Ontology Alignment in the Biomedical Domain Using Entity Definitions and Context

Lucy Lu Wang[†], Chandra Bhagavatula, Mark Neumann, Kyle Lo, Chris Wilhelm, and Waleed Ammar

Allen Institute for Artificial Intelligence
[†]Department of Biomedical Informatics and Medical Education, University of Washington
Seattle, Washington, USA
lucylw@uw.edu

Abstract

Ontology alignment is the task of identifying semantically equivalent entities from two given ontologies. Different ontologies have different representations of the same entity, resulting in a need to de-duplicate entities when merging ontologies. We propose a method for enriching entities in an ontology with external definition and context information, and use this additional information for ontology alignment. We develop a neural architecture capable of encoding the additional information when available, and show that the addition of external data results in an F1-score of 0.69 on the Ontology Alignment Evaluation Initiative (OAEI) largebio SNOMED-NCI subtask, comparable with the entity-level matchers in a SOTA system.

1 Introduction

Ontologies are used to ground lexical items in various NLP tasks including entity linking, question answering, semantic parsing and information retrieval.[1] In biomedicine, an abundance of ontologies (e.g., MeSH, Gene Ontology) has been developed for different purposes. Each ontology describes a large number of concepts in healthcare, public health or biology, enabling the use of ontology-based NLP methods in biomedical applications. However, since these ontologies are typically curated independently by different groups, many important concepts are represented inconsistently across ontologies (e.g., "Myoclonic Epilepsies, Progressive" in MeSH is a broader concept that includes "Dentatorubral-pallidoluysian atrophy" from OMIM).

This poses a challenge for bioNLP applications where multiple ontologies are needed for grounding, but each concept must be represented by only one entity. For instance, in www.semanticscholar.org, scientific publications related to carpal tunnel syndrome are linked to one of multiple entities derived from UMLS terminologies representing the same concept,[2] making it hard to find all relevant papers on this topic. To address this challenge, we need to automatically map semantically equivalent entities from one ontology to another. This task is referred to as ontology alignment or ontology matching.

Several methods have been applied to ontology alignment, including rule-based and statistical matchers. Existing matchers rely on entity features such as names, synonyms, as well as relationships to other entities (Shvaiko and Euzenat, 2013; Otero-Cerdeira et al., 2015). However, it is unclear how to leverage the natural language text associated with entities to improve predictions. We address this limitation by incorporating two types of natural language information (definitions and textual contexts) in a supervised learning framework for ontology alignment. Since the definition and textual contexts of an entity often provide complementary information about the entity's meaning, we hypothesize that incorporating them will improve model predictions. We also discuss how to automatically derive labeled data for training the model by leveraging existing resources. In particular, we make the following contributions:

- We propose a novel neural architecture for ontology alignment and show how to effectively

[1] Ontological resources include ontologies, knowledgebases, terminologies, and controlled vocabularies. In the rest of this paper, we refer to all of these with the term 'ontology' for consistency.

[2] See https://www.semanticscholar.org/topic/Carpal-tunnel-syndrome/248228 and https://www.semanticscholar.org/topic/Carpal-Tunnel-Syndrome/3076

Proceedings of the BioNLP 2018 workshop, pages 47–55
Melbourne, Australia, July 19, 2018.

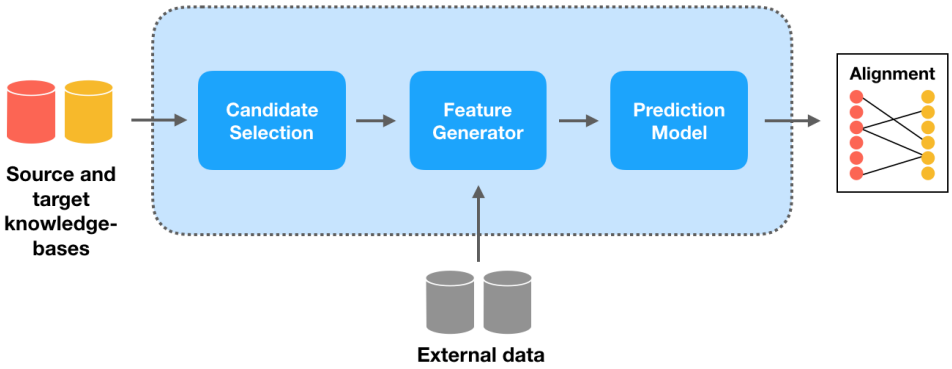

Figure 1: OntoEmma consists of three modules: a) candidate selection (see §2.2 for details), b) feature generation (see §2.2 for details), and c) prediction (see §2.3 for deatils). OntoEmma accepts two ontologies (a source and a target) as inputs, and outputs a list of alignments between their entities. When using a neural network, the feature generation and prediction model are combined together in the network.

integrate natural language inputs such as definitions and contexts in this architecture (see §2 for details).[3]

- We use the UMLS Metathesaurus to extract large amounts of labeled data for supervised training of ontology alignment models (see §3.1). We release our data set to help future research in ontology alignment.[3]
- We use external resources such as Wikipedia and scientific articles to find entity definitions and contexts (see §3.2 for details).

2 OntoEmma

In this section, we describe OntoEmma, our proposed method for ontology matching, which consists of three stages: candidate selection, feature generation and prediction (see Fig. 1 for an overview).

2.1 Problem definition and notation

We start by defining the ontology matching problem: Given a source ontology O^s and a target ontology O^t, each consisting of a set of entities, find all semantically equivalent entity pairs, i.e., $\{(\mathbf{e}^s, \mathbf{e}^t) \in O^s \times O^t : \mathbf{e}^s \equiv \mathbf{e}^t\}$, where $\equiv$ indicates semantic equivalence. For consistency, we preprocess entities from different ontologies to have the same set of attributes: a canonical name ($\mathbf{e}_{\text{name}}$), a list of aliases ($\mathbf{e}_{\text{aliases}}$), a textual definition ($\mathbf{e}_{\text{def}}$), and a list of usage contexts ($\mathbf{e}_{\text{contexts}}$).[4]

2.2 Candidate selection and feature generation

Many ontologies are large, which makes it computationally expensive to consider all possible pairs of source and target entities for alignment. For example, the number of all possible entity pairs in our training ontologies is on the order of 10^{11}. In order to reduce the number of candidates, we use an inexpensive low-precision, high-recall candidate selection method using the inverse document frequency (*idf*) of word tokens appearing in entity names and definitions. For each source entity, we first retrieve all target entities that share a token with the source entity. Given the set of shared word tokens $\mathbf{w_{s+t}}$ between a source and target entity, we sum the *idf* of each token over the set, yielding $idf_{total} = \sum_{i \epsilon \mathbf{w_{s+t}}} idf(i)$. Tokens with higher *idf* values appear less frequently overall in the ontology and presumably contribute more to the meaning of a specific entity. We compute the *idf* sum for each target entity and output the $K = 50$ target entities with the highest value for each source entity, yielding $|O^s| \times K$ candidate pairs.

For each candidate pair $(\mathbf{e}^s, \mathbf{e}^t)$, we precompute a set of 32 features commonly used in the ontology matching literature including the token Jaccard distance, stemmed token Jaccard distance, character n-gram Jaccard distance, root word equivalence, and other boolean and probability values

[3]Implementation and data available at https://www.github.com/allenai/ontoemma/

[4]Some of these attributes may be missing or have low coverage. See §3.2 for coverage details.

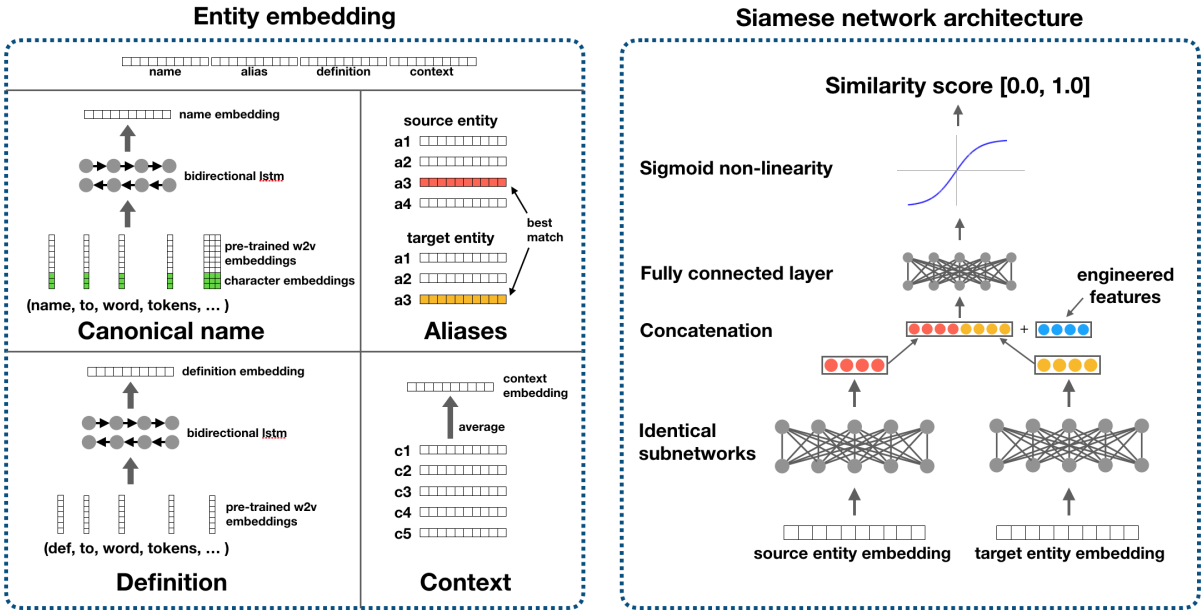

Figure 2: Siamese network architecture for computing entity embeddings for each source and target entity in a candidate entity pair.

over the entity name, aliases, and definition.[5]

2.3 Prediction

Given a candidate pair $(\mathbf{e}^s, \mathbf{e}^t)$ and the precomputed features $\mathbf{f}(\mathbf{e}^s, \mathbf{e}^t)$, we train a model to predict the probability that the two entities are semantically equivalent. Figure 2 illustrates the architecture of our neural model for estimating this probability which resembles a siamese network (Bromley et al., 1993). At a high level, we first encode each of the source and target entities, then concatenate their representations and feed it into a multilayer perceptron ending with a sigmoid function for estimating the probability of a match. Next, we describe this architecture in more detail.

Entity embedding. As shown in Fig. 2 (*left*), we encode the attributes of each entity as follows:

- A canonical name $\mathbf{e}_{\text{name}}$ is a sequence of tokens, each encoded using pretrained `word2vec` embeddings concatenated with a character-level convolutional neural network (CNN). The token vectors feed into a bi-directional long short-term memory network (LSTM) and the hidden layers at both ends of the bi-directional LSTM are concatenated and used as the name vector $\mathbf{v}_{\text{name}}$.
- Each alias in $\mathbf{e}_{\text{aliases}}$ is independently embedded using the same encoder used for canonical names (with shared parameters), yielding a set of alias vectors $\mathbf{v}_{\text{alias}-i}$ for $i = 1, \ldots, |\mathbf{e}_{\text{aliases}}|$.
- An entity definition $\mathbf{e}_{\text{def}}$ is a sequence of tokens, each encoded using pretrained embeddings then fed into a bi-directional LSTM. The definition vector $\mathbf{v}_{\text{def}}$ is the concatenation of the final hidden states in the forward and backward LSTMs.
- Each context in $\mathbf{e}_{\text{contexts}}$ is independently embedded using the same encoder used for definitions (with shared parameters), then averaged yielding the context vector $\mathbf{v}_{\text{contexts}}$.

The name, alias, definition, and context vectors are appended together to create the entity embedding, e.g., the source entity embedding $\mathbf{e}^s$ is: $\mathbf{v}^s = [\mathbf{v}^s_{\text{name}}; \mathbf{v}^s_{\text{alias}-i^*}; \mathbf{v}^s_{\text{def}}; \mathbf{v}^s_{\text{contexts}}]$. In order to find representative aliases for a given pair of entities, we pick the source and target aliases with the smallest Euclidean distance, i.e., $i^*, j^* = \arg\min_{i,j} \|\mathbf{v}^s_{\text{alias}-i} - \mathbf{v}^t_{\text{alias}-j}\|_2$

Siamese network. After the source and target entity embeddings are computed, they are fed into two subnetworks with shared parameters followed by a parameterized function for estimating similarity. Each subnetwork is a two layer feedforward

[5] Even though neural models may obviate the need for feature engineering, feeding highly discriminative features into the neural model improves the inductive bias of the model and reduces the amount of labeled data needed for training.

network with ReLU non-linearities and dropout (Srivastava et al., 2014). The outputs of the two subnetworks are then concatenated together with the engineered features and fed into another feed-forward network with a ReLU layer followed by a sigmoid output layer. We train the model to minimize the binary cross entropy loss for gold labels.

To summarize, the network estimates the probability of equivalence between $\mathbf{e}^s$ and $\mathbf{e}^t$ as follows:

$$\mathbf{h}^s = \textsc{ReLU}(\textsc{ReLU}(\mathbf{v}^s; \theta_1); \theta_2)$$
$$\mathbf{h}^t = \textsc{ReLU}(\textsc{ReLU}(\mathbf{v}^t; \theta_1); \theta_2)$$
$$P(\mathbf{e}^s \equiv \mathbf{e}^t) = \textsc{Sigmoid}(\textsc{ReLU}([\mathbf{h}^s; \mathbf{h}^t]; \theta_3); \theta_4)$$

3 Deriving and enriching labeled data

In this section, we discuss how to derive a large amount of labeled data for training, and how to augment entity attributes with definitions and contexts from external resources.

3.1 Deriving training data from UMLS

The Unified Medical Language System (UMLS) Metathesaurus, which integrates more than 150 source ontologies, illustrates the breadth of coverage of biomedical ontologies (Bodenreider, 2004). Also exemplified by the UMLS Metathesaurus is the high degree of overlap between the content of some of these ontological resources, whose terms have been (semi-)manually aligned. Significant time and effort has gone into cross-referencing semantically equivalent entities across the ontologies, and new terms and alignments continue to be added as the field develops. These manual alignments are high quality, but considered to be incomplete (Morrey et al., 2011; Mougin and Grabar, 2014).

To enable supervised learning for our models, training data was derived from the UMLS Metathesaurus. By exposing our models to labeled data from the diverse subdomains covered in the UMLS Metathesaurus, we hope to learn a variety of patterns indicating equivalence between a pair of entities which can generalize to new ontologies not included in the training data.

We identified the following set of ontologies within UMLS to use as the source of our labeled data, such that they cover a variety of domains without overlapping with the test ontologies used for evaluation in the OAEI: Current Procedural Terminology (CPT), Gene Ontology (GO), Hugo Nomenclature (HGNC), Human Phenotype Ontology (HPO), Medical Subject Headings (MeSH), Online Mendelian Inheritance in Man (OMIM), and RxNorm.

Our labeled data take the form $(\mathbf{e}^s, \mathbf{e}^t, l \in \{0, 1\})$, where $l = 1$ indicates positive examples where $\mathbf{e}^s \equiv \mathbf{e}^t$. For each pair of ontologies, we first derive all the positive mappings from UMLS. We retain the positive mappings for which there are no name equivalences. Then, for each positive example $(\mathbf{e}^s, \mathbf{e}^t_+, 1)$, we sample negative mappings $(\mathbf{e}^s, \mathbf{e}^t_-, 0)$ from the other entities in the target ontology. One "easy" negative and one "hard" negative are selected for each positive alignment, where easy negatives consist of entities with little overlap in lexical features while hard negatives have high overlap. Easy negatives are obtained by randomly sampling entities from the target ontology, for each source entity. Hard negatives are obtained using the same candidate selection method described in §2. In both easy and hard examples, we exclude all target entities which appear in a positive example.[6]

Over all seven ontologies, 50,523 positive alignments were extracted from UMLS. Figure 3 reports the number of positive alignments extracted from each ontology pair. For these positives, 98,948 hard and easy negatives alignments were selected. These positive and negative labeled examples were pooled and randomly split into a 64% training set, a 16% development set, and a 20% test set.

	CPT	GO	HGNC	HPO	MeSH	OMIM	RxNorm
CPT		3	0	0	449	4	19
GO			0	10	741	48	0
HGNC				0	29	15437	0
HPO					2245	9891	0
MeSH						12683	8964
OMIM							0
RxNorm							

Figure 3: Number of positive alignments extracted from each pair of ontologies from UMLS.

3.2 Deriving definitions and mention contexts

Many ontologies do not provide entity definitions (Table 1). In fact, only a few (GO, HPO, MeSH) of the ontologies we included have any definitions at all.

[6] Although the negative examples we collect may be noisy due to the incompleteness of manual alignments in UMLS, this noise is also present in widely adopted evaluation of knowledge base completion problems and relation extraction with distant supervision (e.g., Li et al., 2016; Mintz et al., 2009).

Table 1: Entities with definitions and contexts for each of the training ontologies.

Ont.	# of entities	% w/ def.	% w/ con.
CPT	13,786	0.0	97.9
GO	44,684	100.0	30.5
HGNC	39,816	0.0	0.8
HPO	11,939	72.5	17.9
MeSH	268,162	10.5	35.1
OMIM	98,515	0.0	2.8
RxNorm	205,858	0.0	5.1
Total	682,760	11.9	20.1

We can turn to external sources of entity definitions in such cases. Many biomedical and healthcare concepts are represented in Wikipedia, a general purpose crowd-sourced encyclopedia. The Wikipedia API can be used to search for and extract article content. The first paragraph in each Wikipedia article offers a description of the concept, and can be used as a substitute for a definition. For each entity in the labeled dataset described in the previous section, we query Wikipedia using the entity's canonical name. The first sentence from the top Wikipedia article match is extracted and used to populate the attribute $\mathbf{e}_{\text{def}}$ when undefined in the ontology. For example, a query for OMIM:125370, "Dentatorubral-pallidoluysian atrophy," yields the following summary sentence from Wikipedia: *"Dentatorubral-pallidoluysian atrophy (DRPLA) is an autosomal dominant spinocerebellar degeneration caused by an expansion of a CAG repeat encoding a polyglutamine tract in the atrophin-1 protein."* Based on a human-annotated sample, the accuracy of our externally-derived definitions is 75.5%, based on a random sample of 200 definitions and two annotators with Cohen's kappa coefficient of $\kappa = 0.88$.[7]

Usage contexts are derived from scientific papers in Medline, leveraging entity annotations available via the Semantic Scholar project (Ammar et al., 2018). In order to obtain the annotations, an entity linking model was used to find mentions of UMLS entities in the abstracts of Medline papers. The sentences in which a UMLS entity were mentioned are added to the $\mathbf{e}_{contexts}$ attribute of that entity. For UMLS entity C0751781, "Dentatorubral-Pallidoluysian Atrophy," an example context: *"Dentatorubral-pallidoluysian atrophy (DRPLA) is an autosomal dominant neurodegenerative disease clinically characterized by the presence of cerebellar ataxia in combination with variable neurological symptoms,"* is extracted from Yoon et al (2012) (Yoon et al., 2012). This context sentence was scored highly by the linking model, and provides additional information about this entity, for example, its acronym (*DRPLA*), the type of disease (*autosomal dominant neurodegenerative*), and some of its symptoms (*cerebellar ataxia*). Because there are often numerous linked contexts for each entity, we sample up to 20 contexts per entity when available. The number of entities with context in our labeled data is given in Table 1. The accuracy of usage contexts extracted using this approach is 92.5%, based on human evaluation of 200 contexts with Cohen's kappa coefficient of $\kappa = 1$.[7]

4 Experiments

In this section, we experiment with several variants of OntoEmma: In the first variant (OntoEmma:NN), we only encode native attributes obtained from the source and target ontologies: canonical name, aliases, and native definitions. In the second variant (OntoEmma:NN+f), we also add the manually engineered features as described in §2.2. In the third variant (OntoEmma:NN+f+w), we incorporate external definitions from Wikipedia, as discussed in §3.2. In the fourth variant (OntoEmma:NN+f+w+c), we also encode the usage contexts we derived from Medline, also discussed in §3.2.

Data. We use the training section of the UMLS-derived labeled data to train the model and use the development section to tune the model hyperparameters. For evaluation, we use the test portion of our UMLS-derived data as well as the OAEI largebio subtrack SNOMED-NCI task, the largest task in OAEI largebio. The UMLS test set includes 29,859 positive and negative mappings. The OAEI reference alignments included 17,210 equivalent mappings and 1,623 uncertain mappings between the SNOMED and NCI ontologies.

Baselines. Our main baseline is a logistic regression model (OntoEmma:LR) using the same engineered features described in §2.2. To illustrate how our proposed method performs compared to previous work on ontology matching, we compare

[7] Annotations are available at https://github.com/allenai/ontoemma#human-annotations

Table 2: Model performance on UMLS test dataset

Model	Prec.	Recall	F1
OntoEmma:LR	0.98	0.92	0.95
OntoEmma:NN	0.87	0.85	0.86
OntoEmma:NN+f	0.93	0.96	0.95
OntoEmma:NN+f+w	0.93	0.97	0.95
OntoEmma:NN+f+w+c	0.94	0.97	0.96

Table 3: Model performance on OAEI largebio SNOMED-NCI task

Model	Prec.	Recall	F1
AML:entity	0.81	0.62	0.70
OntoEmma:LR	0.75	0.56	0.65
OntoEmma:NN+f+w+c	0.80	0.61	0.69

to AgreementMakerLight (AML) which has consistently been a top performer in the OAEI subtasks related to biomedicine (Faria et al., 2013). For a fair comparison to OntoEmma, we only use the entity-level matchers in AML; i.e., relation and structural matchers in AML are turned off.[8]

Implementation and configuration details. We provide an open source, modular, Python implementation of OntoEmma where different candidate selectors, feature generators, and prediction modules can be swapped in and out with ease.[3] We implement the neural model using `PyTorch` and `AllenNLP`[9] libraries, and implement the logistic regression model using `scikit-learn`. Our 100-dimensional pre-trained embeddings are learned using the default settings of `word2vec` based on the Medline corpus. The character-level CNN encoder uses 50 filters of size 4 and 5, and outputs a token embedding of size 100 with dropout probability of 0.2. The LSTMs have output size 100, and have dropout probability of 0.2.

Results. The performance of the models is reported in terms of precision, recall and F1 score on the held-out UMLS test set and the OAEI largebio SNOMED-NCI task in Tables 2 and 3, respectively.

Table 2 illustrates how different variants of OntoEmma perform on the held-out UMLS test set. We note that the bare-bones neural network model (OntoEmma:NN) does not match the performance of the baseline logistic regression model (OntoEmma:LR), suggesting that the representations learned by the neural network are not sufficient. Indeed, adding the engineered features to the neural model in (OntoEmma:NN+f) provides substantial improvements, matching the F1 score of the baseline model. Adding definitions and usage context in (OntoEmma:NN+f+w+c) further improves the F1 score by one absolute point, outperforming the logistic regression model.

While the UMLS-based test set in Table 2 represents the realistic scenario of aligning new entities in partially-aligned ontologies, we also wanted to evaluate the performance of our method on the more challenging scenario where no labeled data is available in the source and target ontologies. This is more challenging because the patterns learned from ontologies used in training may not transfer to the test ontologies. Table 3 illustrates how our method performs in this scenario using SNOMED-NCI as test ontologies. For matching of the SNOMED and NCI ontologies, we enriched the entities first using Wikipedia queries. At test time, we also identified and aligned pairs of entities with exact string matches, using the OntoEmma matcher only for those entities without an exact string match. Unsurprisingly, the performance of OntoEmma on unseen ontologies (in Table 3) is much lower than its performance on seen ontologies (in Table 2). With unseen ontologies, we gain a large F1 improvement of 4 absolute points by using the fully-featured neural model (OntoEmma:NN+f+w+c) instead of the logistic regression variant (OntoEmma:LR), suggesting that the neural model may transfer better to different domains. We note, however, that the OntoEmma:NN+f+w+c matcher performs slightly worse than the AML entity matcher. This is to be expected since AML incorporates many matchers which we did not implement in our model, e.g., using background knowledge, acronyms, and other features.

5 Discussion

Through building and training a logistic regression model and several neural network models, we evaluated the possibility of training a supervised machine learning model for ontology alignment based on existing alignment data, and evalu-

[8] The performance of the full AML system on the SNOMED-NCI subtask for OAEI 2017 is: precision: 0.90, recall: 0.67, F1: 0.77.

[9] https://allennlp.org/

ated the efficacy of including definitions and usage context to improve entity matching. For the first question, we saw some success with both the logistic regression and neural network models. The logistic regression model performed better than the simple neural network model without engineered features. Hand-engineered features encode human knowledge, and are less noisy than features trained from a neural network. The NN model required more training data to learn the same sparse information encoded by pre-defined features.

To bolster performance, hand-engineered features and extensive querying of third-party resources were used to increase knowledge about each entity. Definitions and usage contexts had rarely been used by previous ontology matchers, and we sought to exploit the value of these additional pieces of information. Definitions especially, often offer information about an entity's relations and attributes, which may not be explicitly defined in the ontology. The ontologies used for training contained inconsistent information – some had definitions for all entities, some none; some were well-represented in our context linking model, some were not. To take advantage of such information, therefore, we had to turn to external sources of definitions and contexts, which are understandably more noisy than information provided in the ontology itself.

Using Wikipedia and the Medline corpus, we derived definitions and contexts for many of the entities in the UMLS training corpus. Adding definitions improved the performance of our neural network model. However, high quality definitions native to each terminology would likely have improved results further, since we could not ensure that externally derived definitions were always relevant to the entity of interest.

Limitations. Our ontology matcher did not implement any structural matchers, and did not take advantage of relationship data where it existed. In ontologies with well-defined hierarchy or relationships, the structural component provides orthogonal and extremely relevant information for matching. By choosing to focus on entity alignment, we were unable to be competitive on global ontology matching.

Of all the entities in our UMLS training, development, and test datasets, only 11.9% of entities had definitions from their source ontology (Table 1). Similarly, we were only able to derive contexts for 20.1% of the training entities from the Semantic Scholar entity linking model (Table 1). We were hoping for better coverage of the overall dataset. We were, however, able to use Wikipedia to increase the overall definition coverage of the entities in our data set to 82.1%.

Although Wikipedia is a dense resource containing curated articles on many concepts, it is by no means exhaustive. Many of the entities in our training and test data set did not correspond directly to entities in Wikipedia. We also could not review each query to ensure a correct match between the Wikipedia article and the entity. The data is therefore noisy and can introduce error in some cases. Although the overall performance improved upon querying Wikipedia for additional definitions, we believe that dedicated definitions from the source terminologies would perform better where available.

Future work. We are exploring other ways to derive high-quality definitions from external resources, for example, by deriving definitions from synonymous entities in other ontologies, or by generating textual definitions using the logical definitions given in an ontology. Similarly, we can incorporate usage context from other sources. For example, the Semantic MEDLINE Database (SemMedDB) is a database of semantic relationship predictions from PubMed articles (Kilicoglu et al., 2012). The entity-relation triples in this database can be used to retrieve PubMed article context mapped to UMLS terms.

Continuing on, we aim to develop a more flexible ontology matching system that takes into account all of the information available about an entity. Flexible entity embeddings would represent critical information for proper entity alignment, while accounting for a variety of data types, such as list-like and graph-like data. We would also like to incorporate ontology structure and relations in matching. Hierarchical structure is provided by most biomedical terminologies, and provides essential information for a matching system. One possibility is ensembling OntoEmma with other matcher systems that incorporate or focus on using structural features in matching.

Related work The OAEI has been driving ontology matching research in the biomedical domain since 2005. It provides evaluation data supporting several tracks such as the anatomy,

largebio, and more recently introduced phenotype tracks (Faria et al., 2016). Participating matchers implement a variety of matching techniques including rule-based and statistical methods (Faria et al., 2016; Gross et al., 2016; Otero-Cerdeira et al., 2015; Shvaiko and Euzenat, 2013). Features used by matchers can be element-level (extracted from each individual entity), or structure-level (based on the topology of the ontology and its relationships). Content features can be based on terminology (i.e., names of entities), structure (i.e., how entities are connected), annotations (i.e., annotations made to entities), or reasoning output. Some features can also be derived from external sources, such as cross-references to other ontologies, or cross-annotations in other datasets, such as term coincidence in publications, or co-annotation of experiments with terms from different ontologies (Husein et al., 2016).

Notable general purpose matchers that have excelled in biomedical domain matching tasks include AgreementMakerLight (AML), YAM++, and LogMap. AML has consistently been a top performer in the OAEI subtasks related to biomedicine. It uses a combination of different matchers, such as the lexical matcher (looking for complete string matches between the names of entities), mediating matcher (performing the function of the lexical matcher through a third background ontology), word-based string similarity matcher (matching entities with minimal string edit distances), and others. AML then combines these various similarity scores to generate a global alignment between the two input ontologies (Faria et al., 2013). YAM++, another successful matcher, implemented a decision tree learning model over many string similarity metrics, but leaves the challenges of finding suitable training data to the user, defaulting to information retrieval-based similarity metrics for its decision-making when no training data is provided (Ngo and Bellahsene, 2016). LogMap is a matcher specifically designed to efficiently align large ontologies, generating logical output alignments. The matcher uses high-probability matches as anchors from which to deploy its lexical and structural matchers (Jiménez-Ruiz and Cuenca Grau, 2011).

Our system uses neural networks to learn entity representations and features for matching. Several published works discuss using neural networks to learn weights over similarity metrics pre-defined by the user or developer of the matching system (Djeddi and Khadir, 2013; Peng, 2010; Huang et al., 2008; Hariri et al., 2006). These systems do not use neural networks to generate and learn the features most appropriate for entity matching. Qiu et al. (2017) proposes and tests an auto-encoder network for unsupervised entity representation learning over a bag of words vector that treats all descriptive elements of each entity (its name, definitions etc.) equally. We are interested in investigating how these various descriptive elements contribute to entity matching, how sparsity of specific descriptive fields can be offset by deriving information from external resources, and also whether we can use domain-specific training data to optimize a model for the biomedical domain.

Conclusion In this paper, we propose using natural language text associated with entities to improve ontology alignment. We describe a novel neural architecture for ontology alignment which can encode a variety of information, and derive large amounts of labeled data for training the model. To address the limited coverage of definitions and usage contexts describing entities, we turn to external resources to supplement the available information about entities in the test ontologies. Our empirical results illustrate that externally-derived definitions and contexts can effectively be used to improve the performance of ontology matching systems.

6 Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. We also thank John Gennari, Oren Etzioni, Joanna Power as well as the rest of the Semantic Scholar team at the Allen Institute for Artificial Intelligence for helpful comments and insights.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *NAACL (industry track)*.

Olivier Bodenreider. 2004. The Unified

Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res* 32(Database issue):D267–D270. https://doi.org/10.1093/nar/gkh061.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a siamese time delay neural network. In *NIPS*.

Warith Eddine Djeddi and Mohamed Tarek Khadir. 2013. Ontology Alignment Using Artificial Neural Network for Large-scale Ontologies. *Int. J. Metadata Semant. Ontologies* 8(1):75–92. https://doi.org/10.1504/IJMSO.2013.054180.

Daniel Faria, Catia Pesquita, Booma S. Balasubramani, Catarina Martins, João Cardoso, Hugo Curado, Francisco M. Couto, and Isabel F. Cruz. 2016. OAEI 2016 results of AML. volume 1766, pages 138–145.

Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto. 2013. The AgreementMakerLight Ontology Matching System. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, pages 527–541. https://doi.org/10.1007/978-3-642-41030-7_38.

Anika Gross, Cedric Pruski, and Erhard Rahm. 2016. Evolution of biomedical ontologies and mappings: Overview of recent approaches. *Comput Struct Biotechnol J* 14:333–340. https://doi.org/10.1016/j.csbj.2016.08.002.

Babak Bagheri Hariri, Hassan Abolhassani, and Hassan Sayyadi. 2006. A Neural-Networks-Based Approach for Ontology Alignment. Japan Society for Fuzzy Theory and Intelligent Informatics, pages 1248–1252. https://doi.org/10.14864/softscis.2006.0.1248.0.

Jingshan Huang, Jiangbo Dang, Michael N. Huhns, and W. Jim Zheng. 2008. Use artificial neural network to align biological ontologies. *BMC Genomics* 9 Suppl 2:S16. https://doi.org/10.1186/1471-2164-9-S2-S16.

Inne Gartina Husein, Saiful Akbar, Benhard Sitohang, and Fazat Nur Azizah. 2016. Review of ontology matching with background knowledge. In *2016 International Conference on Data and Software Engineering (ICoDSE)*. pages 1–6. https://doi.org/10.1109/ICODSE.2016.7936159.

Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. 2011. Logmap: Logic-based and scalable ontology matching. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 273–288.

Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, Graciela Rosemblat, and Thomas C. Rindflesch. 2012. Semmeddb: a pubmed-scale repository of biomedical semantic predications. *Bioinformatics* 28:3158–60. https://doi.org/10.1093/bioinformatics/bts591.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

Charles Paul Morrey, Ling Chen, Michael Halper, and Yehoshua Perl. 2011. Resolution of redundant semantic type assignments for organic chemicals in the UMLS. *Artif Intell Med* 52(3):141–151. https://doi.org/10.1016/j.artmed.2011.05.003.

Fleur Mougin and Natalia Grabar. 2014. Auditing the multiply-related concepts within the UMLS. *J Am Med Inform Assoc* 21(e2):e185–193. https://doi.org/10.1136/amiajnl-2013-002227.

DuyHoa Ngo and Zohra Bellahsene. 2016. Overview of YAM++(not) Yet Another Matcher for ontology alignment task. *Web Semantics: Science, Services and Agents on the World Wide Web* 41:30–49. https://doi.org/10.1016/j.websem.2016.09.002.

Lorena Otero-Cerdeira, Francisco J. Rodriguez-Martinez, and Alma Gomez-Rodriguez. 2015. Ontology matching: A literature review. *Expert Systems with Applications* 42(2):949–971. https://doi.org/10.1016/j.eswa.2014.08.032.

Yefei Peng. 2010. Ontology Mapping Neural Network: An Approach to Learning and Inferring Correspondences Among Ontologies.

Lirong Qiu, Jia Yu, Qiumei Pu, and Chuncheng Xiang. 2017. Knowledge entity learning and representation for ontology matching based on deep neural networks. *Cluster Comput* 20(2):969–977. https://doi.org/10.1007/s10586-017-0844-1.

Pavel Shvaiko and Jérôme Euzenat. 2013. Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering* 25(1):158–176. https://doi.org/10.1109/TKDE.2011.253.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Won Tae Yoon, Jinyoung Youn, and Jin Whan Cho. 2012. Is cerebral white matter involvement helpful in the diagnosis of dentatorubral-pallidoluysian atrophy? *J Neurol* 259:1694–7. https://doi.org/10.1007/s00415-011-6401-6.