Exploring the Synergies between Biocuration and Ontology Alignment Automation

David Dearing and Terrance Goan

Stottler Henke Associates, Inc.
1107 NE 45th St, Suite 310
Seattle, WA 98105, USA
{ddearing, goan}@stottlerhenke.com

Abstract. Researchers have long recognized the value trapped in natural language publications and have continued to advance the development of ontologies that can help unleash this value. Among these advances are efforts to apply NLP techniques to streamline the labor-intensive process of scientific literature curation, which encodes relevant information in a form that is accessible to both humans and computers. In this paper, we report on our initial efforts to improve ontology alignment within the context of scientific literature curation by exploiting value within a large corpus of annotated PubMed abstracts. We employ an ensemble learning approach to augment a collection of publicly available ontology matching systems with a matching technique that leverages the word embeddings learned from this corpus in order to more successfully match the concepts of two disease ontologies (MeSH and OMIM). Our experiments show that word embedding-based similarity scores do contribute value beyond traditional matching systems. Our results show that the performance of an ensemble trained on a small number of manually reviewed mappings is improved by their inclusion.

Keywords: Ontology Matching Ensembles, Word Embeddings, Biocuration.

1 Introduction

Technological advancements have given rise to an explosion in the rate that biomedical data is generated. The incredible volume of data now far exceeds the ability of researchers to capitalize on it. This is due, in large part, to the vagaries of the natural languages in which that data is published for consumption by human readers. The wide variety of lexical forms employed in the research literature present persistent challenges for both humans and computers in finding, assessing, and assimilating relevant data.

The research community has long recognized the value trapped in natural language publications and has continued to advance the development of ontologies that can mitigate the challenges posed by natural language. Today, ontologies are a critical foundation for emerging technologies that seek to better inform and accelerate biomedical research. Notable among recent advances are efforts to apply Natural Language Processing (NLP) techniques to streamlining the labor-intensive processes of biocuration and systematic scientific reviews.

Biocuration involves the interpretation, representation, and integration of information relevant to biology into a form that is accessible to both humans and computers. This process results in databases or knowledgebases (e.g., UniProt [1], NCBI Database Resources [2], and the Rat Genome Database (RGD) [3]) that assimilate the scientific literature as well as large data sets. Biocuration efforts range in both approach and scope, but they are increasingly supported by automated tools that facilitate information triage and tagging [4, 5].

Similar to biocuration is the systematic review: a literature review that gathers and analyzes research literature according to a structured methodology and guided by one or more specific research questions. The aim of systematic review is to produce an exhaustive summary of current literature relevant to those research questions. Sometimes a systematic review is simply an instance of a biocuration effort without sufficient resources to codify the collected knowledge [6]. As with biocuration, there are increasing efforts to employ natural language processing and other artificial intelligence methods to streamline an expert-driven process that is otherwise very labor intensive [7-10].

Biocuration and systematic review processes (whether manual or automated) are complicated by the applicability of overlapping ontologies that cover a breadth of multispecies knowledge that ranges across biological scales from molecules to populations. Ultimately, the exploitation of numerous (but well-aligned) ontologies will provide a comprehensive landscape of biomedical knowledge that will speed the identification of new hypotheses and avenues of investigation.

In this paper, we report on our initial efforts to improve ontology alignment within the context of scientific literature curation. More specifically, we describe an ensemble learning approach that augments a collection of ontology matching systems with word embeddings generated from an annotated corpus of relevant scientific literature.

The rest of this paper is organized as follows: In the next section, we provide background and discuss related work. In Sections 3 and 4 we describe our experiments, research hypothesis, and results. Finally, in Section 5, we summarize our conclusions and plans for future work, including extensions that support learning from work-centered user interactions.

2 Background and Related Work

The best-performing ontology matching tools all rely on collections of complementary matchers in order to compensate for context-specific weaknesses of each contributing/competing heuristic. The challenge of matcher selection and evidence combination has been addressed in a variety of ways ranging from ad hoc rules and manual settings [11] to ensemble learning methods [12, 13] that utilize machine learning to select and weight contributing matchers. Methods, such as "mapping gain" measurement, are applicable to the related challenge of selecting appropriate background knowledge sources [14].

Background knowledge sources play an important role in the performance of ontology matching tools. While string distance measures and taxonomic structure comparison form the backbone of most tools for ontology matching, it is also widely recognized that ontologies constructed by independent experts can differ significantly in both organization and lexical features. In these situations, researchers commonly seek to bridge the gap by drawing on various sources of background knowledge, such as: other ontologies, thesauri, lexical databases, online encyclopedias, and text corpora [11, 14]. These knowledge sources can then be used to implement matching functions that account for spelling variations and synonyms, and that also support some measure of semantic comparison [15].

One approach to measuring semantic similarity of elements is to employ WordNet similarity [16]. However, WordNet offers little coverage of concepts found in real-world ontologies. Another approach is to learn word embeddings directly from text corpora. Word embeddings are distributed word representations that are trained through deep neural networks. Each dimension of the embeddings represents a latent feature of the word, often capturing useful syntactic and semantic properties [17].

Word embeddings have proved to be useful at improving the performance of a wide range of Natural Language Processing (NLP) tasks [18]. Zhang et al. [15] showed that word embeddings learned over Wikipedia can improve the effectiveness of matcher ensembles applied to OAEI benchmark, conference track, and real-world ontologies.

Our own work is similar to that of Zhang et al. [15] but is differentiated in two primary ways. First, we learn word embeddings from a corpus of annotated scientific literature related to the ontologies to be aligned, rather than from Wikipedia. Second, we employ ensemble learning to integrate open source ontology matchers with our word embedding based matcher.

3 Experimental Setup

Our research centers on the hypothesis that the information gleaned from the word embeddings learned from a relevant, annotated corpus would improve matching results within a learned ensemble of existing open source ontology matchers. We tested this hypothesis with systematic experiments using the datasets and techniques described in the following.

3.1 Datasets

To evaluate our ensemble matching system, we used two ontologies of disease vocabularies: the subset of the Online Mendelian Inheritance in Man (OMIM) disease vocabulary, a flat list of disease terms covering genetic disorders; and the ‘Diseases’ branch of the National Library of Medicine’s Medical Subject Headings (MeSH). A third vocabulary, the Comparative Toxicogenomics Database’s (CTD) ‘merged disease vocabulary’ (MEDIC) [19] serves as a reference alignment between OMIM and MeSH. We chose these datasets primarily because there exists a corpus of PubMed titles and abstracts where disease mentions are annotated with the corresponding MEDIC identifiers—such a corpus is needed to train the model from which we train the underlying neural network for our word embedding matcher. In particular, PubTator (a Web-based tool for accelerating manual literature curation) provides an archive of the computer annotation results for the entire collection of PubMed articles in PubTator[1]. This computer-annotated corpus is generated using the DNorm tool for disease named entity recognition [20].

The data files for our ontologies were collected at the end of 2015 for the MeSH, OMIM, and MEDIC disease vocabularies. The ontology for the MeSH 'Diseases' branch includes 11,344 concepts. The ontology of OMIM genetic disorders includes 8,064 concepts. The MEDIC reference alignment identifies 3,435 direct mappings between MeSH and OMIM concepts. Lastly, the entire PubTator corpus contains 14,412,044 documents.

3.2 Word Embedding Matcher (Word2vec)

Our word embedding matcher uses the similarity scores, as learned by the Word2vec component of the Deeplearning4j library [21], as the confidence for a match between a given pair of ontology concepts. Word2vec is a two-layer neural net that processes text, taking a text corpus as input and outputting a set of feature vectors for words in the corpus. The vectors used to represent words are called *neural word embeddings* and represent a word with numbers based on other neighboring words within the input corpus (see **Table 1**). Given a large enough corpus, Word2vec can make highly accurate guesses about a particular word's meaning—without human intervention—based solely on numerical representations of word features, such as the context of individual words. Word embedding similarity scores are calculated as the cosine similarity of the vectors for a pair of concepts in the MeSH and OMIM ontologies.

Table 1. Examples of neural word embedding vectors learned from the PubTator corpus.

bone	*marrow, (bmt), solid-organ, disseminated, allogeneic, ...*
blood	*pressure, rate, hypotension, arterial, concentration, ...*
heart	*rate, cardiac, re-infarction, pressure, o2, arterial, ...*
liver	*renal, hepatic, failure, acute, function, chronic, ...*

Before training the Word2vec model, we preprocess the PubTator corpus so that the annotated phrases for each PubMed document (title and abstract) are replaced by a unique single-token identifier for the corresponding MeSH or OMIM concept. This is necessary because Word2vec learns similarity vectors based on individual words/tokens (and not multi-word phrases). The unique identifier allows us to look up similarity scores for a given pair of concepts from the trained word embedding model. We used Deeplearning4j's suggested configuration: a word window size of 10 for calculating within-sentence word context and the skip-gram technique for predicting the target context, which produces more accurate results on large datasets.

[1] https://ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/tutorial/index.html#DownloadFTP

Training the Word2vec model for more than 14 million documents is very time consuming (on the order of weeks). Once the model is built, however, extracting the similarity score for a given pair of terms is fast. The training time can be reduced by distributing the processing with, for example, an Apache Spark cluster.

3.3 Ontology Matching Systems

In addition to the word embedding matcher, we also utilized a number of publicly available ontology matching systems. These matchers are used both alone and as part of a learned ensemble to evaluate the relative impact of the addition of our word embedding matcher. These systems have all participated in past Ontology Alignment Evaluation Initiative (OAEI) campaigns.

LogMap. LogMap [22] is a scalable ontology matching system that utilizes highly optimized data structures to index the input ontologies (both lexically and structurally) to compute an initial set of anchor mappings with corresponding confidence values. These anchors are then used in an iterative process of mapping repair and mapping discovery to uncover new mappings.

AgreementMakerLight (AML). AML is an ontology matching framework based on AgreementMaker [23], one of the leading ontology matching systems. However, whereas AgreementMaker is memory-intensive and was not designed to match ontologies with more than a few thousand concepts, AML is a lightweight system developed with a focus on computational efficiency and is specialized on the biomedical domain but applicable to any ontologies.

Generic Ontology Matching and Mapping Management (GOMMA). GOMMA provides a comprehensive and scalable infrastructure to manage large life science ontologies, but as a generic tool it can be used to match ontologies from other domains [24]. GOMMA preprocesses all information relevant for matching ontology concepts (e.g., name, synonyms, comments) and uses maximal string similarity to generate matches before aggregating the mappings, filtering out any mappings below a certain threshold, and applying constraints to improve the consistency of mappings.

(not) Yet Another Matcher (YAM++). The underlying idea of the YAM++ system is that the complexity and, therefore, the cost of the ontology matching algorithms can be reduced by using indexing data structures to avoid exhaustive pair-wise comparisons [25]. YAM++ preprocesses the input ontologies to calculate the information content of each word to determine the weights of labels. Candidate mappings are passed to a process that uses machine learning to combine several different string-based comparisons to compare the labels/synonyms of entities. Those results are then passed to a structural matcher, which looks at related entities to find more mappings, before combining and filtering the results.

Falcon-AO. Falcon-AO is a prominent component of the Falcon infrastructure for Semantic Web applications [26]. For our datasets, Falcon-AO primarily uses partition-based block matching (PBM), which first divides each ontology into blocks that have a high degree of cohesiveness; then, mappings are discovered by matching similar blocks. The similarity between blocks is a function of the number of "anchors" (alignments with high similarity based on string comparison techniques) that they share.

3.4 Ensemble Learning

We utilize machine learning techniques to determine the weights and confidence level thresholds for each ensemble configuration, allowing for the systematic learning of rules for estimating the correctness of a correspondence based on the output of the different techniques. Our experiments were conducted with the Weka Toolkit [27], using the Weka implementation of the REPTree classifier, a fast decision tree learner which builds a tree using information gain as the splitting criterion and then prunes it using reduced error pruning. Our feature vectors comprise the individual mapping confidence scores for each technique being evaluated as well as a single meta-level feature—average matcher confidence. The inclusion of this meta-level feature is based on the findings of Eckert et al. [12] in which it was found that the most significant feature was not the confidence scores themselves, but the fraction of matchers that found a correspondence. All experiments were conducted with the default Weka classifier settings, making our experiments more easily reproducible.

Dealing with imbalanced data. Each individual matcher can generate mappings with a range of confidence scores between 0.0 and 1.0 and, unsurprisingly, a large number of incorrect mappings appear at low confidence levels. This introduces a problem during classifier training known as *class imbalance*—a large difference in the number of positive and negative instances used to train a classifier (i.e., correct vs. incorrect mappings), which may result in a classifier that is biased towards this majority class. At the extreme, this can lead to a classifier with high accuracy that has actually learned to *always* choose the majority class (i.e., that the mapping is incorrect). In order to account for this when training the classifier, we use a common resampling approach in which the training instance are sampled to provide an even distribution of correct and incorrect training instances. We achieve this by using the Resample filter of the Weka framework for sampling without replacement, and biasing towards a uniform class distribution (i.e., an even split between positive and negative instances).

4 Results

Here we describe the results of our experiments to evaluate the performance of our Word2vec-based word embedding matcher. We analyze the performance of the word embedding matcher both in isolation and by measuring its contribution when combined with one or more existing ontology matching systems, showing that this novel technique adds value that is not identified by standard ontology matching systems.

For the evaluation of each particular classifier configuration, we follow a technique meant to mimic a practical training process for each classifier within the context of scientific literature curation. More specifically, we limit the training of each classifier to a small subset of the mappings produced by the corresponding matchers. We split the training collection into *n* folds, with each fold consisting of approximately 362 instances, and train a separate classifier on each of the *n* individual folds. This is meant to simulate the process of training the classifier with a small number of manually reviewed mappings. 362 was chosen as the approximate fixed size for each fold so that the smallest training collection (YAM++ by itself; 3,628 mappings) would have 10 folds for training. Every evaluation uses the same test collection, consisting of the union of all of the potential mappings generated by each of the matching systems (including Word2vec). This allows for a more accurate comparison of the evaluation results across different classifier configurations. We report the average and standard deviation of the traditional precision, recall, and F-measure metrics across each of the *n* folds for each classifier configuration.

4.1 Word Embedding Similarity Scores

We first analyzed the similarity scores produced by the Word2vec technique, which are the cosine similarity of the vectors for each pair of concepts in the MeSH and OMIM ontologies. For comparison, we built two word embedding models for the PubTator corpus: one with the standard configuration and one providing a list of stop words, which Word2vec ignores during training. The chart in **Fig. 1** shows the raw counts of the correct and incorrect mappings for both of these models.

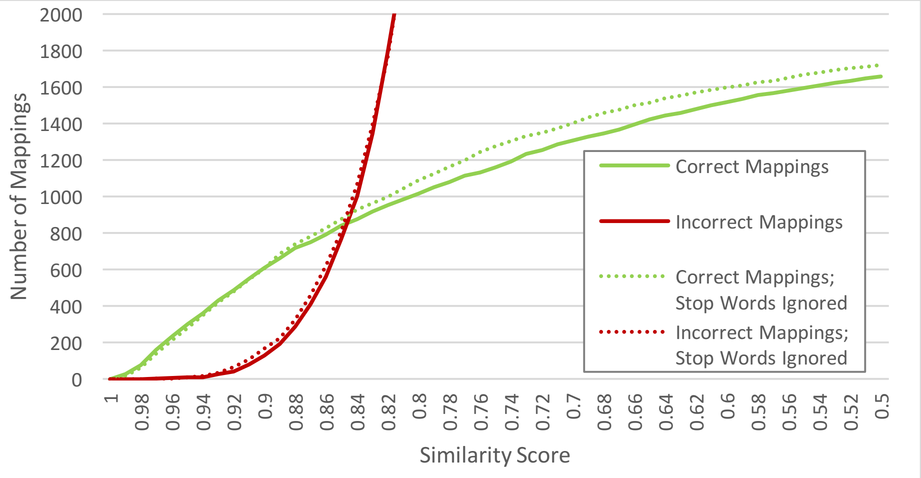

Fig. 1. The raw number of correct and incorrect mappings by Word2vec similarity score for two word embedding models, trained with and without stop words ignored.

The results from both models are very similar, with the global distribution of similarity scores (both correct and incorrect) following a normal distribution. The

Word2vec model that ignores stop words finds slightly more correct mappings when at lower values for the similarity score threshold (i.e., below 0.9). It is understandable that ignoring stop words makes little difference if the window size is sufficient, since the Word2vec model automatically accounts for the information gain afforded by specific context words (which should be near zero for stop words). In both models, the number of incorrect mappings increases drastically as the similarity score threshold decreases, with the number of correct and incorrect mappings being roughly equal with a similarity score threshold of 0.85.

For our experiments, we use similarity scores of at least 0.69. This threshold was chosen so that the number of mappings would be at least twice the size of the larger of the two ontologies (the MeSH ontology contains 11,344 concepts) because a concept in the MeSH ontology may map to more than one concept in the smaller OMIM ontology (8,064 concepts), but not the other way around. By comparison, the number of potential mappings generated by the other ontology matching systems ranges from 3,628 to 7,145. Classifiers trained from the Word2vec similarity scores alone do not perform particularly well (**Table 2**). Surprisingly, precision was high and recall was low, which is the reverse of what we had expected. For our remaining reported experimental results, we use the model with stop words ignored, representing 25,610 total instances (5.6% of which are correct mappings).

Table 2. The average and standard deviation of the F-measure and corresponding precision and recall statistics for each Word2vec word embedding model alone.

	Precision	F-measure	Recall
Word2vec	0.623 ±0.278	0.281 ±0.111	0.190 ±0.082
Word2vec; Stop Words Ignored	0.618 ±0.234	0.301 ±0.099	0.208 ±0.078

4.2 Ensemble Comparisons

For our baseline, we first look at each ontology matching system alone, using our ensemble approach to learn how to distinguish correct from incorrect mappings using only the confidence scores produced by each system (**Table 3**).

The scores for each individual matching system vary widely, which is not particularly surprising given the relatively small fixed-size folds that are used for training each classifier. In the individual configuration, GOMMA and Falcon-AO perform the best on these datasets, with F-measures of 0.590 and 0.546, respectively. Having identified the baseline values for each ontology matching system, we then included the similarity scores generated from our Word2vec word embedding matcher when training a new ensemble for each of the individual ontology matching systems (**Table 3**).

When including the Word2vec similarity scores, we see improved F-measure scores across the board and, in general, the standard deviation for each statistic decreases. The most significant gains are to the recall of the LogMap and AML systems as well as in the precision of LogMap and YAM++. Interestingly, the recall for YAM++ drops when adding Word2vec similarity scores.

Table 3. The average and standard deviation of the F-measure and corresponding precision and recall statistics for each ontology matching system alone and the difference when combined with the Word2vec word embedding matcher.

	Precision	F-measure	Recall
LogMap	0.304 ±0.270	0.260 ±0.269	0.293 ±0.345
Δ LogMap with Word2vec	**+0.243 ±0.179**	**+0.344 ±0.121**	**+0.477 ±0.226**
AML	0.471 ±0.200	0.436 ±0.165	0.530 ±0.148
Δ AML with Word2vec	**+0.131 ±0.123**	**+0.203 ±0.038**	**+0.217 ±0.159**
GOMMA	0.460 ±0.158	0.590 ±0.202	0.821 ±0.282
Δ GOMMA with Word2vec	+0.084 ±0.172	+0.038 ±0.124	+0.025 ±0.239
Falcon-AO	0.500 ±0.122	0.546 ±0.113	0.658 ±0.087
Δ Falcon-AO with Word2vec	+0.039 ±0.179	+0.025 ±0.142	+0.023 ±0.217
YAM++	0.340 ±0.242	0.331 ±0.158	0.705 ±0.288
Δ YAM++ with Word2vec	**+0.236 ±0.106**	**+0.249 ±0.083**	-0.084 ±0.145

Finally, we combined all of the ontology matching systems together to compare the results both with and without Word2vec, as shown in **Table 4**. The F-measure for the model trained using the results from all of the ontology matching systems (without Word2vec) improves over the classifiers trained on the results of each system alone (even if the improvement is only marginal, as in the case of GOMMA). The only evaluation statistics to decrease in the full ensemble configuration are the recall for GOMMA and for YAM++.

Table 4. The average and standard deviation of the F-measure and corresponding precision and recall statistics for all of the ontology matching systems combined and when combined with the Word2vec word embedding matcher.

	Precision	F-measure	Recall
ALL without Word2vec	0.593 ±0.023	0.593 ±0.061	0.683 ±0.165
Δ ALL with Word2vec	+0.053 ±0.151	+0.040 ±0.082	+0.083 ±0.213

Word2Vec contributes value beyond the traditional matching systems: including the Word2vec similarity scores when training the ensemble model boosts recall, precision, and F-measure (the standard deviation across each training fold also increases).

Interestingly, when comparing the performance of the full ensemble classifier (with Word2vec) against the individual matchers each paired with Word2vec, we see that the F-measure for both AML and GOMMA does not change significantly when including the other systems. This would seem to indicate that neither GOMMA nor AML, when combined with Word2vec, are further improved by adding any of the additional matching systems. However, note that GOMMA produces the highest recall of any combination evaluated (0.846 ±0.239), whereas the full ensemble and AML (each including Word2vec) appear to be more balanced as illustrated by their lower recall and higher precision scores.

5 Conclusions and Future Work

In this paper, we have described an ensemble learning approach that augments a collection of ontology matching systems with word embeddings generated from an annotated corpus of relevant scientific literature. We have shown that, within this ensemble approach to ontology matching, the information within word embeddings does contribute to learning an improved model for identifying correct alignments between two ontologies, beyond what state-of-the-art ontology matching systems identify—both individually and in combination. More specifically, the best overall performance (by F-measure) was found in the combination of word embedding-derived similarity scores with either the full ensemble containing *all* of the matching systems under evaluation or the individual AML and GOMMA matching system. However, each of those configurations differed in precision and recall and, therefore, the needs of any particular use case will inform the best configuration for each individual situation.

There are also several items that remain to be answered by future work as well as by our own ongoing research. First, we are currently analyzing the PubTator corpus to extract a list of *multi-word expressions*—using a novel technique for extracting salient variable-length phrases from large text corpora [28]—which we will use in a similar approach to preprocess the corpus and, prior to training the word embedding model, remove all text that is not among the top expressions in the corpus. We also see opportunities to improve upon our ensemble learning approach by providing additional meta-level features when training our ensemble model, such as binary matcher voting, global ontology features, and concept-specific lexical features used by Eckert et al. [12].

Repeating our experiments with different ontologies and/or in a different domain would help to corroborate our results. Training the relevant Word2vec model, however, requires identifying a sufficiently large domain-relevant corpus that is also annotated with concepts from those ontologies. Given a domain-relevant corpus, it may be possible to use an automated system to automatically detect and annotate concept labels in text, as was done by the DNorm disease tagger for the PubTator corpus.

There is also an opportunity to significantly reduce the processing time needed to train a Word2vec model from a given corpus. We briefly explored using Deeplearning4j's support for the Apache Spark cluster-computing framework, but we were unable to fully implement the functionality due to time limitations. With Spark, Deeplearning4j can distribute the processing and train models in parallel for individual shards of the large corpus before iteratively averaging the parameters into a central model.

Lastly, in specific regard to manual biocuration and systematic review processes, we see an opportunity to exploit additional sources of evidence beyond the resulting annotated corpus. More specifically, it may be possible to collect incremental pieces of feedback from work-centered interfaces over the course of a user's normal interaction during biocuration and annotation tasks—for example, while searching for or disambiguating specific concepts for annotating a particular text mention or reference—that can be utilized to further improve ontology matching processes.

Acknowledgments

This work is supported by the US Army Medical Research and Materiel Command under Contract No. W81XWH-13-C-0036.

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

References

1. The Uniprot Consortium (2015) UniProt: a hub for protein information. Nucleic Acids Res. 43: D204–D212. doi: 10.1093/nar/gku989
2. Sayers EW, Barrett T, Benson DA, Bolton E, Bryant SH, Canese K, Feolo M (2012) Database resources of the national center for biotechnology information. Nucleic acids research 40(D1): D13–D25. doi: 10.1093/nar/gks1189
3. Shimoyama M, De Pons J, Hayman GT, Laulederkind SJ, Liu W, Nigam R, Petri V, Smith JR, Tutaj M, Wang SJ, Worthey E, Dwinell M, Jacob H (2015) The rat genome database 2015: genomic, phenotypic and environmental variations and disease. Nucleic Acids Res. 43: D743–50. doi: 10.1093/nar/gku1026
4. Ghiasvand O, Shimoyama M (2016) Introducing a text annotation tool (OnToMate); assisting curation at rat genome database. In: Proceedings of the 7th ACM international conference on bioinformatics, computational biology, and health informatics (BCB '16). ACM, New York, pp 465-465
5. Poux S, Arighi CN, Magrane M, Bateman A, Wei CH, Lu Z, Boutet E, Bye-A-Jee H, Famiglietti ML, Roechert B (2017) On expert curation and scalability: UniProtKB/Swiss-Prot as a case study. Bioinformatics btx439. doi: https://doi.org/10.1093/bioinformatics/btx439
6. Rodriguez-Esteban R (2015) Biocuration with insufficient resources and fixed timelines. Database: The Journal of Biological Databases and Curation 2015; 2015: bav116. doi:10.1093/database/bav116
7. Marshall C, Brereton P (2015) Systematic review toolbox: A catalogue of tools to support systematic reviews. In: Proceedings of the 19th international conference on evaluation and assessment in software engineering. ACM, New York, p 23
8. Choong MK, Galgani F, Dunn AG, Tsafnat G (2014) Automatic evidence retrieval for systematic reviews. J Med Internet Res 2014;16(10): e223. doi: 10.2196/jmir.3369
9. Wallace BC, Kuiper J, Sharma A, Zhu MB, Marshall IJ (2016) Extracting PICO sentences from clinical trial reports using supervised distant supervision. Journal of Machine Learning Research, 17(132): 1–25
10. Basu T, Kumar S, Kalyan A, Jayaswal P, Goyal P, Pettifer S, Jonnalagadda S (2016) Systematic reviews by automatically building information extraction training corpora. arXiv preprint arXiv:1606.06424.
11. Shvaiko P, Euzenat J (2013) Ontology matching: state of the art and future challenges. IEEE transactions on knowledge and data engineering 25(1): pp 158–176. doi: 10.1109/TKDE.2011.253
12. Eckert K, Meilicke C, Stuckenschmidt H (2009) Improving ontology matching using meta-level learning. In: Aroyo L, et al. (eds). LNCS, volume 5554. Springer International Publishing, Cham, Switzerland, pp 158-172. doi: https://doi.org/10.1007/978-3-642-02121-3
13. Gal A (2011) Uncertain schema matching. Synthesis Lectures on Data Management 3(1):1–97

14. Faria D, Pesquita C, Santos E, Cruz IF, Couto FM (2014) Automatic background knowledge selection for matching biomedical ontologies. PLoS ONE 9(11): e111226. doi: https://doi.org/10.1371/journal.pone.0111226
15. Zhang Y, Wang X, Lai S, He S, Liu K, Zhao J, Lv X (2014) Ontology matching with word embeddings. In: Maosong S, Liu Y, Zhao J (eds) Chinese computational linguistics and natural language processing based on naturally annotated big data. Springer International Publishing, Cham, Switzerland, pp 34–45. doi: https://doi.org/10.1007/978-3-319-12277-9
16. Lin F, Sandkuhl K (2008) A survey of exploiting wordnet in ontology matching. In: Bramer M (ed) Artificial intelligence in theory and practice, vol 2. Springer US, New York, pp 341–350. doi: 10.1007/978-0-387-34747-9
17. Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Stroudsburg, PA, pp 384–394
18. Li Y, Yang T (2018) Word embedding for understanding natural language: survey. In: Srinivasan S. (ed) Guide to big data applications. Studies in big data, vol 26. Springer International Publishing, Cham, Switzerland, pp 83–104. doi: 10.1007/978-3-319-53817-4
19. Davis AP, Wiegers TC, Rosenstein MC, Mattingly CJ (2012) MEDIC: a practical disease vocabulary used at the comparative toxicogenomics database. Database: The Journal of Biological Databases and Curation 2012; 2012: bar065. doi: 10.1093/database/bar065
20. Leaman R, Islamaj Doğan R, Lu, Z (2013). DNorm: disease name normalization with pairwise learning to rank. Bioinformatics 29(22): 2909–2917
21. Deeplearning4j Development Team (2016) Deeplearning4j: Open-source distributed deep learning for the JVM. https://deeplearning4j.org/about. Accessed 27 July 2017
22. Jiménez-Ruiz E, Cuenca Grau B (2011) LogMap: Logic-based and scalable ontology matching. In: Aroyo L et al. (eds) The semantic web – ISWC 2011. ISWC 2011. Lecture Notes in Computer Science, vol 7031. Springer, Berlin, Heidelberg, pp 273–288. doi: https://doi.org/10.1007/978-3-642-25073-6_18
23. Faria D, Pesquita C, Santos E, Palmonari M, Cruz IF, Couto FM (2013) The AgreementMakerLight ontology matching system. In: Meersman R et al. (eds) On the move to meaningful internet systems: OTM 2013 Conferences. OTM 2013. Lecture Notes in Computer Science, vol 8185. Springer, Berlin, Heidelberg, pp. 527–541. doi: https://doi.org/10.1007/978-3-642-41030-7_38
24. Kirsten T, Gross A, Hartung M, Rahm E (2011) GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. Journal of Biomedical Semantics 2(1): 6. doi: 10.1186/2041-1480-2-6
25. Duyhoa N, Bellahsene Z (2014) Overview of YAM++-(not) Yet Another Matcher for ontology alignment task. Dissertation, LIRMM
26. Hu W, Qu Y (2008) Falcon-AO: A practical ontology matching system. Web Semantics: Science, Services and Agents on the World Wide Web 6(3): 237–239. doi: 10.1016/j.websem.2008.02.006
27. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I (2009) The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1): 10–18. doi: 10.1145/1656274.1656278
28. Shang J, Liu J, Jiang M, Ren X, Voss CR, Han J (2017) Automated phrase mining from massive text corpora. *arXiv preprint arXiv:1702.04457*