



Background Knowledge in Schema Matching: Strategy vs. Data

Jan Portisch^{1,2}✉, Michael Hladik³, and Heiko Paulheim¹✉

¹ Data and Web Science Group, University of Mannheim, Mannheim, Germany
`{jan,heiko}@informatik.uni-mannheim.de`

² SAP SE Business Technology Platform — One Domain Model, Walldorf, Germany
`jan.portisch@sap.com`

³ SAP SE Business Process Intelligence, Walldorf, Germany
`michael.hladik@sap.com`

Abstract. The use of external background knowledge can be beneficial for the task of matching schemas or ontologies automatically. In this paper, we exploit six general-purpose knowledge graphs as sources of background knowledge for the matching task. The background sources are evaluated by applying three different exploitation strategies. We find that explicit strategies still outperform latent ones and that the choice of the strategy has a greater impact on the final alignment than the actual background dataset on which the strategy is applied. While we could not identify a universally superior resource, BabelNet achieved consistently good results. Our best matcher configuration with BabelNet performs very competitively when compared to other matching systems even though no dataset-specific optimizations were made.

Keywords: Schema matching · Ontology matching · Background knowledge · Knowledge graphs · Knowledge graph embeddings · Data integration

1 Introduction

Ontology matching or *schema matching* is the non-trivial task of finding correspondences between entities of two or more given ontologies or schemas. The matching can be performed manually or through the use of an automated matching system. In both cases, the context is very important and concept knowledge is required. Therefore, automated matching systems require background knowledge to excel at the schema matching task. In most cases, *WordNet* is used as a form of general concept knowledge with a plain synonym lookup strategy. However, over the last decade many other sources of background knowledge that are much larger and also contain instance data have emerged. In addition, strategies to exploit structured knowledge, such as knowledge graph embedding models, have been developed but are rarely used in ontology matching. Exploiting background knowledge for ontology matching is still one of multiple challenges that is yet to be solved [39].

In this paper, we compare the performance of six different background datasets of varying size and characteristics for the task of schema matching. For each dataset, three different strategies are exploited. Besides an in-depth evaluation of the matching performance, we strive to test the following hypotheses:

H1 The strategy is more important than the resource.

H2 The resource is more important than the strategy.

H3 There is a superior resource.

H4 There is a superior strategy.

The remainder of this paper is structured as follows: In the next section, we present an overview on related work. Section 3 describes the general evaluation architecture that is used, as well as the generic matching process that was implemented for this paper. The background datasets and the strategies that are explored are presented in Sects. 4 and 5, respectively. The strategies on the background knowledge datasets are evaluated on four different gold standards in Sect. 6. The paper closes with a summary and an outlook on future work.

2 Related Work

Ontology and schema matching systems are evaluated by the *Ontology Alignment Evaluation Initiative (OAEI)*¹ every year since 2005. While, to our knowledge, there is no large comparison of different general knowledge background sources or exploitation strategies, many individual matching systems exist that make use of external background knowledge. In 2013, Euzenat and Shvaiko [5] counted more than 80 schema matching systems that exploit *WordNet*. Besides *WordNet*, few other general background data sources are used: The *WikiMatch* [12] system exploits the *Wikipedia* search API by determining concept similarity through the overlap of returned Wikipedia articles for a search term. *WeSeE Match* [29] queries search APIs and determines similarity based on TF-IDF scores on the returned Web site titles and excerpts. A synonymy and translation lookup strategy based on *Wiktionary* is used in [34] for monolingual and multilingual matching. Lin and Krizhanovsky [22] exploit *Wiktionary* for translation lookups within a larger matching system.

In the biomedical and life science domain, specialized external background knowledge is broadly available and heavily exploited for ontology matching. Chen et al. [4] extend the LogMap matching system to use *BioPortal*, a portal containing multiple ontologies, alignments, and synonyms, by (i) applying an overlap based approach as well as by (ii) selecting a suitable ontology automatically and using it as mediating ontology. As mappings between biomedical ontologies are available, those are used as well: Groß et al. [9] exploit existing mappings to third ontologies, so called *intermediate ontologies*, to derive mappings. This approach is extended by Annane et al. [1] who use BioPortal by exploiting existing alignments between the ontologies found there for matching through a path-based approach: By linking source and target concepts into the global mapping graph,

¹ <http://oaei.ontologymatching.org/>.

the paths that connect the concepts in that graph are used to derive new mappings. In the same domain, research has also been conducted on background knowledge selection. Faria et al. [7] propose the usage of a metric, called *Mapping Gain (MG)*, which is based on the number of additional correspondences found given a baseline alignment. Quix et al. [35] use a keyword-based vector similarity approach to identify suitable background knowledge sources. Similarly, Hartung et al. [10] introduce a metric, called *effectiveness*, that is based on the mapping overlap between the ontologies to be matched. While in the biomedical domain, many specialized resources are available and data schemas are heavily interlinked, this is not the case for other domains. As a consequence, such methods cannot be easily translated and applied.

Background knowledge sources are also used for multilingual matching tasks. Here, translation APIs are often used such as *Microsoft Bing Translator* by *KEPLER* [17] or *Google Translator* by *LogMap*.

Approaches that exploit vector representations of concepts are rarely found in the ontology or schema matching domain. The *DOME* [14] matching system employs a *doc2vec* [20] approach to concepts within the ontologies to be mapped. Similarly, *AnyGraphMatcher* [23] attempts to embed the ontologies to be mapped at runtime but achieves very low results in the OAEI 2019. *DESKMatcher* [27] applies a knowledge graph embedding approach on external knowledge but did not perform competitively in the OAEI 2020 either. *WebIsA-lod* is exploited as external background knowledge in [31] through a combined string matching and graph embedding strategy.

These examples show that there is a larger body of works exploiting background knowledge with various strategies; however, they are always used in the context of a larger matching system. Ablation studies and therefore statements about the utility of a particular source and/or strategy are not available.

3 General Approach

To close this gap, we propose a simple, generic matching process that can work with different sources of background knowledge and exploitation strategies. Our aim is *not* to build a top-performing matching system, but to provide a testbed for a fair comparison of different background knowledge sets and strategies.

3.1 Overview

Figure 1 depicts the architectural evaluation setting: A generic *matcher* accepts two ontologies and outputs an alignment. Thereby, it applies a *strategy* that can be exchanged independently of other matcher settings. Given labels, the matcher can ask a generic *linker* whether a concept is available in a background knowledge source. Depending on the request type, the linker returns one or more corresponding concepts from the background knowledge. For *Wiktionary*, for instance, the matcher can ask for concept **European Union** and the linker would return **dbnary-eng:European_Union**. This linking process is also known as *anchoring*

or *contextualization* [6]. Now that the matcher knows the representation in the background knowledge set, it can request further information through a generic *resource wrapper* (such as similarities between concepts). Therefore, a *resource* and a corresponding *linking process* (that is wrapped by the *linker*) have to be set. The implementation allows to change the *resource* and the *linking process* independently of other matcher settings such as the *strategy*.

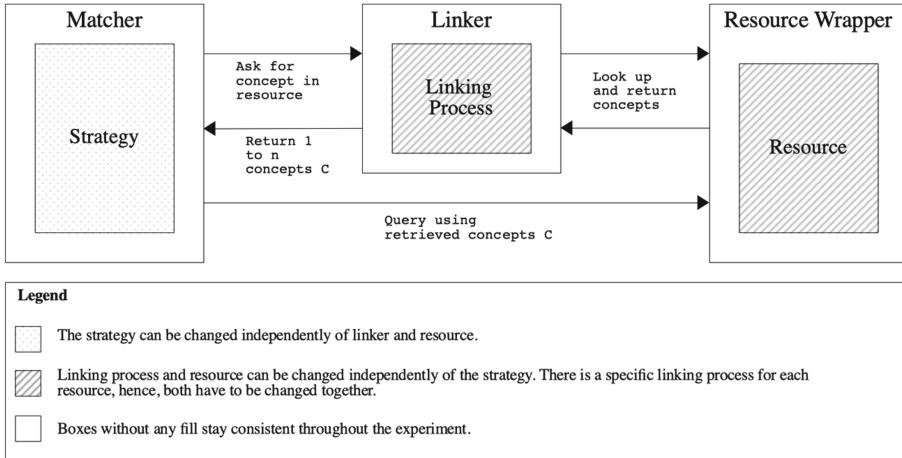


Fig. 1. Architectural setting to evaluate different background datasets exploiting different strategies.

3.2 Matching Process

The matching process can be divided into two parts: linking and matching. The linking operation is implemented as a three step process: (i) *Full Label Linking*, (ii) *Longest Token Linking*, and (iii) *Token Linking*. Later linking steps are only performed when the previous step was not able to link the label. In step (i), the full, i.e. unchanged, label is linked to a concept in the background knowledge source. Often, labels are composite concepts that do not appear in the knowledge source as a whole but in parts. To cover this case, step (ii) tokenizes labels and truncates them from the right. Linked parts are removed and the process is repeated to check for further concepts. This allows to detect long sub-concepts even if the full string cannot be found. Label *conference banquet*, for example, cannot be linked to the Wiktionary background dataset using the full label. However, by applying right-to-left truncation, the label can be linked to two concepts, namely *conference* and *banquet*, and in the following also be matched to concept *conference dinner* which is linked in the same fashion. The last fallback strategy is token linking (iii) which tokenizes each label (using spaces, underscores, and camel case recognition) and links the individual tokens to the background dataset.

After completion of the linking process, the match operation is performed. Multiple strategies are implemented here (see Sect. 5) which operate on the links. For the synonymy strategy, a match would be, for instance, annotated for (*person*, *individual*) given that the two labels are synonymous according to the background dataset employed. If there are multiple links (linking steps (ii) and (iii)), a match requires that every link has a matching partner (according to the strategy applied) in the set of links of the other label. In order to obtain a one-to-one alignment, the Hungarian extraction method [19] is applied.

The overall matching runtime performance is improved by adding string matches directly to the final alignment. This step runs independently of the strategy or the background dataset used. It does not skew the outcome because all strategies under consideration in this paper are purely label-based. Hence, the same label used for two entities would always lead to a match.

Overall, the matching process scales with $O(nm)$ where n is the number of elements in one ontology and m is the number of elements in the other ontology.² It is important to note that the scalability can be improved by adding a candidate pre-selection/blocking component. However, since scalability is not the main concern of this paper, we decided against complicating the matching pipeline.

The matcher is implemented using the *Matching EvaLuation Toolkit* [15, 16] (*MELT*)³, an open-source Java framework for matcher development, tuning, evaluation, and packaging recommended by the OAEI. The matcher is implemented so that it is possible to use different sources of background knowledge and different strategies within the matching process. The implementation of this paper (linker, background sources, significance evaluation) has been unit tested, documented, and contributed to the framework so that other researchers can use the matching parts of the implementation (e.g. to easily use Wikidata synonyms/hypernyms through an API) for their matching system.⁴

4 Background Datasets

For this paper, six knowledge graphs are exploited as background knowledge within the matching process. They are quickly introduced in the following:

BabelNet [28] is a large multilingual knowledge graph that integrates (originally) Wikipedia and WordNet. Later, additional resources such as Wiktionary were added. The integration between the resources is performed in an automated manner. The dataset does not just contain lemma-based knowledge but also instance data (named entities) such as the singer and songwriter *Trent Reznor*.

² The size of the external resource is not relevant within the matching process since all similarity functions applied here are lookup-based. When training an embedding with the external resource, the size of the resource affects scalability; however, the training is a one-time process – once the vectors are available, they can be reused in all other matching tasks.

³ <https://github.com/dwslab/melt/>.

⁴ <https://dwslab.github.io/melt/matcher-development/with-background-knowledge>.

For the embedding strategy, the RDF version of BabelNet 3.6 was used⁵, for the other strategies, the BabelNet 4.1 indices.

Wiktionary is a “collaborative project run by the Wikimedia Foundation to produce a free and complete dictionary in every language”⁶. The project is organized similarly to Wikipedia: Everybody can contribute and edit the dictionary. The content is reviewed in a community process. Like Wikipedia, Wiktionary is available in many languages.

DBnary [38] is an RDF version of Wiktionary that is publicly available.⁷ The DBnary dataset makes use of an extended *LEMON* model [24] to describe the data. For this work, a recent download from March 2021 of the English Wiktionary has been used.

WebIsALOD is a large hypernymy graph based on the *WebIsA* database [37]. The latter is a dataset which consists of hypernymy relations extracted from the *Common Crawl*, a large set of crawled Web pages. The extraction was performed in an automatic manner through Hearst-like [11] lexico-syntactic patterns. For example, from the sentence “[...] added that the country has favourable economic agreements with major economic powers, including the European Union.”, the fact `isA(european_union, major_economic_power)` is extracted.⁸ *WebIsA-LOD* [13] is the Linked Open Data endpoint which allows to query the data in SPARQL.⁹ In addition to the endpoint, machine learning was used to assign confidence scores to the extracted triples. For this work, a confidence threshold of 0.5 for hypernymy relations was chosen. The dataset of the endpoint is filtered, i.e. it contains a subset of the original *WebIsA* database, to ensure a higher data quality. The knowledge graph contains instances as well as more abstract concepts that can also be found in a dictionary.

WordNet [8] is a well-known and heavily used database of English words that are grouped in sets which represent one particular meaning, so called *synsets*. The resource is strictly authored. *WordNet* is publicly available, included in many natural language processing frameworks, and often used in research. An RDF version of the database is also available for download and was used for this work.¹⁰

Wikidata [40] is a collaboratively built knowledge base containing more than 93 million data items. Like Wikipedia and Wiktionary, the project is run by the Wikimedia Foundation. It is publicly available¹¹ and under a permissive license. For this work, a download from March 2021 has been used.

⁵ Unfortunately, there is no RDF version of the latest BabelNet version.

⁶ <https://web.archive.org/web/20190806080601/https://en.wiktionary.org/wiki/Wiktionary/>.

⁷ <http://kaiko.getalp.org/about-dbnary/download/>.

⁸ <http://webisa.webdatacommons.org/417880315>.

⁹ <http://webisa.webdatacommons.org/>.

¹⁰ <http://wordnet-rdf.princeton.edu/about/>.

¹¹ https://www.wikidata.org/wiki/Wikidata:Main_Page.

DBpedia [21] is a knowledge graph that is extracted from Wikipedia infoboxes. The underlying RDF files are available for download. For this work, the latest available files as of March 2021 have been downloaded via the DBpedia Databus¹² (rather than the 2016-10 version of DBpedia that is often used).

5 Strategies

In the following, the exploitation strategies applied on the datasets outlined in the previous section are introduced.

5.1 Synonymy

The synonymy strategy exploits existing synonymy relations in the datasets. On *Wiktionary*, for instance, *tired* is explicitly named as a synonym for *sleepy*. Given two entities $e_1 \in O_1$ and $e_2 \in O_2$ of two ontologies O_1 and O_2 , a match is annotated if the synonymy relation holds between at least one pair of their labels l_{e_1} and l_{e_2} according to the background dataset B that is used. This is depicted in Eq. 1.

$$\text{isMatch}_B(e_1, e_2) = \text{isSynonymous}_B(l_{e_1}, l_{e_2}) \quad (1)$$

The WebIsALOD dataset does not contain explicitly stated synonyms. Here, a synonym is assumed if both labels l_{e_1} and l_{e_2} appear as hypernyms of each other as shown in Eq. 2. This occurs more often than one might assume due to the automatic extraction process that is applied to create this knowledge graph.¹³ The intuition behind the assumption here is that two things X and Y are describing the same thing if it was stated on the Web that X is a Y and that Y is an X .

$$\text{isMatch}_{\text{WebIsALOD}}(e_1, e_2) = \text{isHypernymous}(l_{e_1}, l_{e_2}) \wedge \text{isHypernymous}(l_{e_2}, l_{e_1}) \quad (2)$$

For DBpedia, the properties `rdfs:label`, `foaf:name`, `dbo:alias`, `dbp:name`, and `dbp:otherNames` are used to obtain labels, and two entities are considered synonymous if they have at least one label in common. On Wikidata, we use `rdfs:label` and `skos:altLabel` to obtain labels, and determine synonymy with the same mechanism.

5.2 Synonymy and Hypernymy

The synonymy and hypernymy strategy exploits the synonymy relations in the background datasets and, in addition, the hypernymy relations. Given two labels

¹² <https://databus.dbpedia.org/>.

¹³ For example, *symposium* and *conference* are mutual hypernyms of each other in WebIsALOD.

l_{e_1} and l_{e_2} of two entities e_1 and e_2 , a match is annotated if one of the semantic relations holds between the two labels as depicted in Eq. 3.

$$\begin{aligned} isMatch_B(e_1, e_2) = & isSynonymous_B(l_{e_1}, l_{e_2}) \\ \vee isHypernym_B(l_{e_1}, l_{e_2}) \vee isHypernym_B(l_{e_2}, l_{e_1}) \end{aligned} \quad (3)$$

For DBpedia, the properties `rdf:type` and `dbo:type` are used to obtain hypernyms. On Wikidata, we use `wdt:P31` (instance of) and `wdt:P279` (subclass of).

5.3 Knowledge Graph Embeddings

Knowledge graph embeddings, i.e. the vector-based representation of the elements within a knowledge graph, are a very active research area in recent years. Many such methods are known [18]. For this paper, we exploit the *RDF2Vec* [36] approach: Random walks through the knowledge graph are generated starting from each node. The walks include the named edges of the graph. After the walk generation, the *word2vec* [25] algorithm is applied. Thereby, a vector representation for each node and each edge is obtained. This embedding approach has been chosen due to its simplicity, its good performance on a multitude of tasks (rather than being developed for only one task, *RDF2Vec* is task agnostic), its previous usage in ontology matching, and its scalability. It is important to note that the background knowledge source is transformed into a vector space – not the ontologies that are to be matched.

Two entities $e_1 \in O_1$ and $e_2 \in O_2$ of two different ontologies O_1 and O_2 are matched if their labels l_{e_1} and l_{e_2} can be mapped to a vector $v_{l_{e_1}}$ and $v_{l_{e_2}}$ in the background knowledge dataset B and the cosine similarity sim between the two vectors is larger than a predefined threshold t . Hence:

$$isMatch_B(e_1, e_2) = sim(v_{l_{e_1}}, v_{l_{e_2}}) > t \quad (4)$$

For WebIsALOD and WordNet, the pre-trained models from *KGvec2go*¹⁴ [32] were used. The models were trained with the same configuration and, therefore, allow for comparability. Embeddings for the other three graphs are not available for download and were trained specifically for this paper.

Despite good scalability behavior of the embedding approach, vector representations for BabelNet, Wikidata, and DBpedia could not be calculated within 10 days. Therefore, *RDF2Vec Light* [33] was used for those very large knowledge graphs. The variant is based on the notion that, given a concrete task, only a small set of nodes within a knowledge graph are of actual interest. For example, given the matching task within the anatomy domain, a vector representation of *Year Zero*, a music album by the industrial rock band *Nine Inch Nails*, is not of particular interest. Therefore, a set of nodes of interest is defined in advance and walks are only generated for those. For ontology matching, the set of nodes of interest is known through the linking operation. Experiments showed that the performance of the light variant yields good results on various machine learning

¹⁴ <http://kgvec2go.org/>.

tasks compared to the classic variant [33]. For this work, the following parameters have been used: 500 walks per node, $depth = 4$ (i.e., 4 node hops), SG variant, $window = 5$, and $dimension = 200$. For the matcher configuration, a threshold of $t = 0.7$ was used.

5.4 Combination of Sources

The combination strategy exploits all datasets at the same time with the strategies mentioned above. For the synonymy strategy, a match is annotated if any background dataset finds evidence for a synonymy relation. The same logic is also applied in the synonymy and hypernymy strategy and the embedding strategy.

6 Evaluation

We evaluate all combinations of the strategies presented in Sect. 5 and background datasets presented in Sect. 4 on four evaluation datasets: (i) *OAEI Anatomy* [2], (ii) *OAEI Conference* [3], (iii) *SAP FS* [30], and (iv) *LargeBio*. The experiments were performed on a 24 core server (≥ 2.6 GHz) with 386 Gb of RAM running Debian 10.

6.1 Evaluation Datasets

Dataset (i) consists of two anatomical ontologies where the human anatomy has to be mapped to the anatomy of a mouse. The *Conference* dataset (ii) consists of 16 ontologies from the conference domain and 120 alignment tasks between them. Out of those, 21 reference alignments are publicly available. The results reported in this paper refer to the available alignments. In order to allow for comparability with other matching systems, micro averages are reported; those are also reported by the *OAEI Conference* track organizers. The *SAP FS* dataset (iii) is a proprietary evaluation dataset from the banking and insurance industry consisting of 5 matching tasks. The ontologies in that dataset have been derived from conceptual data models. The dataset has been provided to the authors of this paper for research purposes by *SAP SE Financial Services*. In order to allow for comparability with the numbers reported in the original paper, macro averages are reported here. From the LargeBio track (iv), the FMA/NCI small test case is used for the evaluation here. Overall, 21 matching system variants are evaluated on four tracks with a total of 28 test cases.

6.2 Evaluation Metrics

The alignments are evaluated using precision, recall, and F_1 which is the harmonic mean of the latter two. In addition, it is evaluated whether the alignments obtained by the different strategy-source combinations are significantly different. Therefore, a significance metric is required. For this work, we use McNemar's significance test as proposed by Majid et al. [26]: Be R the reference alignment and

A_1 , A_2 two system alignments. We can now calculate the two relevant elements from the contingency table as follows:

$$\begin{aligned} n_{01} &= |(A_2 \cap R) - A_1| + |A_1 - A_2 - R| \\ n_{10} &= |(A_1 \cap R) - A_2| + |A_2 - A_1 - R| \end{aligned} \quad (5)$$

The significance can then be determined using McNemar's asymptotic test with continuity correction:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \quad (6)$$

For a small sample size ($n = n_{01} + n_{10}$; $n < 25$), McNemar's exact test has to be used to obtain the p value:

$$p = \sum_{x=n_{01}}^n \left(\binom{n}{x} \right) \left(\frac{1}{2} \right)^2 \quad (7)$$

For this paper, a significance level alpha of $\alpha = 0.05$ was chosen. As a side contribution of this work, the evaluation code for significance testing has been contributed to the MELT framework [15] to facilitate reuse by other researchers.

6.3 Results

The performance results in terms of precision, recall, and F_1 are presented in Table 1. The number of significantly different test case alignments is given in Fig. 2. More detailed performance and significance statistics as well as all alignments are available for download.¹⁵ It can be seen that the synonymy strategy consistently achieves the highest precision throughout all background knowledge resources. In terms of F_1 , the synonymy strategy performs best in most cases when evaluating the strategy on each background source separately. The only area where the synonymy strategy falls short is recall. The significance tests show that despite similar scores, the alignments within this strategy group are significantly different in 285 out of 588 cases. This is also visible in Fig. 2 which shows the number of significantly different alignments (given two matching systems). From the figure, it can be seen, for instance, that there are 22 significantly different alignments between DBpedia and Wiktionary using the synonymy strategy but only 5 different alignments between DBpedia and the combination approach using the synonymy strategy.

With the exception of BabelNet, the addition of hypernyms increases recall.¹⁶ However, a drop in precision leads to overall lower F_1 scores (with the exception of DBpedia on SAP FS and Wikidata on FMA/NCI). The results indicate that hypernyms could be used in more complex matching strategies, e.g. as part of candidate generation. Nonetheless, a naïve merge of synonymy and hypernymy

¹⁵ <https://github.com/janathan/bk-strategy-vs-data-supplements/>.

¹⁶ This may seem odd at first. However, lower recall values are due to the Hungarian optimization method to obtain a 1:1 alignment, which, in that case, extracts more false positives.

Table 1. Evaluation results of six different background knowledge datasets exploiting three different strategies on four different gold standards. Note that for the *Conference* task, micro averages are used while for the *SAP FS* task, macro averages are reported. Baselines are given as reported by the OAELI organizers (FMA/NCI baselines are not provided).

Knowledge graph/Strategy	SAP FS					
	Conference	Anatomy	R	P	R	P
BabelNet	P	0.75197889 0.8388	0.67711	0.5655895 0.6117	0.40352626 0.15272657	0.22115869 0.3596947
	SYN + HYP	0.9316405 0.75361701	0.83536349 0.56206896	0.5344126 0.54789159	0.152836758 0.152836758	0.169340659 0.366344
WebALOD	RDF2Vec (light)	0.51411879259 0.6919525	0.690653908 0.690653886	0.26880119 0.28874225	0.2061385478 0.137710	0.16511569 0.318797
	SYN	0.96682027 0.693250639	0.8066128 0.79010919470.4577612	0.4950819 0.56554307	0.13775575 0.14767575	0.143885038 0.22903382
Wiktionary	SYN + HYP	0.91493065 0.701118355	0.6932506390.701118355	0.53472355 0.540836	0.49425154 0.3492426	0.14610464 0.2063198
	RDF2Vec	0.788575667 0.711741	0.7423184350.706672964	0.5390859 0.60295878	0.26073486 0.145607	0.1392453405 0.2210721
WordNet	SYN + HYP	0.9068 0.7124010655	0.82902320 0.6714887119	0.534426 0.53854145	0.4589141 0.45874933	0.145607 0.222107330
	RDF2Vec	0.644016337 0.9635555	0.7064613790.76757967	0.52459016 0.63795227	0.45874 0.14272446	0.137179487 0.19032604
DBpedia	SYN + HYP	0.92875318 0.732222955	0.82162159 0.61363636	0.5311475 0.569120	0.4152099 0.3741607	0.148036455 0.2182367
	RDF2Vec	0.732222958 0.935569285	0.709102960 0.63909208	0.504672897 0.49508196	0.3511475 0.38011583	0.1567717 0.203513597
Wikidata	SYN + HYP	0.935569 0.593905817	0.800302 0.707892	0.6399208 0.49508196	0.583011583 0.583011583	0.1911020 0.2422
	RDF2Vec (light)	0.7072124 0.7141238	0.67072124 0.62928404	0.514754098 0.6360115	0.23751189 0.224630590	0.1519484469 0.16317916
Combinations	SYN + HYP	0.92332105 0.92332121	0.7141238 0.7141242	0.544262295 0.544262295	0.586575 0.446360	0.146557 0.1631796
	RDF2Vec	0.5464720194 0.881918819	0.74076517 0.7882558	0.62295467 0.49377164	0.2437966 0.1777831	0.17090225 0.24141958
Baseline	-	0.997 0.997	0.78627 0.63698	0.404960 0.620	0.194 0.24183006	0.160778 0.162076
	-	-	0.6322 0.766	0.41 0.53	0.15 0.52	- -

Combined RDF2Vec	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28
Combined SYN	28	5	27	10	5	27	5	5	14	7	4	28	10	10	16	9	9	4	7	4		
Combined SYN + HYP	28	5	26	9	7	26	11	11	5	9	3	27	6	6	10	10	10	11	11	7		
BabelNet RDF2Vec	28	27	26	28	28	3	28	28	24	28	28	4	28	28	25	28	28	28	28	28	27	
BabelNet SYN	28	10	9	28		13	27	18	17	13	14	5	28	12	11	13	15	15	6	21	13	
BabelNet SYN + HYP	28	5	7	28	13		27	9	9	15	9	6	28	7	7	13	8	7	4	8	15	
DBpedia RDF2Vec	28	27	26	3	27	27		28	28	23	27	28	7	28	28	17	28	28	27	27	26	
DBpedia SYN	28	5	11	28	18	9	28			13	19	8	28	11	11	15	22	21	7	18	12	
DBpedia SYN + HYP	28	5	11	28	17	9	28	1		13	19	8	28	11	11	15	22	21	7	18	12	
WebIsALOD EMB	28	14	5	24	13	15	23	13	13		15	15	26	12	12	6	16	16	13	13	14	
WebIsALOD SYN	28	7	9	28	14	9	27	19	19	15		9	28	10	10	14	17	16	7	20	10	
WebIsALOD SYN + HYP	28	4	3	28	5	6	28	8	8	15	9		28	6	6	15	4	3	4	6	4	
Wikidata RDF2Vec	28	28	27	4	28	28	7	28	28	26	28	28		28	28	26	28	28	28	28	28	
Wikidata SYN	28	10	6	28	12	7	28	11	11	12	10	6	28		2	14	10	10	4	13	10	
Wikidata SYN + HYP	28	10	6	28	11	7	28	11	11	12	10	6	28	2		14	10	10	4	13	11	
Wiktionary RDF2Vec	28	16	10	25	13	13	17	15	15	6	14	15	26	14	14		13	13	18	15	12	
Wiktionary SYN	28	9	10	28	15	8	28	22	22	16	17	4	28	10	10	13		8	4	17	11	
Wiktionary SYN + HYP	28	9	10	28	15	7	28	21	21	16	16	3	28	10	10	13	8		4	18	11	
WordNet RDF2Vec	28	4	11	28	6	4	27	7	7	13	7	4	28	4	4	18	4	4		10	6	
WordNet SYN	28	7	11	28	21	8	27	18	18	13	20	6	28	13	13	15	17	18	10		17	
WordNet SYN + HYP	28	4	7	27	13	15	26	12	12	14	10	4	28	10	11	12	11	11	6	17		
Combined RDF2Vec Combined SYN + HYP Combined SYN BabelNet RDF2Vec BabelNet SYN BabelNet SYN + HYP DBpedia RDF2Vec DBpedia SYN DBpedia SYN + HYP WebIsALOD EMB WebIsALOD SYN WebIsALOD SYN + HYP Wikidata RDF2Vec Wikidata SYN Wikidata SYN + HYP Wiktionary RDF2Vec Wiktionary SYN + HYP WordNet RDF2Vec WordNet SYN WordNet SYN + HYP																						

Fig. 2. Matrix with the number of significantly different test case alignments given two matcher configurations. A higher total number of significantly different test case alignments has a darker shading in the figure. In total, there are 28 test cases.

sets as main strategy is not generally suitable for precise matching on the given evaluation datasets.

The embedding-based approach falls short of performing competitively. While the recall can be increased in some cases, the method generally scores a significantly lower precision leading to an overall low F_1 score. One likely reason for the bad performance of the embeddings is that the RDF2Vec vector similarity seems to be an indicator for relatedness rather than actual concept similarity – an observation that has also been made earlier [32]. More promising usage scenarios for the embedding models exploited in this paper are likely candidate selection and hybrid strategies. Concerning significance testing, the embedding strategies produce the most significantly different alignments of all strategies evaluated in this paper. In addition, it was observed that embedding large background knowledge datasets is computationally very expensive which does not apply to the matching run time after the models were trained.

Concerning the choice of background knowledge, WordNet, Wiktionary, and BabelNet are similar in the sense that they are focused on lexical facts. BabelNet, the largest of the three, scores the overall best F_1 score on Anatomy and Conference. On the remaining two tracks, the performance is competitive.

Despite its small size, WordNet also achieves competitive results compared to Wiktionary on Anatomy, Conference, and SAP FS and outperforms the latter significantly on the LargeBio task. Nevertheless, unlike WordNet, Wiktionary and BabelNet are constantly growing over time due to a community-driven creation process and might outperform WordNet in the long run.

DBpedia performs in the mid-range in terms of F_1 . The recall is lower than that of the better performing systems (BabelNet, Wiktionary, WordNet). The most likely explanation is a lower concept coverage since DBpedia contains rather instances than class concepts. Interestingly, the addition of hypernyms has rarely any effect on this particular background source.

Wikidata performed similarly to DBpedia. Like the latter dataset, the addition of hypernyms does not change the results significantly.

The WebIsALOD dataset achieves the lowest overall results. The most likely reason is that the dataset is not authored but automatically built leading to a lot of noise contained in the dataset (wrong hypernyms). The comparatively bad performance of the synonymy strategy may be grounded in the fact that WebIsALOD is the only graph evaluated here that does not explicitly state synonyms – but instead those are derived, as outlined before, which is less precise.

The combination of different background knowledge sources increases the recall in all cases. Except on the LargeBio dataset, the drop in precision cannot make up for increases in recall.

When comparing the performance numbers on evaluation dataset level, it can be seen that the *Anatomy* matching task achieves the best results – this is likely due to a high textual overlap of the labels. On the *Conference* task, the matchers achieve a lower precision and recall score. These observations are in line with those at *OAEI* campaigns. On the domain specific *SAP FS* dataset, it can be seen that recall and precision scores are low. Likely explanations here are a domain specific vocabulary, low explicitness of knowledge (many semantic details are hidden in lengthy descriptions) as well as a complex many-to-many matching problem (see [30] for details).

It is important to note that the work presented here is not intended to be a full-scale matching system but rather a comparison of different background knowledge datasets and exploitation strategies. Nonetheless, the performance of the best matching results achieved here on *Anatomy* and *Conference* are comparable to *OAEI* matching results reported in the most recent 2020 campaign. A comparison in terms of F_1 is depicted in Fig. 3. It can be seen, that the best configuration of this paper performs in both cases above the median of the systems submitted in 2020. On Anatomy, it is noteworthy, that the first three systems (AML, Lily, and LogMapBio) use domain-specific resources leading to an advantage over the general-purpose resources exploited in this work.

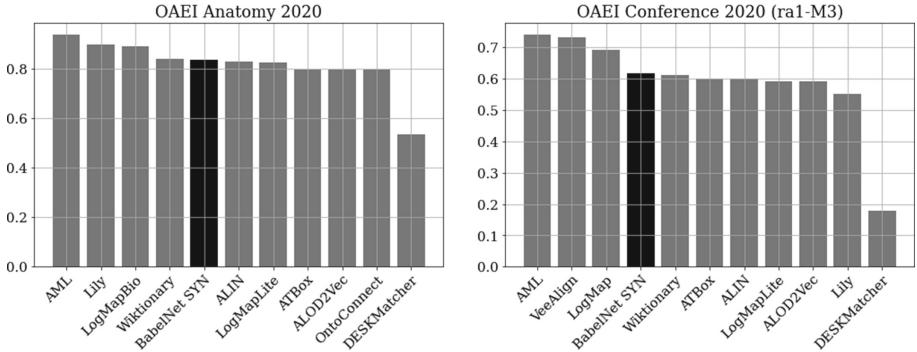


Fig. 3. Performance in terms of F_1 on the OAEI Anatomy and Conference tracks of 2020.

Hypotheses. In order to evaluate hypotheses 1 and 2, we averaged the relative share of significantly different alignments on all test cases (i) while keeping the background source constant and changing the strategy (Eq. 8) and (ii) while keeping the strategy constant and changing the background source (Eq. 9):

$$impact_{strategy} = \frac{\sum_{bk \in BK} \frac{\sum_{tc \in TC} \sum_{s_1 \in S} \sum_{s_2 \in S} sig(m(bk, s_1), m(bk, s_2))}{|TC| * |S|^2 - |TC| * |S|}}{|BK|} \quad (8)$$

$$impact_{source} = \frac{\sum_{s \in S} \frac{\sum_{tc \in TC} \sum_{bk_1 \in BK} \sum_{bk_2 \in BK} sig(m(bk_1, s), m(bk_2, s))}{|TC| * |BK|^2 - |BK| * |S|}}{|S|} \quad (9)$$

where S is the set of strategies, BK is the set of background sources, $sig(alignment_1, alignment_2)$ is the significance function which will return 1 if the two provided alignments are significantly different and else 0, and $m(bk, s)$ is the matching function which returns the alignment by using the specified background knowledge source bk and strategy s .

While keeping the background knowledge source constant and changing the strategy, we observed on average 57.5% significantly different alignments with a standard deviation of $\sigma = 0.163$. On the other hand, while keeping the strategy constant and changing the background knowledge source, we obtained on average 51.76% significantly different alignments with a standard deviation of $\sigma = 0.181$. Given our experimental setup, we hence accept H1 and reject H2 since a variation in the strategic component has a higher impact on the alignments than a variation of the background sources under consideration in this study. It is noteworthy that both components lead on average to more than 50% significantly different alignments. Since our results do not indicate that there is a superior resource over all test sets, we can reject H3. However, it is noteworthy that BabelNet achieves consistently good (on two tracks the best)

results in terms of F_1 when using the synonymy strategy. Similarly, we do not find a superior strategy over each and every single test case and reject H4 – but yet, the synonymy strategy achieved the best F_1 score on 3 out of 4 tracks and consistently performed very well compared to the other strategies.

7 Summary and Future Work

In this paper, we evaluated three different matching strategies using six different general purpose knowledge graphs on various evaluation datasets. We find that the strategy influences the final alignment more than the underlying dataset. Given the strategies evaluated here, those exploiting explicitly stated knowledge outperform a latent strategy. However, the exploitation of graph embeddings for data integration and schema matching is novel and its performance is still very low. While no superior general knowledge dataset could be identified, BabelNet produced consistently good or the best results. The humanly verified datasets outperformed the automatically generated one. Concerning the level of authoring between the datasets, the results indicate no clear superiority of expert-validated knowledge graphs over those created and validated by an open community.

In the future, we plan to exploit further embedding strategies, such as translational approaches, for schema matching as well as graph-based and dataset specific strategies. We further plan to examine more domain-specific matching tasks such as the *SAP FS* dataset.

References

1. Annane, A., Bellahsene, Z., Azouaou, F., Jonquet, C.: Selection and combination of heterogeneous mappings to enhance biomedical ontology matching. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) EKAW 2016. LNCS (LNAI), vol. 10024, pp. 19–33. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49004-5_2
2. Bodenreider, O., Hayamizu, T.F., Ringwald, M., de Coronado, S., Zhang, S.: Of mice and men: aligning mouse and human anatomies. In: AMIA 2005 (2005)
3. Cheatham, M., Hitzler, P.: Conference v2.0: an uncertain version of the OAEI conference benchmark. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 33–48. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_3
4. Chen, X., Xia, W., Jiménez-Ruiz, E., Cross, V.V.: Extending an ontology alignment system with bioportal: a preliminary analysis. In: ISWC 2014 Posters & Demonstrations Track. CEUR Workshop Proceedings, vol. 1272, pp. 313–316 (2014)
5. Euzenat, J.: Ontology Matching, 2nd edn. Springer, New York (2013). <https://doi.org/10.1007/978-3-642-38721-0>
6. Euzenat, J., Shvaiko, P.: Matching strategies. In: Ontology Matching, pp. 149–197. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38721-0_7
7. Faria, D., Pesquita, C., Santos, E., Cruz, I.F., Couto, F.M.: Automatic background knowledge selection for matching biomedical ontologies. PloS One **9**(11) (2014)
8. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Language, Speech, and Communication. MIT Press, Cambridge (1998)

9. Groß, A., Hartung, M., Kirsten, T., Rahm, E.: Mapping composition for matching large life science ontologies. In: Proceedings of the 2nd International Conference on Biomedical Ontology. CEUR Workshop Proceedings, vol. 833 (2011)
10. Hartung, M., Gross, A., Kirsten, T., Rahm, E.: Effective composition of mappings for matching biomedical ontologies. In: Simperl, E., et al. (eds.) ESWC 2012. LNCS, vol. 7540, pp. 176–190. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46641-4_13
11. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: 14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, 23–28 August 1992, pp. 539–545 (1992)
12. Hertling, S., Paulheim, H.: WikiMatch - Using Wikipedia for ontology matching. In: OM@ISWC 2012, vol. 946, pp. 37–48 (2012)
13. Hertling, S., Paulheim, H.: WebIsALOD: providing hypernymy relations extracted from the web as linked open data. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10588, pp. 111–119. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68204-4_11
14. Hertling, S., Paulheim, H.: DOME results for OAEI 2018. In: OM@ISWC 2018, pp. 144–151 (2018)
15. Hertling, S., Portisch, J., Paulheim, H.: MELT - matching EvaLuation toolkit. In: Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y. (eds.) SEMANTiCS 2019. LNCS, vol. 11702, pp. 231–245. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33220-4_17
16. Hertling, S., Portisch, J., Paulheim, H.: Supervised ontology and instance matching with MELT. In: OM@ISWC 2020. CEUR Workshop Proceedings, vol. 2788, pp. 60–71 (2020). [CEUR-WS.org](http://ceur-ws.org)
17. Kachroudi, M., Diallo, G., Yahia, S.B.: KEPLER at OAEI 2018. In: OM@ISWC 2018, pp. 173–178 (2018)
18. Kazemi, S.M., et al.: Relational representation learning for dynamic (knowledge) graphs: a survey. CoRR abs/1905.11485 (2019)
19. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logistics Q. **2**(1–2), 83–97 (1955)
20. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning (ICML), pp. 1188–1196 (2014)
21. Lehmann, J., et al.: DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. Semant. Web **6**(2), 167–195 (2015)
22. Lin, F., Krizhanovsky, A.: Multilingual ontology matching based on Wiktionary data accessible via SPARQL endpoint. In: RCDL 2011. CEUR Workshop Proceedings, vol. 803, pp. 1–8 (2011)
23. Lütke, A.: AnyGraphMatcher submission to the OAEI knowledge graph challenge 2019. In: OM@ISWC 2019 (2019)
24. McCrae, J., et al.: Interchanging lexical resources on the Semantic Web. Lang. Resour. Eval. **46**(4), 701–719 (2012)
25. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: 27th Annual Conference on Neural Information Processing Systems 2013, pp. 3111–3119 (2013)
26. Mohammadi, M., Atashin, A.A., Hofman, W., Tan, Y.: Comparison of ontology alignment systems across single matching task via the McNemar's test. ACM Trans. Knowl. Discov. Data **12**(4), 51:1–51:18 (2018)
27. Monych, M., Portisch, J., Hladík, M., Paulheim, H.: DESKMatcher. In: OM@ISWC 2020. CEUR Workshop Proceedings, vol. 2788, pp. 181–186 (2020)

28. Navigli, R., Ponzetto, S.P.: BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.* **193**, 217–250 (2012)
29. Paulheim, H.: WeSeE-match results for OEAII 2012. In: OM@ISWC 2012 (2012)
30. Portisch, J., Hladík, M., Paulheim, H.: Evaluating ontology matchers on real-world financial services data models. In: Posters & Demos of SEMANTiCS 2019 (2019)
31. Portisch, J., Hladík, M., Paulheim, H.: Alod2vec matcher results for OAEI 2020. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) OM@ISWC 2020. CEUR Workshop Proceedings, vol. 2788, pp. 147–153 (2020)
32. Portisch, J., Hladík, M., Paulheim, H.: KGvec2go - knowledge graph embeddings as a service. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC), Marseille, France (2020)
33. Portisch, J., Hladík, M., Paulheim, H.: Rdf2Vec light - a lightweight approach for knowledge graph embeddings. In: ISWC 2020 Demos and Industry Tracks. CEUR Workshop Proceedings, vol. 2721, pp. 79–84 (2020)
34. Portisch, J., Paulheim, H.: Wiktionary matcher results for OAEI 2020. In: OM@ISWC 2020. CEUR Workshop Proceedings, vol. 2788, pp. 225–232 (2020)
35. Quix, C., Roy, P., Kensche, D.: Automatic selection of background knowledge for ontology matching. In: SWIM 2011, p. 5. ACM (2011)
36. Ristoski, P., Rosati, J., Noia, T.D., Leone, R.D., Paulheim, H.: Rdf2vec: RDF graph embeddings and their applications. *Semant. Web* **10**(4), 721–752 (2019)
37. Seitner, J., et al.: A large database of hypernymy relations extracted from the web. In: LREC 2016 (2016)
38. Sérasset, G.: DBnary: Wiktionary as a lemon-based multilingual lexical resource in RDF. *Semant. Web* **6**(4), 355–361 (2015)
39. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *IEEE Trans. Knowl. Data Eng.* **25**(1), 158–176 (2013)
40. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. *Commun. ACM* **57**(10), 78–85 (2014)