Kolyvakis *et al. Journal of Biomedical Semantics* (2018) 9:21
https://doi.org/10.1186/s13326-018-0187-8

Journal of
Biomedical Semantics

RESEARCH

Open Access

Biomedical ontology alignment: an approach based on representation learning

Prodromos Kolyvakis[1]*, Alexandros Kalousis[2], Barry Smith[3] and Dimitris Kiritsis[1]

Abstract

Background: While representation learning techniques have shown great promise in application to a number of different NLP tasks, they have had little impact on the problem of ontology matching. Unlike past work that has focused on feature engineering, we present a novel representation learning approach that is tailored to the ontology matching task. Our approach is based on embedding ontological terms in a high-dimensional Euclidean space. This embedding is derived on the basis of a novel phrase retrofitting strategy through which semantic similarity information becomes inscribed onto fields of pre-trained word vectors. The resulting framework also incorporates a novel outlier detection mechanism based on a denoising autoencoder that is shown to improve performance.

Results: An ontology matching system derived using the proposed framework achieved an F-score of 94% on an alignment scenario involving the Adult Mouse Anatomical Dictionary and the Foundational Model of Anatomy ontology (FMA) as targets. This compares favorably with the best performing systems on the Ontology Alignment Evaluation Initiative anatomy challenge. We performed additional experiments on aligning FMA to NCI Thesaurus and to SNOMED CT based on a reference alignment extracted from the UMLS Metathesaurus. Our system obtained overall F-scores of 93.2% and 89.2% for these experiments, thus achieving state-of-the-art results.

Conclusions: Our proposed representation learning approach leverages terminological embeddings to capture semantic similarity. Our results provide evidence that the approach produces embeddings that are especially well tailored to the ontology matching task, demonstrating a novel pathway for the problem.

Keywords: Ontology matching, Semantic similarity, Sentence embeddings, Word embeddings, Denoising autoencoder, Outlier detection

Background

Ontologies seek to alleviate the Tower of Babel effect by providing standardized specifications of the intended meanings of the terms used in given domains. Formally, an ontology is "a representational artifact, comprising a taxonomy as proper part, whose representations are intended to designate some combinations of universals, defined classes and certain relations between them" [1]. Ideally, in order to achieve a unique specification for each term, ontologies would be built in such a way as to be non-overlapping in their content. In many cases, however, domains have been represented by multiple ontologies and there thus arises the task of *ontology matching*, which consists in identifying correspondences among entities (types, classes, relations) across ontologies with overlapping content.

Different ontological representations draw on the different sets of natural language terms used by different groups of human experts [2]. In this way, different and sometimes incommensurable terminologies are used to describe the same entities in reality. This issue, known as the *human idiosyncrasy* problem [1], constitutes the main challenge to discovering equivalence relations between terms in different ontologies.

Ontological terms are typically common nouns or noun phrases. According to whether they do or do not include prepositional clauses [3], the latter may be either composite (for example *Neck of femur*) or simple (for example *First tarsometatarsal joint* or just *Joint*). Such grammatical complexity of ontology terms needs to be taken into account in identifying semantic similarity. But account

*Correspondence: prodromos.kolyvakis@epfl.ch
[1]École Polytechnique Fédérale de Lausanne (EPFL), Route Cantonale, 1015 Lausanne, Switzerland
Full list of author information is available at the end of the article

must be taken also of the ontology's axioms and definitions, and also of the position of the terms in the ontology graph formed when we view these terms as linked together through the *is_a* (subtype), *part_of* and other relations used by the ontology.

The primary challenge to identification of semantic similarity lies in the difficulty we face in distinguishing true cases of similarity from cases where terms are merely "descriptively associated"[1]. As a concrete example, the word "harness" is descriptively associated with the word "horse" because a harness is often used on horses [4]. Yet the two expressions are not semantically similar. The sorts of large ontologies that are the typical targets of semantic similarity identification contain a huge number of such descriptively associated term pairs. This difficulty in distinguishing similarity from descriptive association is a well-studied problem in both cognitive science [5] and NLP [6].

Traditionally, feature engineering has been the predominant way to approach the ontology matching problem [7]. In machine learning, a feature is an individual measurable property of a phenomenon in the domain being observed [8]. Here we are interested in features of terms, for instance the number of incoming edges when a term is represented as the vertex of an ontology graph; or a terms's tf-idf value – which is a statistical measure of the frequency of a term's use in a corpus [9]. Feature engineering consists in crafting features of the data that can be used by machine learning algorithms in order to achieve specific tasks. Unfortunately determining which hand-crafted features will be valuable for a given task can be highly time consuming. To make matters worse, as Cheatham and Hitzler have recently shown, the performance of ontology matching based on such engineered features varies greatly with the domain described by the ontologies [10].

As a complement to feature engineering, attempts have been made to develop machine-learning strategies for ontology matching based on binary classification [11]. This means a classifier is trained on a set of alignments between ontologies in which correct and incorrect mappings are identified with the goal of using the trained classifier to predict whether an assertion of semantic equivalence between two terms is or is not true. In general, the number of true alignments between two ontologies is several orders of magnitude smaller than the number of all possible mappings, and this introduces a serious class imbalance problem [12]. This abundance of negative examples hinders the learning process, as most data mining algorithms assume balanced data sets and so the classifier runs the risk of degenerating into a series of predictions to the effect that every alignment comes to be marked as a misalignment.

Both standard approaches thus fail: feature engineering because of the failure of generalization of the engineered features, and supervised learning because of the class imbalance problem. Our proposal is to address these limitations through the exploitation of unsupervised learning approaches for ontology matching drawing on the recent rise of distributed neural representations (DNRs), in which for example words and sentences are embedded in a high-dimensional Euclidean space [13–17] in order to provide a means of capturing lexical and sentence meaning in an unsupervised manner. The way this works is that the machine learns a mapping from words to high-dimensional vectors which take account of the contexts in which words appear in a plurality of corpora. Vectors of words that appear in the same sorts of context will then be closer together when measured by a similarity function. That the approach can work without supervision stems from the fact that meaning capture is merely a positive externality of context identification, a task that is unrelated to the meaning discovery task.

Traditionally, corpus driven approaches were based on the *distributional hypothesis*, i.e. the assumption that semantically similar or related words appear in similar contexts [18]. This meant that they tended to learn embeddings that capture both similarity (*horse, stallion*) and relatedness (*horse, harness*) reasonably well, but do very well on neither [6, 19]. In an effort to correct for these biases a number of pre-trained word vector refining techniques were introduced [6, 20, 21]. These techniques are however restricted to retrofitting single words and do not easily generalize to the sorts of nominal phrases that appear in ontologies. Wieting et al. [22, 23] make one step towards addressing the task of tailoring phrase vectors to the achievement of high performance on the semantic similarity task by focusing on the task of paraphrase detection. A paraphrase is a restatement of a given phrase that use different words while preserving meaning. Leveraging what are called universal compositional phrase vectors [24] for the purposes of paraphrase detection provides training data for the task of semantic similarity detection which extends the approach from single words to phrases. Unfortunately, the result still fails as regards the problem of distinguishing semantic similarity and descriptive association on rare phrases [22] – constantly appearing on ontologies – which thus again harms performance in ontology matching tasks.

In this work, we tackle the aforementioned challenges and introduce a new framework for representation learning based ontology matching. Our ontology matching algorithm is structured as follows: To represent the nouns and noun-phrases in an ontology, we exploit the context information that accompanies the corresponding expressions when they are used both inside and outside the ontology. More specifically, we create vectors for ontology terms on the basis of information extracted not only from natural language corpora but also from terminological and lexical resources and we join this with information captured both explicitly and implicitly from the ontologies themselves. Thus we capture contexts in which words are used in definitions and in statements of synonym relations. We also draw inferences from the ontological resources themselves, for example to derive statements of descriptive association – the absence of a synonymous statement between two terms with closely similar vectors is taken to imply that as a statement of descriptive association obtains between them. We then cast the problem of ontology matching as an instance of the Stable Marriage problem [25] discovering in that way terminological mappings in which there is no pair of terms that would rather be matched to each other than their current matched terms. In order to compute the ordering of preferences for each term, that the Stable Marriage problem requires, we use the terminological representations' pairwise distances. We compute the aforementioned distances using the cosine distance over the phrases representations learned by the phrase retrofitting component. Finally, an outlier detection component sifts through the list of the produced alignments so as to reduce the number of misalignments.

Our main contributions in this paper are: (i) We demonstrate that word embeddings can be successfully harnessed for ontology matching; a task that requires phrase representations tailored to semantic similarity. This is achieved by showing that knowledge extracted from semantic lexicons and ontologies can be used to inscribe semantic meaning on word vectors. (ii) We additionally show that better results can be achieved on the discrimination task between semantic similarity and descriptive association, by casting the problem as an outlier detection. To do so, we present a denoising autoencoder architecture, which implicitly tries to discover a hidden representation tailored to the semantic similarity task. To the best of our knowledge, the overall architecture used for the outlier detection as well as its training procedure is applied for the first time to the problem of discriminating among semantically similar and descriptively associated terms. (iii) We use the biomedical domain as our application, due to its importance, its ontological maturity, and to the fact that it constitutes the domain with the larger ontology alignment datasets owing to its high variability in expressing terms. We compare our method to state-of-the-art ontology matching systems and show significant performance gains. Our results demonstrate the advantages that representation learning bring to the problem of ontology matching, shedding light on a new direction for a problem studied for years in the setting of feature engineering.

Problem formulation

Before we proceed with the formal definition of an ontological entity alignment, we will introduce the needed formalism. Let O, O' denote two set of terms used in two distinct ontologies and let R be a set of binary relations' symbols. For instance, $=, \neq, is\_a$ can be some of the R set's citizens. We introduce a set $T = \{(e, r, e') | e \in O, e' \in O', r \in R\}$ to denote a set of possible binary relations between O and O' [26]. Moreover, let $f: T \rightarrow [0, 1] \subset \mathcal{R}$ be a function, called "confidence function", that maps an element of T to a real number v, such that $0 \leq v \leq 1$. The real number v corresponds to the degree of confidence that exists a relation r between e and e' [27].

We call a set T of possible relations to be "valid despite integration inconsistency", iff T is satisfiable. As an counterexample, the set $\{(e, =, e'), (e, \neq, e')\}$ corresponds to a non-valid despite integration inconsistency set of relations. It should be noted that we slightly differentiated from the notation used in Description Logics [28], where a relation (Role) between two entities is denoted as: $r(e, e')$. Moreover, it is important to highlight the role of the phrase "despite integration inconsistency" in our definition. The ontology resulting from the integration of two ontologies O and O' via a set of alignments T may lead to semantic inconsistencies [29, 30]. As the focus of ontology alignment lays on the discovery of alignments between two ontologies, we treat the procedure of inconsistency check as a process that starts only after the end of the ontology matching process[2].

Based on the aforementioned notations and definitions, we will proceed with the formal definition of what an ontological entity alignment is. Let, T be a valid despite integration inconsistency set of relations and f be a confidence function defined over T. Let $(e, r, e') \in T$, we define an ontological entity correspondence between two entities $e \in O$ and $e' \in O'$ as the four-element tuple:

$$cor_r(e, e') = (e, r, e', f(e, r, e')) \quad (1)$$

where r is a matching relation between e and e' (e.g., equivalence, subsumption) and $f(e, r, e') \in [0, 1]$ is the degree of confidence of the matching relation between e and e'. According to the examples presented in Fig. 1, (triangular bone, =, ulnar carpal bone, 1.00) and (triangular bone, *is_a*, forelimb bone, 1.00) present one equivalence as well as a subsumption entity correspondence, accordingly. In this work, we focus on discovering one-to-one equivalence correspondences between two ontologies. In absence of further relations, the produced set of relations by our algorithm will always correspond to a valid despite integration inconsistency set.

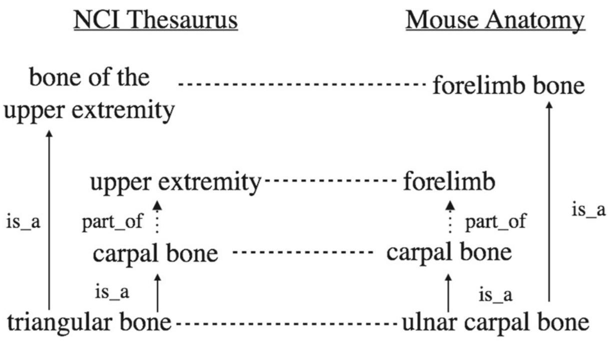

Fig. 1 Example of alignments between the NCI Thesaurus and the Mouse Ontology (adapted from [56]). The dashed horizontal lines correspond to equivalence matchings between the NCI Thesaurus and the Mouse Anatomy ontology

System architecture overview

Our ontology matching system is composed of two neural network components that learn which term alignments correspond to semantic similarity. The first component discovers a large amount of true alignments between two ontologies but is prone to errors. The second component corrects these errors. We present below an overview of the two components.

The first component, which we call *phrase retrofitting* component, retrofits word vectors so that when they are used to represent sentences, the produced sentence embeddings will be tailored to semantic similarity. To inscribe semantic similarity onto the sentence embeddings, we construct an optimization criterion which rewards matchings of semantically similar sentence vectors and penalizes matchings of descriptively associated ones. Thus the optimization problem adapts word embeddings so that they are more appropriate to the ontology matching task. Nonetheless, one of the prime motivations of our work comes from the observation that although supervision is used to tailor phrase embeddings to the task of semantic similarity, the problem of discriminating semantically similar vs descriptively associated terms is not targeted directly. This lack will lead to the presence of a significant number of misalignments, hindering the performance of the algorithm.

For that reason, we further study the discrimination problem in the setting of unsupervised outlier detection. We use the set of sentence representations produced by the phrase retrofitting component to train an denoising autoencoder [31]. The denoising autoencoder (DAE) aims at deriving a hidden representation that captures intrinsic characteristics of the distribution of semantically similar terms. We force the DAE to leverage new sentence representations by learning to reconstruct not only the original sentence but also its paraphrases, thus boosting the semantic similarity information that the new representation brings. Since we are using paraphrases to do so we bring in additional training data, doing essentially data augmentation for the semantically similar part of the problem. The DAE corresponds to our second component which succeeds in discovering misalignments by capturing intrinsic characteristics of semantically similar terms.

Methods

We present a representation learning based ontology matching algorithm that approaches the problem as follows: We use the ontologies to generate negative training examples that correspond to descriptively associated examples, and additional knowledge sources to extract paraphrases that will correspond to positive examples of semantic similarity. We use these training data to refine pre-trained word vectors so that they are better suited for the semantic similarity task. This task is accomplished by the phrase retrofitting component. We represent each ontological term as the bag of words of its textual description[3] which we complement with the refined word embeddings. We construct sentence representations of the terms' textual description by averaging the phrase's aforementioned word vectors. We match the entities of two different ontologies using the Stable Marriage algorithm over the terminological embeddings' pairwise distances. We compute the aforementioned distances using the cosine distance. Finally, we iteratively pass through all the produced alignments and we discard those that violate a threshold which corresponds to an outlier condition. We compute the outlier score using the cosine distance over the features created by an outlier detection mechanism.

Preliminaries

We introduce some additional notation that we will use throughout the paper. Let $sen_i = \left\{w_1^i, w_2^i, \ldots, w_m^i\right\}$ be the phrasal description[3] of a term i represented as a bag of m word vectors. We compute the sentence representation of the entity i, which we denote s_i, by computing the mean of the set sen_i, as per [24]. Let $s_i, s_j \in \mathbb{R}^d$ be two d-dimensional vectors that correspond to two sentence vectors, we compute their cosine distance as follows: $dis(s_i, s_j) = 1 - cos(s_i, s_j)$. In the following, d will denote the dimension of the pre-trained and retrofitted word vectors. For $x \in \mathbb{R}$, we denote the *rectifier* activation function as: $\tau(x) = \max(x, 0)$, and the *sigmoid* function as: $\sigma(x) = \frac{1}{1+e^{-x}}$.

Building sentence representations

In this section, we describe the neural network architecture that will produce sentence embeddings tailored to semantic similarity. Quite recently several works addressed the challenge of directly optimizing word vectors to produce sentence vectors by averaging the bag of the word vectors [22, 32, 33]. The interplay between semantical and physical intuition is that word vectors can be thought as corresponding to the positions of equally weighted masses, where the center of their masses provides information of the mean location of their semantic distribution. Intuitively, the word vectors' "center of the mass" provide a means for measuring where the semantic content primarily "concentrates". Despite the fact that vector addition is insensitive to word order [24], it has been proven that this syntactic agnostic operation provides results that compete favorably with more sophisticated syntax-aware composing operations [33]. We base our phrase retrofitting architecture on an extension of the Siamese CBOW model [32]. The fact that Siamese CBOW provides a native mechanism for discriminating between sentence pairs from different categories explains our choice to build upon this architecture.

Siamese CBOW is a log linear model aiming at predicting a sentence from its adjacent sentences; addressing the research question whether directly optimizing word vectors for the task of being averaged leads to better suited word vectors for this task compared to word2vec [15]. Let $V = \{v_1, v_2, \dots v_N\}$ be an indexed set of word vectors of size N. The Siamese CBOW model transforms a pretrained vector set V into a new one, $V' = \{v'_1, v'_2, \dots v'_N\}$, based on two sets of positive, S_i^+, and negative, S_i^-, constraints for a given training sentence s_i. The supervised training criterion in Siamese CBOW rewards co-appearing sentences while penalizing sentences that are unlikely to appear together. Sentence representations are computed by averaging the sentence's constituent word vectors. The reward is given by the pairwise sentence cosine similarity over their learned vectors. Sentences which are likely to appear together should have a high cosine similarity over their learned representations. In the initial paper of Siamese CBOW [32], the set S_i^+ corresponded to sentences appearing next to a given s_i, whereas S_i^- corresponded to sentences that were not observed next to s_i.

Since we want to be able to differentiate between semantically similar and descriptively associated sentences we let the sets S_i^+ and S_i^- to be sentences that are semantically similar and descriptively associated to a given sentence s_i. In the rest of the section we revise the main elements of the Siamese CBOW architecture and describe the modifications we performed in order to exploit it for learning sentence embeddings that reflect semantic similarity. To take advantage of the semantic similarity information already captured in the initial word vectors, an important characteristic as demonstrated in various word vectors retrofitting techniques [20–22], we use *knowledge distillation* [34] to penalize large changes in the learned word vectors with regard to the pre-trained ones.

Our paraphrase retrofitting model retrofits a pre-trained set of word vectors with the purpose of leveraging a new set V', solving the following optimization problem:

$$\min_{V'} \kappa_S L_S(V') + \kappa_{LD} L_{KD}(V, V'), \tag{2}$$

where k_S and k_{LD} are hyperparameters controlling the effect of $L_S(V')$ and $L_{KD}(V, V')$ losses, accordingly. The $L_S(V')$ term is defined as $\frac{1}{N}\sum_{i=1}^{N} L_{S_i}$, where N denotes the number of the training examples. The L_{S_i} term corresponds to categorical cross-entropy loss defined as:

$$L_{S_i} = -\sum_{s_j \in \{S_i^+ \cup S_i^-\}} p(s_i, s_j) \cdot \log(p_\theta(s_i, s_j)), \tag{3}$$

where $p(\cdot)$ is the target probability the network should produce, and $p_\theta(\cdot)$ is the prediction it estimates based on parameters θ, using Eq. 5. The target distribution simply is:

$$p(s_i, s_j) = \begin{cases} \frac{1}{|S^+|}, & \text{if } s_j \in S_i^+ \\ 0, & \text{if } s_j \in S_i^-. \end{cases} \tag{4}$$

For instance, if there are two positive and two negative examples, the target distribution is (0.5, 0.5, 0, 0). For a pair of sentences (s_i, s_j), the probability $p_\theta(s_i, s_j)$ is constructed to reflect how likely it is for the sentences to be semantically similar, based on the model parameter θ. The probability $p_\theta(s_i, s_j)$ is computed on the training data set based on the softmax function as follows:

$$p_\theta(s_i, s_j) = \frac{e^{\left(\cos\left(\mathbf{s_i^\theta}, \mathbf{s_j^\theta}\right)\right)^{1/T}}}{\sum_{s_k \in \{S_i^+ \cup S_i^-\}} e^{\left(\cos\left(\mathbf{s_i^\theta}, \mathbf{s_k^\theta}\right)\right)^{1/T}}}, \tag{5}$$

where s_x^θ denotes the embedding of sentence s_x, based on the model parameter θ. To encourage the network to better discriminate between semantically similar and descriptively associated terms, we extend the initial architecture by introducing the parameter T. The parameter T, named *temperature*, is based on the recent work of [34, 35]. Hinton et al. [34] suggest that setting $T > 1$ increases the weight of smaller logit (the inputs of the softmax function) values, enabling the network to capture information hidden in small logit values.

To construct the set S^-, we sample a set of descriptively associated terms from the ontologies to be matched. Given a sentence s_i, we compute its cosine distance with every term from the two ontologies to be matched, based on the initial pre-trained word vectors. Thereafter, we choose the n terms demonstrating the smaller cosine distance to be the negative examples. To account for that fact that among these n terms there may be a possible alignment, we exclude the n_* closest terms. Equivalently, given the increasingly sorted sequence of the cosine distances, we choose the terms in index positions starting from n_*

up to $n + n_*$. For computationally efficiency, we carry this process out only once before the training procedure starts.

Hinton et al. [34] found that using the class probabilities of an already trained network as "soft targets" for another one network constitutes an efficient way of communicating already discovered regularities to the latter network. We exploit, thus, knowledge distillation to emit the original semantic information captured in the pre-trained word vectors to the new ones leveraged by Siamese CBOW. Therefore, we add the Knowledge Distillation loss $L_{KD}(V, V') = \frac{1}{N}\sum_{i=1}^{N} L_{KD_i}$ to the initial Siamese CBOW's loss. The L_{KD_i} term:

$$L_{KD_i} = -\sum_{s_j \in \{S^+ \cup S^-\}} p_{\theta_I}(s_i, s_j) \cdot \log(p_\theta(s_i, s_j)), \tag{6}$$

is defined as the categorical cross-entropy between the probabilities obtained with the initial parameters (i.e. θ_I) and the ones with parameters θ.

Based on the observations of Hinton et al. [34], these "soft targets" act as an implicit reguralization, guiding the Siamese CBOW's solution closer to the initial word vectors. We would like to highlight that we experimented with various regularizers, such as the ones presented in the works of [20, 21, 23, 36], however, we obtained worse results than the ones reported in our experiments. Figure 2 summarizes the overall architecture of our phrase retrofitting model. The dashed rectangles in the *Lookup Layer* correspond to the initial word vectors, which are used to encourage the outputs of the Siamese CBOW network to approximate the outputs produced with the pre-trained ones in every epoch. The word embeddings are averaged in the next layer to produce sentence representations. The cosine similarities between the sentence representations are calculated in the penultimate layer and are used to feed a softmax function so as to produce a final probability distribution. Specifically, we compute the cosine similarity between the sentence representation of the noun phrase and the sentence representations of every positive and negative example of semantic similarity. In the final layer, this probability distribution is used to compute two different categorical cross entropy losses. The left loss encourages the probability distribution values to approximate a target distribution, while the right one penalizes large changes in the learned word vectors with regard to the pre-trained ones. The double horizontal lines in the *Cosine Layer* highlight that these rectangles denote in fact the same probability distribution, computed in the penultimate layer.

Outlier detection

The extension of the Siamese CBOW network retrofits pre-trained word vectors to become better suited for constructing sentence embeddings that reflect semantic similarity. Although we sample appropriate negative examples (i.e., descriptively associated terms) from the ontologies to be matched, we will never have all the negative examples needed. Moreover, allowing a larger number, *n*, of negative examples increases the computation needed making it inefficient. We depart from these problems by further casting the problem of discriminating between semantically similar and related terms as an outlier detection. To leverage an additional set of sentence representations more robust to semantic similarity, we use the hidden representation of a Denoising Autoencoder (DAE) [31].

The Siamese CBOW network learns to produce sentence embeddings of ontological terms that are better suited for the task of semantic similarity. We now use the learned sentence vectors to train a DAE. We extend the *standard* architecture of DAEs to reconstruct not only the sentence representation fed as input but also paraphrases of that sentence. Our idea is to improve the sentence representations produced by the Siamese CBOW and make them more robust to paraphrase detection. At the same time, this constitutes an efficient data augmentation technique; very important in problems with relatively small training data sets.

We train the autoencoder once the training of the Siamese CBOW network has been completed. Even if layer-wise training techniques [37] are outweighed nowadays by end-to-end training, we decide to adopt this

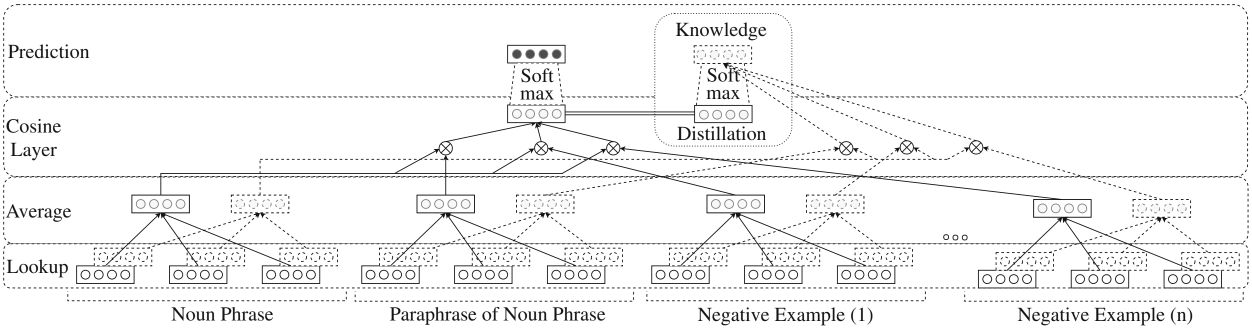

Fig. 2 Phrase Retrofitting architecture based on a Siamese CBOW network [32] and Knowledge Distillation [34]. The input projection layer is omitted

strategy for two reasons. Firstly, we aim to capture with the DAE intrinsic characteristics of the distribution of the semantically similar terms. DAEs have been proven to really capture characteristics of the data distribution, namely the derivative of the log-density with respect to the input [38]. However, training the DAE on a dataset that does not reflect the true distribution of semantically similar terms introduces surely a barrier to our attempt. Therefore, we leverage in advance sentence representations, through the Siamese CBOW network, more robust to semantic similarity; an action that allows the DAE to act on a dataset with significantly less noise and less bias. Secondly, combining the extended Siamese CBOW architecture together with the DAE and training them end-to-end significantly increases the number of the training parameters. This increase is a clear impediment to a problem lacking an oversupply of training data.

Let $x, y \in \mathbb{R}^d$ be two d-dimensional vectors, representing the sentence vectors of two paraphrases. Our target is not only to reconstruct the sentence representation from a corrupted version of it, but also to reconstruct a paraphrase of the sentence representation based on the partial destroyed one. The corruption process that we followed in our experiments is the following: for each input x, a fixed number of νd $(0 < \nu < 1)$ components are chosen at random, and their value is forced to 0, while the others are left untouched. The corrupted input $\tilde{x}$ is then mapped, as with the basic autoencoder, to a hidden representation $h = \tau(W\tilde{x} + b)$ from which we reconstruct a $z = \sigma(W'h + b')$. The dimension d_h of the hidden representation $h \in \mathbb{R}^{d_h}$ is treated as a hyperparameter. Similar to the work in [31], the parameters are trained to minimize, over the training set, an average reconstruction error. However, we aim not only to reconstruct the initial sentence but also its paraphrases. For that reason, we use the following reconstruction loss: $L(x, z) + L(y, z) =$

$$
\begin{aligned}
&= -\sum_{k=1}^{d} [x_k \log z_k + (1 - x_k) \log(1 - z_k)] \\
&\quad - \sum_{k=1}^{d} [y_k \log z_k + (1 - y_k) \log(1 - z_k)] \,.
\end{aligned} \tag{7}
$$

The x_k, z_k, y_k correspond to the Cartesian coordinates of vectors x, z and y, respectively. The overall process is depicted in Fig. 3. In this figure, the Lookup and Average layers are similar to the ones depicted in Fig. 2. A sentence representation x is corrupted to $\tilde{x}$. The autoencoder maps it to h (i.e., the hidden code) and attempts to reconstruct both x and the paraphrase embedding y.

Ontology matching

The two components that we have presented were build in such a way so that they learn sentence representations which try to disentagle semantic similarity and descriptive association. We will now use these representations to solve the ontology matching problem (Fig. 4). To align the entities from two different ontologies, we use the extension of the Stable Marriage Assignment problem to unequal sets [25, 39]. This extension of the stable marriage algorithm computes $1-1$ mappings based on a preference $m \times n$ matrix, where m and n is the number of entities in ontologies O and O', respectively. In our setting, a matching is not stable if: (i) there is an element $e_i \in O$ which prefers some given element $e_j \in O'$ over the element to which e_i is already matched, and (ii) e_j also prefers e_i over

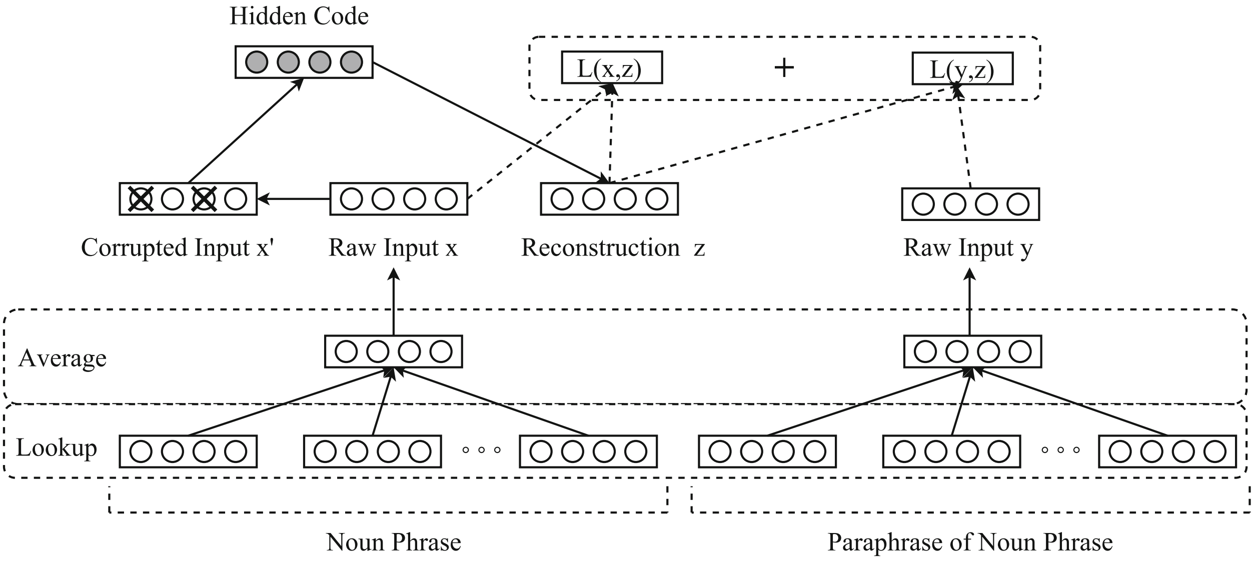

Fig. 3 Autoencoder Architecture

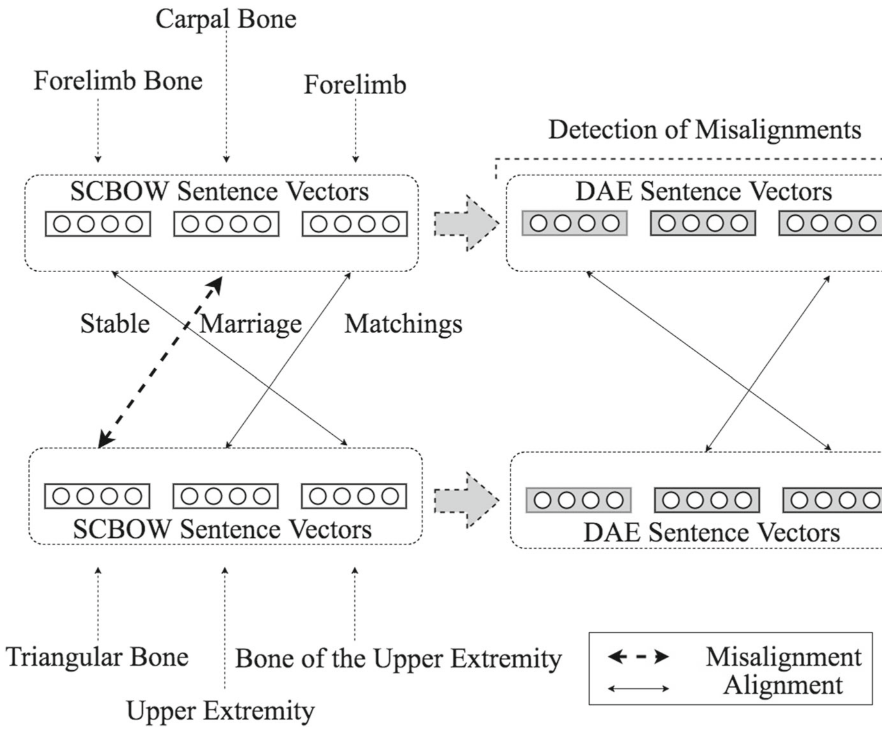

Fig. 4 Overall proposed ontology matching architecture

the element to which e_j is already matched. These properties of a stable matching impose that it does not exist any match (e_i, e_j) by which both e_i and e_j would be individually matched to more similar entities compared to the entities to which they are currently matched. This leads to a significant reduction in the number of misalignments due to descriptive association, provided that the learned representations do reflect the semantic similarity.

The steps of our ontology matching algorithm are the following: We represent each ontological term as the bag of words of its textual description, which we complement with the refined word vectors produced by the phrase retrofitting component. In the next step, we construct phrase embeddings of the terms' textual description[3] by averaging the phrase's word vectors. We cast the problem of ontology matching as an instance of the Stable Marriage problem using the entities' semantic distances. We compute these distances using the cosine distance over the sentences vectors. We iteratively pass through all the produced alignments and we discard those with a cosine distance greater than a certain threshold, t_1. These actions summarize the work of the first component. Note that the violation of the triangle inequality by the cosine distance is not an impediment to the Stable Marriage algorithm [25].

In the next step, we create an additional set of phrase vectors by passing the previously constructed phrase vectors through the DAE architecture. Based on this new embedding's set, we iteratively pass through all the alignments produced in the previous step and we discard those that report a threshold violation. Specifically, we discard those that exhibit a cosine distance, computed over the vectors produced by the DAE, greater than a threshold t_2. This corresponds to the final step of the outlier detection process as well as of our ontology matching algorithm.

Results and discussion

In this section, we present the experiments we performed on biomedical evaluation benchmarks coming from the Ontology Alignment Evaluation Initiative (OAEI), which organizes annual campaigns for evaluating ontology matching systems. We have chosen the biomedical domain for our evaluation benchmarks owing to its ontological maturity and to the fact that its language use variability is exceptionally high [40]. At the same time, the biomedical domain is characterized by rare words and its natural language content is increasing at an extremely high speed, making hard even for people to manage its rich content [41]. To make matters worse, as it is difficult to learn good word vectors for rare words from only a few examples [42], their generalization on their ontology matching task is questionable. This is a real challenge for domains, such as the biomedical, the industrial, etc, in which existence of words with rare senses is typical. The existence of rare words makes the presence of the phrase retroffiting component crucial to the performance of our ontology alignment framework.

Biomedical ontologies

We give a brief overview of the four ontologies used in our ontology mapping experiments. Two of them (the Foundational Model of Anatomy and the Adult Mouse anatomical ontologies) are pure anatomical ontologies, while the other two (SNOMED CT and NCI Thesaurus) are broader biomedical ontologies of which anatomy consists a subdomain that they describe [3]. Although more recent versions of these resources are available, we refer to the versions that appear in the Ontology Alignment Evaluation Initiative throughout this work in order to facilitate comparisons across the ontology matching systems.

Foundational Model of Anatomy (FMA): is an evolving ontology that has been under development at the University of Washington since 1994 [43, 44]. Its objective is to conceptualize the phenotypic structure of the human body in a machine readable form.

Adult Mouse Anatomical Dictionary (MA): is a structured controlled vocabulary describing the anatomical structure of the adult mouse [45].

NCI Thesaurus (NCI) provides standard vocabularies for cancer [46] and its anatomy subdomain describes naturally occurring human biological structures, fluids and substances.

SNOMED Clinical Terms (SNOMED): is a systematically organized machine readable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting [47].

Semantic lexicons

We provide below some details regarding the procedure we followed in order to construct pairs of semantically similar phrases. Let $(word^1_1, word^1_2, \ldots, word^1_m)$, be a term represented as a sequence of m words. The strategy that we have followed in order to create the paraphrases is the following: We considered all the contiguous subsequences of this term. Namely, we considered all the possible contiguous subsequences of the form: $(word^1_i, word^1_{(i+1)}, \ldots, word^1_j), \forall i,j \in \mathbb{N} : 0 \leq i \leq j \leq m$. Based on these contiguous subsequences, we queried the semantic lexicons for paraphrases. Below we give a brief summary of the semantic lexicons that we used in our experiments:

ConceptNet 5: a large semantic graph that describes general human knowledge and how it is expressed in natural language [48]. The scope of ConceptNet includes words and common phrases in any written human language.

BabelNet: a large, wide-coverage multilingual semantic network [49, 50]. BabelNet integrates both lexicographic and encyclopedic knowledge from WordNet and Wikipedia.

WikiSynonyms: a semantic lexicon which is built by exploiting the Wikipedia redirects to discover terms that are mostly synonymous [51].

Apart from the synonymy relations found in these semantic lexicons, we have exploited the fact that in some of the considered ontologies, a type may have one preferred name and some additional paraphrases [3], expressed through multiple rdfs:label relations.

Training

We tuned the hyperparameters on a set of 1000 alignments which we generated by subsampling the SNOMED-NCI ontology matching task[4]. We chose the vocabulary of the 1000 alignments so that it is disjoint to the vocabulary that we used in the alignment experiments, described in the evaluation benchmarks, in order to be sure that there is no information leakage from training to testing. We tuned to maximize the $F1$ measure. We trained with the following hyperparameters: word vector has size (d) 200 and is shared across everywhere. We initialized the word vectors from word vectors pre-trained on a combination of PubMed and PMC texts with texts extracted from a recent English Wikipedia dump [52]. All the initial out-of-vocabulary word vectors are sampled from a normal distribution ($\mu = 0, \sigma^2 = 0.01$). The resulted hyperparameters controlling the effect of retrofitting k_S and knowledge distillation k_{LD} were 10^6 and 10^3, accordingly. The resulted size of the DAE hidden representation (d_h) is 32 and ν is set to 0.4. We used $T = 2$ according to a grid search, which also aligns with the authors' recommendations [34]. For the initial sampling of descriptively associated terms, we used: $n_* = 2$ and $n = 7$. The best resulted values for the thresholds were the following: $t_1 = t_2 = 0.2$. The phrase retrofitting model was trained over 15 epochs using the Adam optimizer [53] with a learning rate of 0.01 and gradient clipping at 1. The DAE was trained over 15 epochs using the Adadelta optimizer [54] with hyperparameters $\epsilon = 1e - 8$ and $\rho = 0.95$.

Evaluation benchmarks

We provide some details regarding the respective size of each ontology matching task on Table 1.

The reference alignment of the MA - NCI matching scenario is based on the work of Bodenreider et al. [55].

Table 1 Respective sizes of the ontology matching tasks

Ontology Matching between:				#Matchings
Ontology I	#Types	Ontology II	#Types	
MA	2744	NCI	3304	1489
FMA	3696	NCI	6488	2504
FMA	10157	SNOMED	13412	7774

To represent each ontological term for this task, we used the unique rdfs:label that accompanies every type in the ontologies. The alignment scenarios between FMA - NCI and FMA - SNOMED are based on a small fragment of the aforementioned ontologies. The reference alignments of these alignment scenarios are based on the UMLS Metathesaurus [56], which currently consists the most comprehensive effort for integrating independently developed medical thesauri and ontologies. To represent each ontological term for these tasks, we exploited the textual information appearing on the rdf:about tag that accompanies every type in the ontologies. We did not use the rdf:about tag on the MA - NCI matching scenario, since their rdf:about tags provide a language agnostic unique identifier with no direct usable linguistic information. We would like to note that since the Stable Marriage algorithm provides one-to-one correspondences, we have only focused on discovering one-to-one matchings. In addition, a textual preprocessing that we performed led a small number of terms to degenerate into a single common phrase. This preprocessing includes case-folding, tokenization, removal of English stopwords and words coappearing in the vast majority of the terms (for example the word "structure" in SNOMED). Thereafter, we present on Table 1 the number of one-to-one types' equivalences remained after the preprocessing step.

Last but not least, it is of significant importance to highlight that the reference alignments based on UMLS Metathesaurus will lead to an important number of logical inconsistencies [57, 58]. As our method does not apply reasoning, whether it produces or not incoherence-causing matchings is a completely random process. In our evaluation, we have chosen to also take into account *incoherence-causing* mappings. However, various concerns can be raised about the fairness of comparing against ontology matching systems that make use of automated alignment repair techniques [58, 59]. For instance, the state-of-the-art systems AML [60, 61], LogMap and LogMapBio [62], which are are briefly described in the next section, do employ automated alignment repair techniques. Our approach to use the original and incoherent mapping penalizes these systems that perform additional incoherence checks.

Nonetheless, our choice to include inconsistence mappings can be justified in the following way. First, it is a direct consequence of the fact that we approach the problem of Ontology Matching from the viewpoint of discovering semantically similar terms. A great number of these inconsistent mappings do correspond to semantically similar terms. Second, we believe that ontology matching can also be used as a curation process during the ontological (re)design phase so as to alleviate the possibility of inappropriate terms' usage. The fact that two distinct truly semantically similar terms from two different ontologies lead to logical inconsistencies during the integration phase can raise an issue for modifying the source ontology [57]. Third, although ontologies constitute a careful attempt to ascribe the intended meaning of a vocabulary used in a target domain, they are error prone as every human artifact. Incoherence check lays on the assumption that both of the ontologies that are going to be matched are indeed error-free representational artifacts. We decide not to make this assumption.

Therefore, we have chosen to treat even the systems that employ automated alignment repair techniques error-prone. For that reason, we considered appropriate to report the performance of the aforementioned systems on the complete reference alignment in the next section. Nevertheless, we refer the reader to the [58] for details on the performance of these systems on incoherence free subsets of the reference alignment set. Under the assumption that the ontologies to be matched are error-free, it can be observed that the automated alignment repair mechanisms of these systems are extremely efficient; a fact that demonstrates the maturity and the robustness of these methods.

Experimental results

Table 2 shows the performance of our algorithm compared to the six top performing systems on the evaluation benchmarks, according to the results published in OAEI Anatomy track (MA - NCI) and in the Large BioMed track (FMA-NCI, FMA-SNOMED)[5]. To check for the statistical significance of the results, we used the procedure described in [63]. The systems presented in Table 2 starting from the top of the table up to and including LogMapBio fall into the category of feature engineering[6]. CroMatcher [64], AML [60, 61] and XMap [65] perform ontology matching based on heuristic methods that rely on aggregation functions. FCA_Map [66, 67] uses Formal Concept Analysis [68] to derive terminological hierarchical structures that are represented as lattices. The matching is performed by aligning the constructed lattices taking into account the lexical and structural information that they incorporate. LogMap and LogMapBio [62] use logic-based reasoning over the extracted features and cast the ontology matching to a satisfiability problem. Some of the systems compute many-to-many alignments between ontologies. For a fair comparison of our system with them, we have also restricted these systems in discovering one-to-one alignments. We excluded the results of XMap for the Large BioMed track, because it uses synonyms extracted by the UMLS Metathesaurus. Systems that use the UMLS Metathesaurus as background knowledge will have a notable advantage since the Large BioMed track's reference alignments are based on it.

We describe in the following the procedure that we followed in order to evaluate the performance of the various

Table 2 Performance of ontology matching systems across the different matching tasks.

System	MA - NCI			FMA-NCI			FMA-SNOMED		
	P	R	F1	P	R	F1	P	R	F1
AML	0.943	0.94	**0.941**	0.908	0.94	0.924	0.938	0.784	0.854
CroMatcher	0.942	0.912	0.927	-	-	-	-	-	-
XMap	0.924	0.877	0.9	-	-	-	-	-	-
FCA_Map	0.922	0.841	0.880	0.89	0.947	0.918	0.918	0.857	0.886
LogMap	0.906	0.850	0.878	0.894	0.930	0.912	0.933	0.721	0.814
LogMapBio	0.875	0.900	0.887	0.88	0.938	0.908	0.93	0.727	0.816
Wieting	0.804	0.879	0.839	0.840	0.857	0.849	0.867	0.851	0.859
Wieting+DAE(O)	0.952	0.871	0.909	0.909	0.851	0.879	0.929	0.832	0.878
SCBOW	0.847	0.917	0.881	0.899	0.895	0.897	0.843	0.866	0.855
SCBOW+DAE(O)	0.968	0.913	0.94	0.976	0.892	**0.932**	0.931	0.856	**0.892**

Note: Bold and underlined numbers indicate the best *F*1-score and the best precision on each matching task, respectively

ontology matching systems. Since the incoherence-causing mappings were also taken into consideration, all the mappings marked as "?" in the reference alignment were considered as positive. To evaluate the discovery of one-to-one matchings, we clustered all the m-to-n matchings and we counted only once when any of the considered systems discovers any of the m-to-n matchings. Specifically, let $T = \{(e, =, e')|e \in O, e' \in O'\}$ be a set of clustered m-to-n matchings. Once an ontology matching system discovers for the first time a $(e, =, e') \in T$, we increase the number of the discovered alignments. However, whenever the same ontology matching system discovers an additional $(e_*, =, e'_*) \in T$, where $(e, =, e') \neq (e_*, =, e'_*)$, we did not take this discovered matching into account. Finally, to evaluate the performance of AML, CroMatcher, XMap, FCA_MAP, LogMap, and LogMapBio, we used the alignments provided by OAEI 2016[5] and applied the procedure described above to get their resulted performance.

To explore the performance details of our algorithm, we report in Table 2 its performance results with and without outlier detection. Moreover, we included experiments in which instead of training word embeddings based on our extension of the Siamese CBOW, we have used the optimization criterion presented in [23] to produce an alternative set of word vectors. As before, we present experiments on which we exclude our outlier detection mechanism and experiments on which we allow it[7]. We present these experiments under the listings: SCBOW, SCBOW+DAE(O), Wieting, Wieting+DAE(O), accordingly.

SCBOW+DAE(O) is the top performing algorithm in two of the three ontology mappings tasks (FMA-NCI, FMA-SNOMED); in these two its F1 score is significantly better than that of all the other algorithms. In MA-NCA its F1 score is similar to AML, the best system there, but the performance difference is statistically significant. At the same time, SCBOW+DAE(O) achieves the highest precision on two out of three ontology matching tasks. In terms of recall, SCBOW+DAE(O) demonstrates lower performance in the ontology matching tasks. However, we would like to note that we have not used any semantic lexicons specific to the biomedical domains compared to the other systems. For instance, AML uses three sources of biomedical background knowledge to extract synonyms. Specifically, it exploits the Uber Anatomy Ontology (Uberon), the Human Disease Ontology (DOID), and the Medical Subject Headings (MeSH). Hence, our reported recall can be explained due to the lower coverage of biomedical terminology in the semantic lexicons that we have used. Our motivation for relying only on domain-agnostic semantic lexicons[8] stems from the fact that our intention is to create an ontology matching algorithm applicable to many domains. The success of these general semantic lexicons for such a rich in terminology domain, provides additional evidence that the proposed methodology may also generalize to other domains. However, further experimentation is needed to verify the adequacy and appropriateness of these semantic lexicons to other domains. It is among our future directions to test the applicability of our proposed algorithm to other domains.

Comparing the recall[9] of SCBOW and SCBOW+DAE(O), we see that the incorporation of the DAE produces sentence embeddings that are tailored to the semantic similarity task. The small precision of SCBOW, in all experiments, indicates a semantic similarity and descriptive association coalescence. Considering both the precision and the recall metric, we can observe that the outlier detection mechanism identifies misalignments while preserving most of the true alignments. This fact provides empirical support on the necessity of the outlier detection. To validate the importance of our phrase retrofitting component, we further analyze the behavior of aligning ontologies based on the word embedding produced by running the procedure described in [23] (listed as Wieting). As we can see SCBOW achieves statistically significant higher recall than Wieting in all our experiments and in two of the three cases statistically significant greater precision. This behavior indicates the superiority of SCBOW in injecting semantic similarity to word embeddings as well as to produce word vectors tailored to the ontology matching task. We further extended the Wieting experiment by applying our outlier detection mechanism trained on the word vectors produced by the procedure described in [23]. It can be seen that this extension leads to the same effects as the ones summarized in the SCBOW - SCBOW+DAE(O) comparison. These results give evidence that our DAE-based outlier detection component constitutes a mechanism applicable to various sentence embeddings' producing architectures.

Ablation study

In this section, an ablation study is carried out to investigate the necessity of each of the described components, as well as their effect on the ontology matching performance. Figure 5 shows a feature ablation study of our method; in Table 3 we give the descriptions of the experiments. We conducted experiments on which the phrase retrofitting component was not used, hence the ontology matching task was only performed based on the pre-trained word vectors. Moreover, we have experimented on performing the ontology matching task with the features generated by the DAE. Our prime motivation was to test whether the features produced by the DAE could be used to compute the cosine distances needed for estimating the preference matrix used by the Stable Marriage's algorithm. Hence, we differentiate in this subsection and we allow the DAE features to be used for Matching and/or Outlier Detection.

To begin with, it can be observed that all the performance metrics' figures undergo the same qualitative behavior. This result demonstrates that our algorithm exhibits a consistent behavior under the ablation study across all the experiments, which constitutes an important factor for inducing conclusions from experiments. The experiment W2V gives the results of executing the algorithm without the phrase retrofitting process, just by providing the pre-trained word vectors [52]. The performance of W2V in terms of Precision/Recall is systematically lower compared to all cases in which the initial word2vec vectors are retrofitted. These results support the importance of the phrase retrofitting process (experiments of which are presented under the listing SCBOW in Fig. 5), which succeeds in tailoring the word embeddings to the ontology matching task. The pre-trained word vectors, even though they were trained on PubMed and PMC texts, retain small precision and recall. This fact indicates a semantic similarity and descriptive association coalescence and sheds light on the importance of the retrofitting procedure.

Training the DAE on the pre-trained word vectors - DAE(O) - adds a significant performance gain on precision, which witnesses the effectiveness of the architecture for outlier detection. However, DAE(O)'s precision is almost the same as the one presented in the SCBOW experiment. Only when the phrase retrofitting component is combined with the DAE for outlier detection - SCBOW+DAE(O) - we manage to surpass the aforementioned precision value and achieve our best $F1$-score. Finally, our experiments on aligning ontologies by only using the DAE features demonstrate that these features are inadequate for this task. One prime explanation of

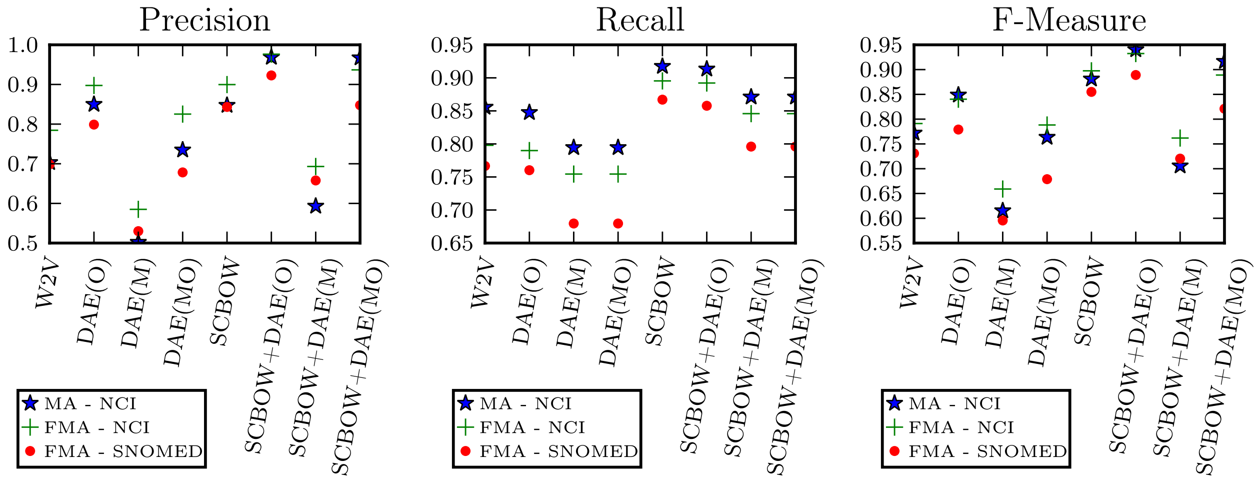

Fig. 5 Feature ablation study of our proposed approach across all the experimental ontology matching tasks

Table 3 Ablation study experiment's listings

Experiment's code:	Phrase retrofitting	DAE features:	
		Matching	Outlier detection
W2V	-	-	-
DAE(O)	-	-	✓
DAE(M)	-	✓	-
DAE(MO)	-	✓	✓
SCBOW	✓	-	-
SCBOW+DAE(O)	✓	-	✓
SCBOW+DAE(M)	✓	✓	-
SCBOW+DAE(MO)	✓	✓	✓

this behavior is that DAE features are only exposed to synonymy information. At the same time, the dimensionality reduction of DAE features may lead them to lose a lot of valuable information captured in them for discriminating between semantically similar and descriptively associated terms. Note also that the preference matrix required by the Stable Marriage solution requires each term of an ontology O to be compared across all the possible terms of another ontology O'. Thereafter, the vectors based on which the preference matrix will be computed need to capture the needed information adequate for discriminating between semantically similar and descriptive associated terms.

Error analysis

Recent studies provide evidence that different sentence representations objectives yield different intended representation preferable for different intended applications [33]. Moreover, our results reported in Table 2 on aligning ontologies with word vectors trained based on the method presented in [23] provide further evidence in the same direction. In Table 4, we demonstrate a sample of misalignments produced by aligning ontologies using the Stable Marriage's solution based on a preference matrix computed either on SCBOW or Word2Vec vectors. It can be seen that the SCBOW misalignments demonstrate even a better spatial consistency compared to the Wor2Vec misalignments. This result combined with high $F1$-score reported in the SCBOW results in Table 5 show that ontological knowledge can be an important ally in the task of harnessing terminological embeddings tailored to semantic similarity. Moreover, this error analysis provides additional support for the significance of retrofitting general-purpose word embeddings before being applied in a domain-specific setting. It can be observed that general-purpose word vectors capture both similarity and relatedness reasonably well, but neither perfectly as it has been already observed in various works [6, 19].

Runtime analysis

In this section, we report the runtimes of our ontology matching algorithm for the different matching scenarios. Since our method – SCBOW+DAE(O) – consists of three major steps, we present in Table 5 the time devoted to each of them as well as their sum. In brief, the steps of our algorithm are the following: the training of the phrase retrofitting component (Step 1), the solution to the stable marriage assignment problem (Step 2), and finally the training of the DAE-based outlier detection mechanism (Step 3). All the reported experiments were performed on a desktop computer with an Intel® Core™ i7-6800K (3.60GHz) processor with 32GB RAM and two NVIDIA® GeForce® GTX™ 1080 (8GB) graphic cards. The implementation was done in Python using Theano [69, 70].

As it can be seen on Table 5, the majority of the time is allotted to the training of the phrase retrofitting framework. In addition, it can be observed that the training overhead of the outlier detection mechanism is

Table 4 Sample misalignments produced by aligning ontologies using either SCBOW or Word2Vec vectors

Terminology to be matched	Matching based on SCBOW	Matching based on Word2Vec
MA-NCI		
gastrointestinal tract	digestive system	respiratory tract
tarsal joint	carpal tarsal bone	metacarpo phalangeal joint
thyroid gland epithelial tissue	thyroid gland medulla	prostate gland epithelium
FMA-NCI		
cardiac muscle tissue	heart muscle	muscle tissue
set of carpal bones	carpus bone	sacral bone
white matter of telencephalon	brain white matter	white matter
FMA-SNOMED		
zone of ligament of ankle joint	accessory ligament of ankle joint	entire ligament of elbow joint
muscle of anterior compartment of leg	compartment of lower leg	entire interosseus muscle of hand
dartos muscle	dartos layer of scrotum	tendon of psoas muscle

Table 5 Runtimes of the steps in the proposed algorithm

Matching task	Running time (seconds)			
	Step 1	Step 2	Step 3	Total
MA - NCI	337	34	36	407
FMA - NCI	490	82	40	612
FMA - SNOMED	609	490	41	1140

significantly smaller compared to the other steps. However, one important tendency can be observed in the FMA - SNOMED matching scenario. Specifically, the runtime of the second step has considerably increased and is comparable to the runtime of the first step. This can be explained by the worst-case time complexity of the McVitie and Wilson's algorithm [39], that has been used, which is $\mathcal{O}\left(n^2\right)$. Moreover, the computation of the preference matrix required for defining the stable marriage assignment problem's instance has worst-case time complexity $\Theta\left(n^2 \log n\right)$. At the same time, the space complexity of the second step is $\mathcal{O}\left(n^2\right)$, since it requires the storage of the preference matrices. On the contrary, various techniques [71, 72] and frameworks [69, 70, 73, 74] have been proposed and implemented for distributing the training and inference task of DNRs. Although our implementation exploits these highly optimized frameworks for DNRs, the choice of using the McVitie and Wilson's algorithm introduces a significant performance barrier for aligning larger ontologies than the ones considered in our experiments. However, it was recently shown that a relationship exists between the class of computing greedy weighted matching problems and the stable marriage problems [75]. The authors exploit this strong relationship to design scalable parallel implementations for solving large instances of the stable marriage problems. It is among our future work to test the effectiveness of those implementations as well as to experiment with different graph matching algorithms that will offer better time and space complexity.

Importance of the ontology extracted synonyms

As described in "Semantic lexicons" section, apart from the synonymy information extracted from ConceptNet 5, BabelNet, and WikiSynonyms, we have exploited the fact that, in some of the considered ontologies, a type may have one preferred name and some additional paraphrases expressed through multiple rdfs:label relations. In this section, we provide an additional set of experiments that aims to measure the importance of these extracted synonyms. This extracted synonymy information constitutes the 0.008%, 0.26%, 0.65% of the training data used in the MA - NCI, FMA - NCI, FMA - SNOMED matching scenarios, respectively. The high variance in their contribution to the training data provide us a means for partially evaluating the correlation between the relative change in the training data and the F1-score.

In Table 6, we compare the performance of SCBOW and SCBOW+DAE(O) trained with only the available information from the semantic lexicons, with that presented in Table 2 where all the the synonymy information was available. It can be observed that the additional synonymy information affects positively both SCBOW and SCBOW+DAE(O). To better illustrate this correlation, we present in Fig. 6 how the relative change in the training data is reflected to the relative difference in the performance of our algorithm. It transpires that the F1-score's relative change monotonically increases with the relative difference in the available data. This behavior constitutes a consistency check for our proposed method, since it aligns with our intuition that increasing the synonymy information leads to producing terminological embeddings more robust to semantic similarity. Regarding the additional benefit that this additional synonymy information brings, a maximum gain of 0.07 in the F1-score is observed across all the matching scenarios. This fact provides supplementary empirical support on the adequacy of the used general semantic lexicons as a means of providing the semantic similarity training data needed by our method. Although this additional synonymy information is important for comparing favorably with the state-of-the-art systems, it does not constitute a catalytic factor for the method's success.

Nonetheless, further experimentation is needed to verify the adequacy of these general semantic lexicons as well as to investigate the correlation between the training data size and the proposed method's performance. We leave for future work the further experimentation with supplementary matching scenarios, different training data sizes and synonymy information sources.

Table 6 Proposed algorithm's performance in relation to the used synonymy information sources

System	Training data	MA - NCI			FMA - NCI			FMA - SNOMED		
		P	R	F1	P	R	F1	P	R	F1
SCBOW	SL	0.845	0.911	0.877	0.897	0.840	0.868	0.795	0.773	0.784
SCBOW	SL + AS	0.847	0.917	0.881	0.899	0.895	0.897	0.843	0.866	0.855
SCBOW + DAE(O)	SL	0.946	0.905	0.925	0.972	0.830	0.895	0.912	0.759	0.829
SCBOW + DAE(O)	SL + AS	0.968	0.913	0.94	0.976	0.892	0.932	0.931	0.856	0.892

Note: SL: synonyms only from ConceptNet 5, BabelNet, and WikiSynonyms; AS: additional synonyms found in the ontologies to be matched

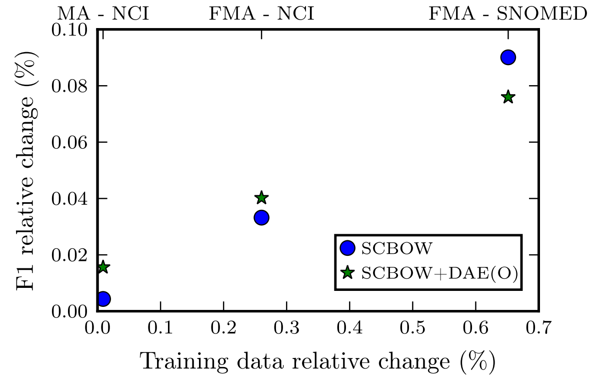

Fig. 6 Correlation between the relative change in training data's size and $F1$-score

Threshold sensitivity analysis

In this section, we perform a sensitivity analysis for the thresholds t_1 and t_2. These thresholds constitute a means for quantifying if two terms are semantically similar or descriptively associated. It is worth noting that the tuning of these thresholds can be decoupled. Equivalently, the t_1 threshold can be tuned to optimize the performance of SCBOW, and based on the resulted value the tuning of t_2 can be performed so as to optimize the performance of the outlier detection mechanism. Figure 7 shows a threshold sensitivity analysis of our method. For exploring the effect of t_1, we present on the left sub-figure of Fig. 7 the performance of SCBOW for all the different matching scenarios when varying the value of threshold t_1 between 0 and 1.0. Similarly, the right sub-figure of Fig. 7 shows the performance of SCBOW+DAE(O) when t_1 is set to 0.2 and the value of t_2 varies in $[0, 1.0]$.

To begin with, it can be seen that both of the threshold sensitivity analysis' figures undergo analogous qualitative behavior across the different ontology matching tasks. At the same time, it is observed that the performance (F1-score) monotonically increases when the value of t_1 varies between 0 and approximately 0.2. In the t_1 sub-figure, the performance monotonically decreases with $t_1 \in [0.2, 0.6]$ and reaches an asymptotic value at about 0.6. In the case of t_2, although the performance decreases when the value of t_2 exceeds 0.2, the rate of the decrease is significantly lower compared to the rate of decrease of t_1.

It can be seen that although further tuning and experimentation with the values of t_1, t_2 can give better results for each ontology matching task, the values that resulted from the hyperparameter tuning (described in "Training" section) are significantly close to the optimal ones. Moreover, it can be concluded that t_1 values greater than 0.2 have a greater negative impact on the performance compared to the performance drop when t_2 exceeds 0.2. Finally, it should be highlighted that apart from the hyperparameter tuning, no additional direct supervision based on the ground truth alignments is used by our method when we align the ontologies of the considered matching scenarios.

Implications & limitations

Traditionally, ontology matching approaches have been based on feature engineering in order to obtain different measures of similarity [27]. This plethora of multiple and complementary similarity metrics has introduced various challenges including choosing the most appropriate set of similarity metrics for each task, tuning the various cut-off thresholds used on these metrics, etc. [76]. As a solution to these challenges, various sophisticated solutions have been proposed such as automating the configuration selection process by applying machine learning algorithms on a set of features extracted from the ontologies [76]. Unlike in our approach, only one similarity distance is used; the cosine distance upon the learned features of the phrase retrofitting and the DAE framework. Therefore, there is a drastic decrease in the used similarity metrics and thresholds.

At the same time, it was an open question whether ontology's structural information is really required for performing ontology matching. Our proposed algorithm manages to compare favorably against state-of-the-art systems without using any kind of structural information.

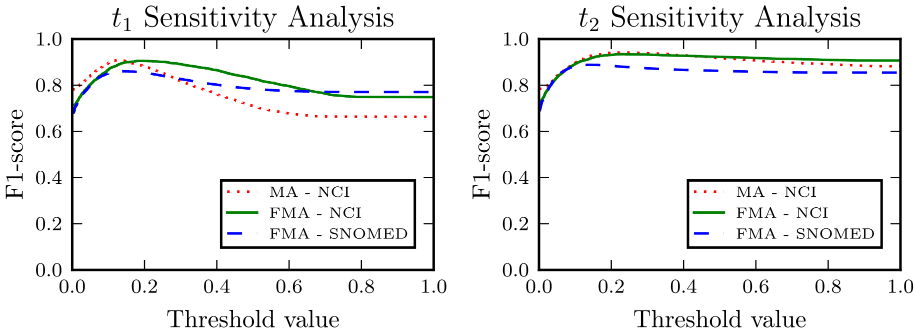

Fig. 7 Sensitivity analysis of the proposed algorithm's performance with different threshold values

Our results support that a great ontology matching performance can be achieved even in the absence of any graph-theoretic information. However, we avoid to conclude that structural information is not necessary. We leave for future work the investigation of how the ontology's structural information can be exploited in the frame of DNRs. Similarly, our method relies on word vectors pre-trained on large external corpora and on synonymy information provided by semantic lexicons also including the ontologies to be matched. Consequently, we can make the conclusion that external corpora and semantic lexicons provide sufficient information to perform ontology matching by only exploiting the ontologies' terms.

Nonetheless, our approach has also certain shortcomings. To begin with, our proposed algorithm is restricted on discovering one-to-one correspondences between two ontologies. At the same time, the use of the McVitie and Wilson's algorithm in our current implementation introduces a significant performance barrier for aligning lager ontologies than the ones considered in our experiments. Although our experimental results demonstrated that high precision can be achieved without using the OWL's reasoning capabilities, our recall remains lower compared to the state-of-the-art systems across all the ontology matching tasks. Taking into account the results presented in "Importance of the ontology extracted synonyms" section, it may be concluded that more synonymy information is required to be extracted from supplementary semantic lexicons so as to increase this performance metric. This observation introduces another one weakness of our algorithm; that of closely depending on available external corpora and semantic lexicons. All the aforementioned open questions and shortcomings demonstrate various interesting and important directions for our future work and investigation.

Related work

Representation Learning for Ontology Matching: Ontology matching is a rich research field where multiple and complementary approaches have been proposed [7, 77]. The vast majority of the proposed approaches, applied on the matching scenarios used in this paper, perform ontology matching by exploiting various terminological and structural features extracted from the ontologies to be matched. In parallel, they make use of various external semantic lexicons such as Uberon, DOID, Mesh, BioPortal ontologies and Wordnet as a means for incorporating background knowledge useful for discovering semantically similar terms. CroMatcher [64], AML [60, 61] and XMap [65] extract various sophisticated features and use a variety of the aforementioned external domain-specific semantic vocabularies to perform ontology matching. Moreover, LogMap, AML and XMap exploit complete and incomplete reasoning techniques so as to repair incoherent mappings [78]. Unlike the aforementioned approaches, FCA_Map [66, 67] uses Formal Concept Analysis [68] to derive terminological hierarchical structures that are represented as lattices. The matching is performed by aligning the constructed lattices taking into account the lexical and structural information that they incorporate. PhenomeNet [79] exploits an axiom-based approach for aligning ontologies that make use of the PATO ontology and Entity-Quality definition patterns [80, 81]; complementing in that way some of the shortcomings of feature-based methods.

Representation learning has so far limited impact on ontology matching. To the best of our knowledge, only two approaches, [82–84], have explored so far the use of unsupervised deep learning techniques. Both of these approaches use a combination of the class ID, labels, comments, etc. to describe an ontological entity in their algorithms. Zhang et al. [82] are the first ones that investigated the use of word vectors for the problem of ontology matching. They align ontologies based on *word2vec* [14] vectors trained on Wikipedia. They were the first that reported that the general-purpose word vectors were not good candidates for the task of ontology matching. Xiang et al. [83, 84] proposed an entity representation learning algorithm based on Stacked Auto-Encoders [37, 85]. However, training such powerful models with so small training sets is problematic. We overcome both of the aforementioned problems by using a transfer learning approach, known to reduce learning sample complexity [86], which retrofits pre-trained word vectors to a given ontological domain.

Sentence Representations from Labeled Data: To constrain the analysis, we compare neural language models that derive sentence representations of short texts optimized for semantic similarity based on pre-trained word vectors. Nevertheless, we consider in our comparison the initial Siamese CBOW model [32]. Likewise, we do not focus on innovative supervised sentence models based on neural networks architectures with more than three layers including [87, 88] and many others. The most similar approach to our extension on Siamese CBOW is the work of Wieting et al. [22]. Wieting et al. address the problem of paraphrase detection where explicit semantic knowledge is also leveraged. Unlike in our approach, a margin-based loss function is used, and negative examples should be sampled at every step introducing an additional computational cost. The most crucial difference is that this model was not explicitly constructed for alleviating the coalescence of semantically similar and descriptively associated terms. Finally, the initial Siamese CBOW model was conceived for learning distributed representations of sentences from unlabeled data. To take advantage of the semantic similarity information already captured in the initial word embeddings, an important characteristic as demonstrated in various word vectors retrofitting techniques [20–22], we extended the initial model with an knowledge distillation regularizer. Finally, we further extended the initial softmax setting, with a tempered softmax, with the purpose of enabling the network to capture information hidden in small logit values.

Autoencoders for Outlier Detection: Neural networks applications to the problem of outlier detection have been studied for a long time [89, 90]. Autoencoders seem to be a recent and a very prominent approach to the problem. As has been pointed out in [91], they can be seen as a generalization of the class of linear schemes [92]. Usually, the reconstruction error is used as the outlier score [91]. Recently, Denoising Autoencoders (DAEs) have been used for outlier detection in various applications, such as acoustic novelty detection [93], network's intrusion detection [91], anomalous activities' discovery in video [94]. To the best of our knowledge, this is the first time that the problem of semantic similarity is seen from the viewpoint of outlier detection based on DAEs. Unlike the other approaches, we want to detect outliers in pairs of input. To achieve that we use the cosine distance over the two produced hidden representations as an outlier score, instead of using the reconstruction error which is customary in the literature. Our motivation is that intrinsic characteristics of the distribution of semantically similar terms are captured in the hidden representation and their cosine distance could serve as an adequate outlier score. Unlike the majority of the aforementioned work, we do not train end-to-end the DAE but we follow a layer-wise training scheme based on sentence representations produced by our extension of Siamese CBOW. Our impetus is to let the DAE to act on a dataset with significant less noise and bias.

Conclusions

In this paper, we address the problem of ontology matching from a representation learning perspective. We propose the refinement of pre-trained word vectors so that when they are used to represent ontological terms, the produced terminological embeddings will be tailored to the ontology matching task. The retrofitted word vectors are learned so that they incorporate domain knowledge encoded in ontologies and semantic lexicons. We cast the problem of ontology matching as an instance of the Stable Marriage problem using the terminological vectors' distances to compute the preference matrix. We compute the aforementioned distances using the cosine distance over the terminological vectors learned by our proposed phrase retrofitting process. Finally, an outlier detection component, based on a denoising autoencoder, sifts through the list of the produced alignments so as to reduce the number of misalignments. Our experimental results demonstrate significant performance gains over the state-of-the-art and indicate a new pathway for ontology matching; a problem which has been traditionally studied under the setting of feature engineering.

Endnotes

[1] This term is known in the NLP community as "conceptually associated". We have chosen to depart from the standard terminology for reasons summarized in [95, p. 7].

[2] We provide further justification for this choice in "Evaluation benchmarks" section.

[3] We provide further details on the textual information used in our experiments in "Evaluation benchmarks" section.

[4] available on OAEI's 2016 Large BioMed Track.

[5] http://oaei.ontologymatching.org/2016/

[6] For a detailed overview and comparison of the systems refer to [96].

[7] We have also performed hyperparameter tuning in the SNOMED-NCI matching task, which gave the same hypeparameters as the ones reported in [23].

[8] except for the synonymy information found in some ontologies and is expressed through multiple labels (rdfs:label) for a given type.

[9] All the experiments are statistically significant with a p-value ≤ 0.05.

Funding
This project was supported by the Swiss State Secretariat for Education, Research and Innovation SERI (SERI; contract number 15.0303) through the European Union's Horizon 2020 research and innovation programme (grant agreement No 688203; bIoTope). This paper reflects the authors' view only, and the EU as well as the Swiss Government is not responsible for any use that may be made of the information it contains.

Availability of data and materials
The ontologies and the ground truth alignments used for these ontology alignment tasks are publicly available and can be found at the following url: http://oaei.ontologymatching.org/2016/. The word vectors pre-trained on PubMed, PMC texts and an English Wikipedia dump [52] can be found at http://evexdb.org/pmresources/vec-space-models/wikipedia-pubmed-and-PMC-w2v.bin. The code for the experiments with the word vectors produced by the work of Wieting et al. [22] can be downloaded from https://github.com/jwieting/iclr2016. Our implementation of the ontology matching framework based on representation learning, which was presented in this paper, can be accessed from https://doi.org/10.5281/zenodo.1173936.

Authors' contributions
PK designed and implemented the system, carried out the evaluations and drafted the manuscript. AK participated in the design of the method, discussed the results and corrected the manuscript. BS and DK motivated the study, corrected and revised the manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]École Polytechnique Fédérale de Lausanne (EPFL), Route Cantonale, 1015 Lausanne, Switzerland. [2]Business Informatics Department, University of Applied Sciences, HES-SO, Western Switzerland Carouge, Switzerland. [3]Department of Philosophy and Department of Biomedical Informatics, 104 Park Hall, University at Buffalo, 14260 Buffalo, NY, USA.

Received: 1 March 2018 Accepted: 16 July 2018
Published online: 15 August 2018

References

1. Smith B, Kusnierczyk W, Schober D, Ceusters W. Towards a reference terminology for ontology research and development in the biomedical domain. vol. 2006. In: KR-MED 2006, Formal Biomedical Knowledge Representation, Proceedings of the Second International Workshop on Formal Biomedical Knowledge Representation: "Biomedical Ontology in Action" (KR-MED 2006), Collocated with the 4th International Conference on Formal Ontology in Information Systems (FOIS-2006), Baltimore, Maryland, USA, November 8, 2006; 2006. p. 57–66.
2. Faber Benítez P. The cognitive shift in terminology and specialized translation. MonTI. Monografías de Traducción e Interpretación. 2009;1: 107–134.
3. Zhang S, Bodenreider O. Experience in aligning anatomical ontologies. Int J Semant Web Inf Syst. 2007;3(2):1.
4. Lofi C. Measuring semantic similarity and relatedness with distributional and knowledge-based approaches. Database Soc Jpn (DBSJ) J. 2016;14(1):1–9.
5. Tversky A. Features of similarity. Psychol Rev. 1977;84(4):327.
6. Kiela D, Hill F, Clark S. Specializing word embeddings for similarity or relatedness. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics; 2015. p. 2044–8. https://doi.org/10.18653/v1/D15-1242. http://www.aclweb.org/anthology/D15-1242.
7. Shvaiko P, Euzenat J. Ontology matching: state of the art and future challenges. IEEE Trans Knowl Data Eng. 2013;25(1):158–76.
8. Bishop CM. Neural Networks for Pattern Recognition. Oxford: Oxford university press; 1995.
9. Sparck Jones K. A statistical interpretation of term specificity and its application in retrieval. J Doc. 1972;28(1):11–21.
10. Cheatham M, Hitzler P. String similarity metrics for ontology alignment. In: International Semantic Web Conference. Heidelberg: Springer; 2013. p. 294–309.
11. Mao M, Peng Y, Spring M. Ontology mapping: as a binary classification problem. Concurr Comput Pract Experience. 2011;23(9):1010–25.
12. Mao M, Peng Y, Spring M. Ontology mapping: as a binary classification problem. In: Fourth International Conference on Semantics, Knowledge and Grid, SKG '08, Beijing, China, December 3-5, 2008; 2008. p. 20–25. https://doi.org/10.1109/SKG.2008.101. https://doi.org//10.1109/SKG.2008.101. https://dblp.org/rec/bib/conf/skg/MaoPS08.
13. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res. 2011;12(Aug):2493–537.
14. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. CoRR. 2013;abs/1301.3781. http://arxiv.org/abs/1301.3781.
15. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States; 2013. p. 3111–9. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality. https://dblp.org/rec/bib/conf/nips/MikolovSCCD13.
16. Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics; 2014. p. 1532–43. http://www.aclweb.org/anthology/D14-1162.
17. Le Q, Mikolov T. Distributed representations of sentences and documents. vol. 32, no. 2. In: Xing EP, Jebara T, editors. Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research. Bejing: PMLR; 2014. p. 1188–96. http://proceedings.mlr.press/v32/le14.pdf. http://proceedings.mlr.press/v32/le14.html.
18. Harris ZS. Distributional structure. Word. 1954;10(2-3):146–62.
19. Hill F, Reichart R, Korhonen A. SimLex-999: evaluating semantic models with (genuine) similarity estimation. Comput Linguist. 2015;41(4):665–95. http://www.aclweb.org/anthology/J15-4004.
20. Faruqui M, Dodge J, Jauhar SK, Dyer C, Hovy E, Smith NA. Retrofitting word vectors to semantic lexicons. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver, Colorado: Association for Computational Linguistics; 2015. p. 1606–15. http://www.aclweb.org/anthology/N15-1184.
21. Mrkšić N, Ó Séaghdha D, Thomson B, Gašić M, Rojas-Barahona LM, Su P-H, Vandyke D, Wen T-H, Young S. Counter-fitting word vectors to linguistic constraints. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: Association for Computational Linguistics; 2016. p. 142–8. http://www.aclweb.org/anthology/N16-1018.
22. Wieting J, Bansal M, Gimpel K, Livescu K. Towards universal paraphrastic sentence embeddings. arXiv preprint arXiv:1511.08198. 2015.
23. Wieting J, Bansal M, Gimpel K, Livescu K, Roth D. From paraphrase database to compositional paraphrase model and back. Trans Assoc Comput Linguist. 2015;3:345–58.
24. Mitchell J, Lapata M. Vector-based models of semantic composition. In: Proceedings of ACL-08: HLT. Columbus: Association for Computational Linguistics; 2008. p. 236–44. http://www.aclweb.org/anthology/P08-1028.
25. Gale D, Shapley LS. College admissions and the stability of marriage. Am Math Mon. 1962;69(1):9–15.
26. Groß A, Pruski C, Rahm E. Evolution of biomedical ontologies and mappings: Overview of recent approaches. Comput Struct Biotechnol J. 2016;14:333–40.
27. Euzenat J, Shvaiko P. Ontology Matching, 2nd edn. Heidelberg (DE): Springer; 2013.
28. Baader F. The Description Logic Handbook: Theory, Implementation and Applications. New York: Cambridge University Press; 2003.
29. Jiménez-Ruiz E, Grau BC, Horrocks I, Berlanga R. Ontology integration using mappings: Towards getting the right logical consequences. In: European Semantic Web Conference. Heidelberg: Springer; 2009. p. 173–87.
30. Solimando A, Jiménez-Ruiz E, Guerrini G. Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In: International Semantic Web Conference. Switzerland: Springer International Publishing; 2014. p. 1–16.
31. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08. New York: ACM; 2008. p. 1096–103. http://doi.acm.org/10.1145/1390156.1390294. https://doi.org/10.1145/1390156.1390294.
32. Kenter T, Borisov A, de Rijke M. Siamese cbow: Optimizing word embeddings for sentence representations. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin: Association for Computational Linguistics; 2016. p. 941–51. http://www.aclweb.org/anthology/P16-1089.
33. Hill F, Cho K, Korhonen A. Learning distributed representations of sentences from unlabelled data. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: Association for Computational Linguistics; 2016. p. 1367–77. http://www.aclweb.org/anthology/N16-1162.
34. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. 2015.
35. Li Z, Hoiem D. Learning without forgetting. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016. Cham: Springer International Publishing; 2016. p. 614–629.

36. Chen M. Efficient vector representation for documents through corruption. CoRR. 2017;abs/1707.02377. http://arxiv.org/abs/1707.02377. http://dblp.uni-trier.de/rec/bib/journals/corr/Chen17aa.
37. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. Adv Neural Inf Process Syst. 2007;19:153.
38. Alain G, Bengio Y. What regularized auto-encoders learn from the data-generating distribution. J Mach Learn Res. 2014;15(1):3563–93.
39. McVitie D, Wilson LB. Stable marriage assignment for unequal sets. BIT Numer Math. 1970;10(3):295–309.
40. Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb Med Inform. 2008;35(128):44.
41. Wang C, Cao L, Zhou B. Medical synonym extraction with concept space models. In: Proceedings of the 24th International Conference on Artificial Intelligence. IJCAI'15. Buenos Aires: AAAI Press; 2015. p. 989–95. http://dl.acm.org/citation.cfm?id=2832249.2832386.
42. Sergienya I, Schütze H. Learning better embeddings for rare words using distributional representations. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics; 2015. p. 280–5. https://doi.org/10.18653/v1/D15-1033. http://www.aclweb.org/anthology/D15-1033.
43. Rosse C, Mejino JL. A reference ontology for biomedical informatics: the foundational model of anatomy. J Biomed Inform. 2003;36(6):478–500.
44. Noy NF, Musen MA, Mejino JL, Rosse C. Pushing the envelope: challenges in a frame-based representation of human anatomy. Data Knowl Eng. 2004;48(3):335–59.
45. Hayamizu TF, Mangan M, Corradi JP, Kadin JA, Ringwald M. The adult mouse anatomical dictionary: a tool for annotating and integrating data. Genome Biol. 2005;6(3):29.
46. De Coronado S, Haber MW, Sioutos N, Tuttle MS, Wright LW. NCI Thesaurus: using science-based terminology to integrate cancer research results. In: MEDINFO 2004 - Proceedings of the 11th World Congress on Medical Informatics, San Francisco, California, USA, September 7-11, 2004; 2004. p. 33–37. https://doi.org/10.3233/978-1-60750-949-3-33. https://doi.org/10.3233/978-1-60750-949-3-33. https://dblp.org/rec/bib/conf/medinfo/CoronadoHSTW04.
47. Donnelly K. Snomed-ct: The advanced terminology and coding system for ehealth. Stud Health Technol Inform. 2006;121:279.
48. Speer R, Havasi C. Representing general relational knowledge in ConceptNet 5. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012; 2012. p. 3679–86. http://www.lrec-conf.org/proceedings/lrec2012/summaries/1072.html. https://dblp.org/rec/bib/conf/lrec/SpeerH12.
49. Navigli R, Ponzetto SP. Babelnet: Building a very large multilingual semantic network. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala: Association for Computational Linguistics; 2010. p. 216–25. https://doi.org/http://www.aclweb.org/anthology/P10-1023.
50. Navigli R, Ponzetto SP. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif Intell. 2012;193:217–50.
51. Dakka W, Ipeirotis PG. Automatic extraction of useful facet hierarchies from text databases. In: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering. ICDE '08. Washington: IEEE Computer Society; 2008. p. 466–75. https://doi.org/10.1109/ICDE.2008.4497455. https://doi.orgg/10.1109/ICDE.2008.4497455.
52. Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S. Distributional semantics resources for biomedical text processing. In: Proceedings of the 5th International Symposium on Languages in Biology and Medicine (LBM); 2013. p. 39–44.
53. Kingma D, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014.
54. Zeiler MD. ADADELTA: an adaptive learning rate method. CoRR. 2012;abs/1212.5701. http://arxiv.org/abs/1212.5701. http://dblp.uni-trier.de/rec/bib/journals/corr/abs-1212-5701.
55. Bodenreider O, Hayamizu TF, Ringwald M, De Coronado S, Zhang S. Of mice and men: aligning mouse and human anatomies. In: AMIA 2005, American Medical Informatics Association Annual Symposium, Washington, DC, USA, October 22-26, 2005; 2005. p. 61. http://knowledge.amia.org/amia-55142-a2005a-1.613296/t-001-1.616182/f-001-1.616183/a-012-1.616655/a-013-1.616652. https://dblp.org/rec/bib/conf/amia/BodenreiderHRCZ05.
56. Bodenreider O. The unified medical language system (umls): integrating biomedical terminology. Nucleic Acids Res. 2004;32(suppl_1):267–70.
57. Jiménez-Ruiz E, Grau BC, Horrocks I, Berlanga R. Logic-based assessment of the compatibility of umls ontology sources. J Biomed Semant. 2011;2(1):2.
58. Achichi M, Cheatham M, Dragisic Z, Euzenat J, Faria D, Ferrara A, Flouris G, Fundulaki I, Harrow I, Ivanova V, Jiménez-Ruiz E, Kuss E, Lambrix P, Leopold H, Li H, Meilicke C, Montanelli S, Pesquita C, Saveta T, Shvaiko P, Splendiani A, Stuckenschmidt H, Todorov K, dos Santos CT, Zamazal O. Results of the ontology alignment evaluation initiative 2016. vol. 1766. In: Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. RWTH; 2016. p. 73–129.
59. Pesquita C, Faria D, Santos E, Couto FM. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In: Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013; 2013. p. 13–24. http://ceur-ws.org/Vol-1111/om2013_Tpaper2.pdf. https://dblp.org/rec/bib/conf/semweb/PesquitaFSC13.
60. Faria D, Pesquita C, Santos E, Palmonari M, Cruz IF, Couto FM. The AgreementMakerLight ontology matching system. vol. 8185 LNCS. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer; 2013. p. 527–541.
61. Faria D, Pesquita C, Santos E, Cruz IF, Couto FM. AgreementMakerLight 2.0: towards efficient large-scale ontology matching. In: Proceedings of the 2014 International Conference on Posters &#38; Demonstrations Track - Volume 1272. ISWC-PD'14. Aachen: CEUR-WS.org; 2014. p. 457–60. http://dl.acm.org/citation.cfm?id=2878453.2878568.
62. Jiménez-Ruiz E, Cuenca Grau B. LogMap: logic-based and scalable ontology matching. In: Aroyo L, Welty C, Alani H, Taylor J, Bernstein A, Kagal L, Noy N, Blomqvist E, editors. The Semantic Web – ISWC 2011. Berlin: Springer; 2011. p. 273–88.
63. Yeh A. More accurate tests for the statistical significance of result differences. In: Proceedings of the 18th Conference on Computational linguistics-Volume 2. COLING '00. Stroudsburg: Association for Computational Linguistics; 2000. p. 947–53. https://doi.org/10.3115/992730.992783. https://doi.org/10.3115/992730.992783.
64. Gulić M, Vrdoljak B, Banek M. Cromatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment. Web Semant Sci, Serv Agents World Wide Web. 2016;41:50–71.
65. Djeddi WE, Khadir MT. A novel approach using context-based measure for matching large scale ontologies. In: Bellatreche L, Mohania MK, editors. Data Warehousing and Knowledge Discovery. Cham: Springer International Publishing; 2014. p. 320–31.
66. Zhao M, Zhang S. Identifying and validating ontology mappings by formal concept analysis. In: Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016; 2016. p. 61–72. http://ceur-ws.org/Vol-1766/om2016_Tpaper6.pdf. https://dblp.org/rec/bib/conf/semweb/ZhaoZ16.
67. Zhao M, Zhang S, Li W, Chen G. Matching biomedical ontologies based on formal concept analysis. J Biomed Semant. 2018;9(1):11.
68. Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ordered Sets. Dordrecht: Springer; 1982. p. 445–470.
69. Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y. Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy). Austin; 2010.
70. Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow I, Bergeron A, Bouchard N, Warde-Farley D, Bengio Y. Theano: new features and speed improvements. arXiv preprint arXiv:1211.5590. 2012.
71. Le QV, Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY. On optimization methods for deep learning. ICML'11. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. USA: Omnipress; 2011. p. 265–72. http://dl.acm.org/citation.cfm?id=3104482.3104516.
72. Dean J, Corrado GS, Monga R, Chen K, Devin M, Le QV, Mao MZ, Ranzato M'A, Senior A, Tucker P, Yang K, Ng AY. Large scale distributed

deep networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. USA: Curran Associates Inc.; 2012. p. 1223–31. http://dl.acm.org/citation.cfm?id=2999134.2999271.
73. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker PA, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X. TensorFlow: a system for large-scale machine learning. vol. 16. In: 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016; 2016. p. 265–83.
74. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A. Automatic differentiation in pytorch. 2017.
75. Manne F, Naim Md, Lerring H, Halappanavar M. On stable marriages and greedy matchings. In: 2016 Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing. SIAM; 2016. p. 92–101.
76. Cruz IF, Fabiani A, Caimi F, Stroe C, Palmonari M. Automatic configuration selection using ontology matching task profiling. In: Simperl E, Cimiano P, Polleres A, Corcho O, Presutti V, editors. The Semantic Web: Research and Applications. Berlin: Springer; 2012. p. 179–94.
77. Otero-Cerdeira L, Rodríguez-Martínez FJ, Gómez-Rodríguez A. Ontology matching: A literature review. Expert Syst Appl. 2015;42(2):949–71.
78. Meilicke C. Alignment incoherence in ontology matching. 2011. https://ub-madoc.bib.uni-mannheim.de/29351.
79. Rodríguez-García MÁ, Gkoutos GV, Schofield PN, Hoehndorf R. Integrating phenotype ontologies with phenomenet. J Biomed Semant. 2017;8(1):58.
80. Gkoutos GV, Green EC, Mallon A-M, Hancock JM, Davidson D. Using ontologies to describe mouse phenotypes. Genome Biol. 2005;6(1):8.
81. Mungall CJ, Gkoutos GV, Smith CL, Haendel MA, Lewis SE, Ashburner M. Integrating phenotype ontologies across multiple species. Genome Biol. 2010;11(1):2.
82. Zhang Y, Wang X, Lai S, He S, Liu K, Zhao J, Lv X. Ontology matching with word embeddings. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Berlin: Springer; 2014. p. 34–45.
83. Xiang C, Jiang T, Chang B, Sui Z. Ersom: A structural ontology matching approach using automatically learned entity representation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics; 2015. p. 2419–29. http://aclweb.org/anthology/D15-1289.
84. Song S, Zhang X, Qin G. Multi-domain ontology mapping based on semantics. Clust Comput. 2017;20(4):3379–91.
85. Coates A, Lee H, Ng AY. An analysis of single-layer networks in unsupervised feature learning. Ann Arbor. 2010;1001(48109):2.
86. Pentina A, Ben-David S. Multi-task and lifelong learning of kernels. In: Chaudhuri K, Gentile C, Zilles S, editors. Algorithmic Learning Theory. Cham: Springer International Publishing; 2015. p. 194–208.
87. Socher R, Pennington J, Huang EH, Ng AY, Manning CD. Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A Meeting of SIGDAT, a Special Interest Group of The ACL; 2011. p. 151–61. https://doi.org/http://www.aclweb.org/anthology/D11-1014. https://doi.org/http://dblp.uni-trier.de/rec/bib/conf/emnlp/SocherPHNM11.
88. Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Baltimore: Association for Computational Linguistics; 2014. p. 655–665. http://www.aclweb.org/anthology/P14-1062.
89. Williams G, Baxter R, He H, Hawkins S, Gu L. A comparative study of RNN for outlier detection in data mining. In: Proceedings of the 2002 IEEE International Conference on Data Mining. ICDM '02. Washington: IEEE Computer Society; 2002. p. 709–12. http://dl.acm.org/citation.cfm?id=844380.844788.
90. Markou M, Singh S. Novelty detection: a review - part 2: neural network based approaches. Signal Proc. 2003;83(12):2499–521.
91. Chen J, Sathe S, Aggarwal C, Turaga D. Outlier detection with autoencoder ensembles. In: Proceedings of the 2017 SIAM International Conference on Data Mining. SIAM; 2017. p. 90–98.
92. Hawkins S, He H, Williams G, Baxter R. Outlier detection using replicator neural networks. In: Kambayashi Y, Winiwarter W, Arikawa M, editors. Data Warehousing and Knowledge Discovery. Berlin: Springer Berlin Heidelberg; 2002. p. 170–80.
93. Marchi E, Vesperini F, Eyben F, Squartini S, Schuller B. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks. In: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference On. IEEE; 2015. p. 1996–2000.
94. Xu D, Yan Y, Ricci E, Sebe N. Detecting anomalous events in videos by learning deep representations of appearance and motion. Comp Vision Image Underst. 2017;156(Supplement C):117–27. Image and Video Understanding in Big Data.
95. Arp R, Smith B, Spear AD. Building Ontologies with Basic Formal Ontology. Cambridge: The MIT Press; 2015.
96. Dragisic Z, Ivanova V, Li H, Lambrix P. Experiences from the anatomy track in the ontology alignment evaluation initiative. J Biomed Semant. 2017;8(1):56.