification of an etype. For instance, in Figure 2, as a lower-layer etype, *Artist* can be identified by the property *academy award* but not by the property *name*. Based on this intuition, we propose $L(p)$ as a metric for measuring property specificity. We model $L(p)$ as follows:

$$LETG(p) = w_p * \theta * \min_{E \in etype(p)} layer(E) \in [-1, 1] \quad (4)$$

where: θ is a constraint factor which normalized the range of the function; $etype(p)$ outputs all the etypes that are described by the property p ; and $layer(E)$ refers to the layer of the inheritance hierarchy where an etype E is defined. We define the vertical etype similarity metric ES_v as from below.

$$ES_v(E_a, E_b) = \frac{1}{2} \sum_{i=1}^k \left(\frac{L_A(p_i)}{|prop(E_a)|} + \frac{L_B(p_i)}{|prop(E_b)|} \right) \in [0, 1] \quad (5)$$

Similar to the definition of ES_h , we have candidate etypes $E_a \in A$ and $E_b \in B$ and the properties $prop(E)$ associated with the etype E . The key difference is that ES_v exploits the property specificity based on the layer information $L(p)$. $L_A(p_i)$ and $L_B(p_i)$ refer to the highest layer of the aligned property p_i in A and B , respectively. Notice that ES_v is symmetric as well. ES_v is also normalized by z-score normalization, in the same way as ES_h . Finally the range of ES_v is between 0 to 1.

4 Etype Graph Matching

Figure 3 presents the Processing chart of our etype graph matching approach. It mainly consists of two matchers, the property matcher and the etype matcher. After parsing the input etype graph pair, properties are first sent into the NLP-based property matcher, where string-based and language-based similarity metrics are exploited to match two property labels [3, 4]. Then we generate the FCA

contexts according to the etypes and correlated property pairs. In this phase, we will also generate our property-based etype similarity metrics ES_h and ES_v and then send them all to the etype matcher. We develop a ML-based matcher which considers etype matching as a binary classification task. Thus, our etype matcher will predict two incoming etypes as match or unmatched and output the matched etypes as the final results.

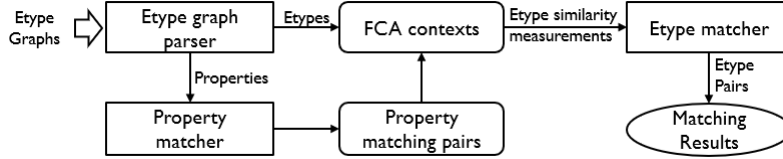


Fig. 3. Processing chart of our etype graph matching approach

Algorithm 1 below presents the step-by-step process for calculating property-based etype similarity metrics ES_h and ES_v . After formalizing etype graphs into FCA contexts, we assume that the two candidate FCA contexts f_a and f_b are generated. PM refers to the property pairs which are aligned by the property matcher, EM refers to the candidate etype pairs which are waiting for matching. For every etype pair in EM , we check their correlated properties and update the specificity values to ES_h or ES_v if the property pair is aligned. After traversing all the candidate etype pairs, we obtain completed etype similarities which will be used for training the ML model, or predicting if two etypes are matching. Table 1 provides some representative examples to show the etype similarity ES_v and ES_h between etypes from *cmt-confop* and *cmt-conference* in conference track.

Table 1. Examples of values of etype similarity ES_v and ES_h

| etype-cmt | etype-confop | ES_v | ES_h |
|-------------|---------------------|--------|--------|
| Paper | Contribution | 1 | 0.853 |
| Author | Author | 0.756 | 0.740 |
| SubjectArea | Topic | 0.198 | 0.961 |
| Meta-Review | Poster | 0 | 0.312 |
| etype-cmt | etype-conference | ES_v | ES_h |
| Chairman | Chair | 1 | 0.559 |
| Person | Person | 1 | 0.970 |
| Person | Conference_document | 0.02 | 0.06 |
| Chairman | Publisher | 0 | 0.07 |

Algorithm 1 Etype similarity generation. $ES_h, ES_v = \text{etypesim}(f_a, f_b)$

Input:Candidate FCA contexts f_a and f_b ;**Output:**Property-based etype similarity ES_h, ES_v ;

- 1: $PM = (p_a, p_b) = \text{PropertyMatcher}(f_a, f_b)$; {align p_a and p_b as property pairs by property matcher, where $p_a \in f_a$ and $p_b \in f_b$.}
 - 2: $EM = (E_a, E_b) = \text{EtypeSelector}(f_a, f_b)$; {select etypes E_a, E_b from f_a, f_b and assemble them as candidate etype pairs EM .}
 - 3: **for** all $EM_i \in EM$ **do**
 - 4: **for** all $p_a \in f_a, p_b \in f_b$ **do**
 - 5: **if** $(p_a, p_b) \in PM$ **then**
 - 6: $ES_h(EM_i).add(SP(p_a), SP(p_b))$; {add the horizontal specificity to etype similarity $ES_h(EM_i)$, refers to equation 3}
 - 7: $ES_v(EM_i).add(L(p_a), L(p_b))$; {add the vertical specificity to etype similarity $ES_v(EM_i)$, refers to equation 5}
 - 8: **end if**
 - 9: **end for**
 - 10: **end for**
 - 11: **return** ES_h, ES_v
-

5 Evaluation

We first describe the evaluation set-up and then provide the results from the experiments.

5.1 Evaluation Set-up

The main decision for the evaluation was to take OAEI as the main reference for the selection of the matching problems. As of today, this in fact the major source of ontology matching problems.

Our approach focuses on ontologies that contain etypes associated with a fair number of properties. As a result, we have selected the following cases: the bibliographic ontology dataset [10] and conference track [38] (*ra1* version). From the bibliographic ontology dataset, we select series #101 and series #301-304, which present real-life ontologies for bibliographic references from the web. We set these bibliographic ontologies as the training set for training our ML-based etype matcher. The conference track contains 16 ontologies, dealing with conference organizations, and 21 reference alignments. We set the 21 reference alignments from the conference track as the testing set to validate our etype matcher. We select the training and testing set from different cases since we aim to prove the adaptation of our approach, which also prevents our approach from overfitting. Notice that there is an unbalanced positive and negative sample issue when we match two candidate ontologies, which means negative samples will be produced much more than positive samples. To address this issue, we propose a model training strategy that decreases the negative samples and duplicates

a part of positive samples to achieve a balanced training set and to alleviate overfitting.

In this paper, our matching approach applies a general binary classification strategy, which is independent of the specific ML model. Thus, the data label is 1 or 0, which means two etypes are matching or unmatching respectively. The data consists of three kinds of attributes, which are string-based similarity metrics (N-garm [12], Longest common subsequence [12], Levenshtein distance [37]), language-based similarity metrics (Wu and Palmer similarity [28], Word2vec [8]) and property-based similarity metrics (ES_h and ES_v). These etype similarities aim to measure different aspects of the relatedness between two etypes. Here we select some of the most common string-based and language-based similarity metrics as additional metrics working with our property-based similarity metrics for achieving better etype matching results.

5.2 Experimental Results

For better evaluating the validity of our approach, we apply 4 different ML models, which are: random forest [27], stochastic gradient descent (SGD) classifier [22], decision tree [32] and logistic regression [26]. We have compared our work with state-of-the-art matching methods, as they came out of previous OAEI evaluation campaigns. The involved state of the art systems are: FCAMap [7], AML [14], LogMap and LogMapLt [21]. We calculate precision, recall, F_1 -measure, and also $F_{0.5}$ -measure and F_2 -measure [30].

Table 2. *Comparing our method with SOTA methods*

| ConferenceTrack | Prec. | Rec. | $F_{0.5}$ -m. | F_1 -m. | F_2 -m. |
|-------------------------|--------------|--------------|---------------|--------------|--------------|
| FCAMap | 0.680 | 0.625 | 0.668 | 0.651 | 0.635 |
| AML | 0.832 | 0.630 | 0.782 | 0.717 | 0.662 |
| LogMap | 0.798 | 0.592 | 0.746 | 0.680 | 0.624 |
| LogMapLt | 0.716 | 0.554 | 0.676 | 0.625 | 0.580 |
| Ours-RandomForest | 0.529 | 0.884 | 0.575 | 0.662 | 0.779 |
| Ours-SGDClassifier | 0.779 | 0.632 | 0.744 | 0.698 | 0.656 |
| Ours-DecisionTree | 0.671 | 0.703 | 0.677 | 0.687 | 0.696 |
| Ours-LogisticRegression | 0.556 | 0.808 | 0.593 | 0.659 | 0.741 |

Table 2 shows the results of our approach with the different models mentioned above, compared with the results of state-of-the-art methods. Firstly, we can find our approach with different models produce slightly different results, the SGD classifier performs the best in general, which leads the precision, $F_{0.5}$ -measure and F_1 -measure. And, random forest advances in recall and F_2 -measure. Decision tree and logistic regression classifiers are marginally powerless than the other two in conference track ontologies. Secondly, extend to the overall comparison, we can find that AML has the best overall results. Leading the precision, $F_{0.5}$ -measure

and F1-measure. Our approach with random forest leads the results on recall and F_2 -measure. Considering that the average results of our approach with different models are performing close to the state-of-the-art on F_1 -measure, we can say that our approach leads to similar results as state-of-the-art competitors, while advances in different aspects¹.

The comparison to state-of-the-art methods shows the validity of our etype matcher. Moreover, we design a second experiment which is an ablation test to evaluate if our designed property-based etype similarity metrics are effective. In this experiment, we test on the backbone model (B) which was trained only by string-based and language-based similarity metrics. We also test on the model with ES_h , ES_v , and both ES_h and ES_v (ours), respectively. Note that the backbone model refers to *Ours-SGDClassifier* in table 2.

Table 3 shows the results of the ablation test, it is easy to find by using our designed metrics, the results significantly improved comparing with the results of the backbone model. Moreover, although B+ ES_h achieve the best recall measure, B+ ES_v + ES_h leads in precision and F_1 -measure, which means the best overall performance. This observation shows both our designed metrics are effective on the etype matching task. At the same time, the etype matcher achieves the best performance by simultaneously using ES_v and ES_h .

Table 3. Ablation test

| Models | Prec. | Rec. | F_1 -m. |
|--------------------|--------------|--------------|--------------|
| B | 0.621 | 0.605 | 0.613 |
| B+ ES_v | 0.650 | 0.700 | 0.674 |
| B+ ES_h | 0.634 | 0.729 | 0.678 |
| B+ ES_v + ES_h | 0.779 | 0.632 | 0.698 |

6 Related work

Based on the idea originally introduced in [19] and different from all the previous work, our approach is based on the idea of exploiting properties as the main means for matching etypes. We provide below a short summary of the four main techniques that we exploit in the implementation of property-based etype similarity, namely, label matching, graph matching, and the use of ML and FCA.

In the early stages of ontology matching, etype matching methods mostly focused on string-based methods. The work in [36] reviews a wide range of string similarity metrics and propose an ontology alignment method by selecting different powerful similarity metrics. Later, ensemble metrics strategies were introduced in some studies [25], which apply multiple matchers based on different

¹ All approaches do not have significant differences in running times since the conference track contains no large ontology.

string-based metrics. The principle of these works is that the combined matchers are more powerful than an individual matcher.

The structure of an etype graph has also been considered as important information for identifying etypes, like [2,18]. The LogMap system [21] uses a two-step matching strategy, that is, matches two etypes E_a and E_b by a lexical matcher, and then considers the etypes that are semantically close to E_a are more likely to be semantically close to E_b . AML [14] introduces an ontology matching system that consists of a string-based matcher and a structure-based matcher, building internal correspondences by exploiting *is-a* and *part-of* relationships.

Some work on matching etypes is based on the use of ML. This work models the etype matching task as a binary classification task, trying to encode the information like string similarities and structure information as attributes. For instance, the work in [5] achieves promising results by encoding the lexical similarity of the superclass and subclass as structural similarity.

Finally, FCA lattices have been applied in etype matching methods in the work described in [7,35]. To refine health records searching outputs, the work in [9] introduced a matching method based on FCA which assists the end-user in defining their queries. In turn, in [35] a bottom-up ontology merging approach was proposed where FCA lattices were used to keep track of the ontology hierarchy.

7 Conclusions

In this paper, we have introduced a novel etype graph matching approach via property-based similarity measurement. Firstly, we discussed a novel formalization method for etype graphs, which encodes etypes and properties into FCA contexts. Then we proposed two novel metrics for measuring the contextual similarity between two etypes, namely horizontal similarity and vertical similarity. Based on our proposed metrics, we have developed a ML-based framework for etype graph matching. The experimental results show the validity of our approach.

Acknowledgements

The research conducted by Fausto Giunchiglia has received funding from the *InteropEHRRate* project, co-funded by the European Union (EU) Horizon 2020 programme under grant number 826106, and the research conducted by Daqian Shi has received funding from the program of China Scholarships Council (No. 202007820024).

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)

2. Autayeu, A., Giunchiglia, F., Andrews, P.: Lightweight parsing of classifications into lightweight ontologies. In: European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2010). Glasgow, United Kingdom (September 2010)
3. Bella, G., Giunchiglia, F., McNeill, F.: Language and domain aware lightweight ontology matching. *Journal of Web Semantics* **43**, 1–17 (2017)
4. Bella, G., Zamboni, A., Giunchiglia, F.: Domain-based sense disambiguation in multilingual structured data. In: DIVERSITY Workshop at ECAI 2016 (2016)
5. Bulygin, L., Stupnikov, S.A.: Applying of machine learning techniques to combine string-based, language-based and structure-based similarity measures for ontology matching. In: DAMDID/RCDL. pp. 129–147 (2019)
6. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: International semantic web conference. pp. 294–309. Springer (2013)
7. Chen, G., Zhang, S.: Identifying mappings among knowledge graphs by formal concept analysis. In: OM@ ISWC. pp. 25–35 (2019)
8. Church, K.W.: Word2vec. *Natural Language Engineering* **23**(1), 155–162 (2017)
9. Curé, O.C., Maurer, H., Shah, N.H., Le Pendu, P.: A formal concept analysis and semantic query expansion cooperation to refine health outcomes of interest. *BMC medical informatics and decision making* **15**(1), 1–6 (2015)
10. Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., et al.: Results of the ontology alignment evaluation initiative 2009 (2010)
11. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology alignment evaluation initiative: six years of experience. In: *Journal on data semantics XV*, pp. 158–192. Springer (2011)
12. Euzenat, J., Shvaiko, P., et al.: *Ontology matching*, vol. 18. Springer (2007)
13. Färber, M., Ell, B., Menne, C., Rettinger, A.: A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web Journal* **1**(1), 1–5 (2015)
14. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In: *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*. pp. 527–541. Springer (2013)
15. Finch, S.R.: *Mathematical constants*. Cambridge university press (2003)
16. Fumagalli, M., Daqian, S., Giunchiglia, F.: Ranking schemas by focus: a cognitively-inspired approach. In: *26th International Conference on Conceptual Structures (2021)*
17. Ganter, B., Wille, R.: *Formal concept analysis: mathematical foundations*. Springer Science & Business Media (2012)
18. Giunchiglia, F., Autayeu, A., Pane, J.: S-match: an open source framework for matching lightweight ontologies. *Semantic Web* **3**(3), 307–317 (2012)
19. Giunchiglia, F., Fumagalli, M.: Entity type recognition—dealing with the diversity of knowledge. In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. vol. 17, pp. 414–423 (2020)
20. Giunchiglia, F., Zaihrayeu, I., Kharkevich, U.: Formalizing the get-specific document classification algorithm. In: *International Conference on Theory and Practice of Digital Libraries*. pp. 26–37. Springer (2007)
21. Jiménez-Ruiz, E., Grau, B.C.: Logmap: Logic-based and scalable ontology matching. In: *International Semantic Web Conference*. pp. 273–288. Springer (2011)
22. Kabir, F., Siddique, S., Kotwal, M.R.A., Huda, M.N.: Bangla text document categorization using stochastic gradient descent (sgd) classifier. In: *2015 International*

- Conference on Cognitive Computing and Information Processing (CCIP). pp. 1–4. IEEE (2015)
23. Lenzerini, M.: Data integration: A theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 233–246 (2002)
 24. Lonsdale, D., Embley, D.W., Ding, Y., Xu, L., Hepp, M.: Reusing ontologies and language components for ontology generation. *Data & Knowledge Engineering* **69**(4), 318–330 (2010)
 25. Nezhadi, A.H., Shadgar, B., Osareh, A.: Ontology alignment using machine learning techniques. *International Journal of Computer Science & Information Technology* **3**(2), 139 (2011)
 26. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: *Advances in neural information processing systems*. pp. 841–848 (2002)
 27. Pal, M.: Random forest classifier for remote sensing classification. *International journal of remote sensing* **26**(1), 217–222 (2005)
 28. Palmer, M., Wu, Z.: Verb semantics and lexical zhibiao w u. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico. pp. 133–138 (1994)
 29. Patro, S., Sahu, K.K.: Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462 (2015)
 30. Pour, N., Algergawy, A., Amini, R., Faria, D., Fundulaki, I., Harrow, I., Hertling, S., Jiménez-Ruiz, E., Jonquet, C., Karam, N., et al.: Results of the ontology alignment evaluation initiative 2020. In: *Proceedings of the 15th International Workshop on Ontology Matching (OM 2020)*. vol. 2788, pp. 92–138. CEUR-WS (2020)
 31. Rios-Alvarado, A.B., Lopez-Arevalo, I., Sosa-Sosa, V.J.: Learning concept hierarchies from textual resources for ontologies construction. *Expert Systems with Applications* **40**(15), 5907–5915 (2013)
 32. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics* **21**(3), 660–674 (1991)
 33. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering* **25**(1), 158–176 (2011)
 34. Sleeman, J., Finin, T., Joshi, A.: Entity type recognition for heterogeneous semantic graphs. *AI Magazine* **36**(1), 75–86 (2015)
 35. Stumme, G., Maedche, A.: Fca-merge: Bottom-up merging of ontologies. In: *IJCAI*. vol. 1, pp. 225–230 (2001)
 36. Sun, Y., Ma, L., Wang, S.: A comparative evaluation of string similarity metrics for ontology alignment. *Journal of Information & Computational Science* **12**(3), 957–964 (2015)
 37. Yujian, L., Bo, L.: A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence* **29**(6), 1091–1095 (2007)
 38. Zamazal, O., Svátek, V.: The ten-year ontofarm and its fertilization within the onto-sphere. *Journal of Web Semantics* **43**, 46–53 (2017)