Expert Systems With Applications 204 (2022) 117446

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Mapping layperson medical terminology into the Human Phenotype Ontology using neural machine translation models

Enrico Manzini, Jon Garrido-Aguirre *, Jordi Fonollosa, Alexandre Perera-Lluna

B2SLab, Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Spain
Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), Madrid, Spain
Institut de Recerca Pediàtrica Hospital Sant Joan de Déu, Esplugues de Llobregat, Barcelona, Spain

ARTICLE INFO

Keywords:
Machine translation
Word embedding
Deep learning
Medical informatics
Deep phenotyping
Human Phenotype Ontology

ABSTRACT

In the medical domain there exists a terminological gap between patients and caregivers and the healthcare professionals. This gap may hinder the success of the communication between healthcare consumers and professionals in the field, with negative emotional and clinical consequences. In this work, we build a machine learning-based tool for the automatic translation between the terminology used by laypeople and that of the Human Phenotype Ontology (HPO). HPO is a structured vocabulary of phenotypic abnormalities found in human disease. Our method uses a vector space to represent an HPO-specific embedding as the output space for a neural network model trained on vector representations of layperson versions and other textual descriptors of medical terms. We explored different output embeddings coupled to different neural network architectures for the machine translation stage. We compute a similarity measure to evaluate the ability of the model to assign an HPO term to a layperson input. The best-performing models resulted with a similarity higher than 0.7 for more than 80% of the terms, with a median between 0.98 and 1. The translator model is made available in a web application at this link: https://hpotranslator.b2slab.upc.edu.

1. Introduction

In healthcare there is a gap between the language used by patients and caregivers (i.e. laypeople) and the medical terminology, or jargon, that may hinder the success of the communication between healthcare consumers and professionals in the field. The perception of medical jargon as confusing, ambiguous, or obsolete, may lead to frustration and cause distress in patients and caregivers (Tong et al., 2020). In the particular case of rare diseases, a rapid and effective communication between patients and clinicians is of utmost importance, because diagnosis is usually challenging, and accelerating diagnosis would have profound consequences for the treatment and management of rare diseases and important implications for the quality of life of patients and caregivers.[1] Those facts, together with the increasing availability of digital health tools such as smartphone health apps or health social networks in which self-reported data is part of the available information, provide an opportunity to design and implement solutions to bridge the terminological gap in healthcare.

The use of ontologies, in particular the Human Phenotype Ontology (HPO) (Köhler et al., 2018), to annotate patient clinical profiles allows clinicians and researchers to leverage a structured representation of the knowledge domain, enabling computation and machine-based inference over individual patient profiles and patient cohorts, that might be helpful to aid in diagnosis and prognosis. There are public resources for HPO annotation and HPO-aided inference — for example, Phenomizer,[2] Phenolyzer,[3] but they all require prior knowledge of clinical terminology and familiarity with biomedical ontologies, and thus are of limited use except by qualified professionals. Nevertheless, as already noted, in a context of increased use of digital health tools by patients and caregivers, the terminological gap between medical jargon and lay language might prevent making the most out of existing knowledge bases. The gap in HPO has been addressed in recent versions of the ontology by including layperson synonyms of HPO terms (Vasilevsky et al., 2018). However, although incorporated into the ontological representation of phenotypic abnormalities, annotation

* Corresponding author at: B2SLab, Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Spain.
E-mail addresses: enrico.manzini@upc.edu (E. Manzini), jon.garrido@upc.edu (J. Garrido-Aguirre), jordi.fonollosa.m@upc.edu (J. Fonollosa), alexandre.perera@upc.edu (A. Perera-Lluna).

[1] International Joint Recommendations to Address Specific Needs of Undiagnosed Rare Disease Patients, EURORDIS. 2016.
[2] Phenomizer.
[3] Phenolyzer.

https://doi.org/10.1016/j.eswa.2022.117446
Received 2 November 2020; Received in revised form 1 April 2022; Accepted 27 April 2022
Available online 6 May 2022

by patients would require matching a query to the lay terms available which, in practice, is not very different from the way in which resources such as patient vocabularies are used (see Section 2).

Driven by the need to reduce the gap between layperson terminology and medical jargon in order to promote patient and caregiver participation in healthcare and effective patient–clinician communication, this paper proposes an automatic method to translate layperson words and expressions into their corresponding HPO classes. We propose a method that makes use of neural network models to map layperson terminology into a semantic space representing HPO.

2. Related work

2.1. Consumer languages

The problem of translation between layperson language and technical jargon within the medical domain has been addressed in different ways. In the last twenty years, several efforts have been made to develop consumer health vocabularies (CHVs). Seminal work in the field of CHVs (Zielstorff, 2003) had already noted the mismatch between lay and professional language in the medical field as a handicap for accessing relevant information, sharing clinical data, and allowing effective patient–healthcare worker communication. CHVs are built beginning with the identification of lay terminology; this vocabulary is then mapped to professional terms contained in controlled vocabularies, such as the Unified Medical Language System (UMLS).[4] These dictionaries are validated through expert review. CHVs are based on static vocabularies, and so are limited by the fact that laypeople lexicon is richer than the technical language which, in contrast, is very precise. As put in Smith, Stavri, and Chapman (2002) "the notion of a paradigmatic consumer using a vocabulary specific to her consumer status may be ill-founded". It is thus infeasible to cover all the terms encountered in clinical situations in such dictionaries (Keselman et al., 2008). An alternative to dictionaries is to use pattern-based methods for text mining. In general, it has been demonstrated that counting the co-occurrence of word pairs and other contextual information extracted from large text corpora can be used for the identification of synonyms (Baroni & Siri, 2004; Hagiwara, Ogawa, & Toyama, 2006). In the biomedical domain, Vydiswaran, Mei, Hanauer, and Zheng (2014) applied a pattern-based method to a corpus based on Wikipedia to identify related lay and professional terms. They identified synonym pairs in texts and automatically labelled them as either consumer or professional terms. CHVs have been used to map medical concepts from electronic health records (EHRs) to layperson terminology to make them more accessible to end users (Zeng-Treitler, Goryachev, Kim, Keselman, & Rosendale, 2007).

2.2. Medical term representations

In the last decade, work on natural language modelling has shown that using word embedding techniques the components of natural language can be represented in continuous vector spaces that reproduce semantic rules (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). There are many methods available for representing words and sentences in a vector space, and several studies have attempted to outline the advantages and drawbacks of each, both for the general (Baroni, Dinu, & Kruszewski, 2014), and the biomedical domains (Pakhomov, Finley, McEwan, Wang, & Melton, 2016; Wang et al., 2018). The representation of biomedical terms using word embedding techniques has been used in many applications, including named entity recognition, synonym extraction, chemical-disease drug–drug interaction and protein–protein relation extraction, and abbreviation disambiguation (Wang et al., 2018). In the case of HPO-specific word embeddings Pilehvar and Collier (2016) used linear combinations of word embeddings related to each HPO term to obtain an HPO-specific embedding.

[4] https://www.nlm.nih.gov/research/umls/index.html.

2.3. Machine translation

In general, CHVs for sentence translation between domains entails some loss of the information given by the context in which a term appears. In an effort to solve this problem, Weng, Chung, and Szolovits (2019) proposed a translation machine for medical terms and sentences based on an unsupervised bilingual dictionary induction (BDI) algorithm. To our knowledge, this is the first attempt in the clinical context to automatically translate entire sentences between the professional and consumer domains. With the same goal, Luo et al. (2020) introduced MedLane, a human-annotated dataset to align medical terminology and layperson expressions. They used these data to train PMBERT-MT, a translation model built on PubMedBERT (Gu et al., 2022).

2.4. Use of ontologies in healthcare

The use of medical ontologies can improve healthcare delivery, from improving accuracy of diagnoses to building more interoperable information systems (Ivanović & Budimac, 2014). For this reason, in recent years some studies have been published focusing on translating free text (mainly from medical records and other domain specific texts) into specific terms in ontologies and other specialized vocabularies. Pérez, Gojenola, Casillas, Oronoz, and de Ilarraza (2015), for example, used a model based on Finite State Transducers (FTS) to automatically map diagnostic terms written by clinicians into terms of the 9th Revision of the International Classification of Diseases (ICD-9). More recently, Zhang et al. (2019) trained an autoencoder to classify text from EHRs into HPO terms, while Zhang et al. (2021) aimed to the same using a BERT based architecture.

There is, however, a lack of solutions for the automatic mapping of layperson expressions to technical medical terminology. In the present work we propose a novel method to automatically map short sentences and expressions into an ontology of phenotypic abnormalities. We aim for a solution that provides an automatic mapping of layperson terms into a structured space of medical jargon.

3. Materials and methods

We propose a method to translate from layperson terms and short sentences to HPO terms in which a vector space that represents HPO i.e. an HPO-specific embedding, is used as the output space for a neural network model trained on vector representations of layperson terms and other textual descriptors. The methodological framework can be found in Fig. 1.

3.1. The human phenotype ontology

Terms in HPO describe human phenotypic abnormalities (Köhler et al., 2018) and are identified by a unique identifier and a label —i.e. its name in medical jargon. Most of the terms contain a brief description provided by clinicians or external sources, and a list of synonyms. In addition, HPO terms are linked to other HPO classes and external terminologies and ontologies. HPO is divided into 5 sub-ontologies, namely: `Phenotypic abnormality` (the main sub-ontology, containing the description of the phenotypes included in HPO); `Mode of inheritance`; `Clinical modifier`; `Clinical course`; `Frequency`. Conveniently, `Phenotypic abnormality` is divided in 25 sub-categories –or branches– including phenotypes divided by human systems and anatomical structures.

3.2. Word embeddings

Here, we followed the nomenclature proposed by Sarma, Liang, and Sethares (2018) which defined three types of word embeddings

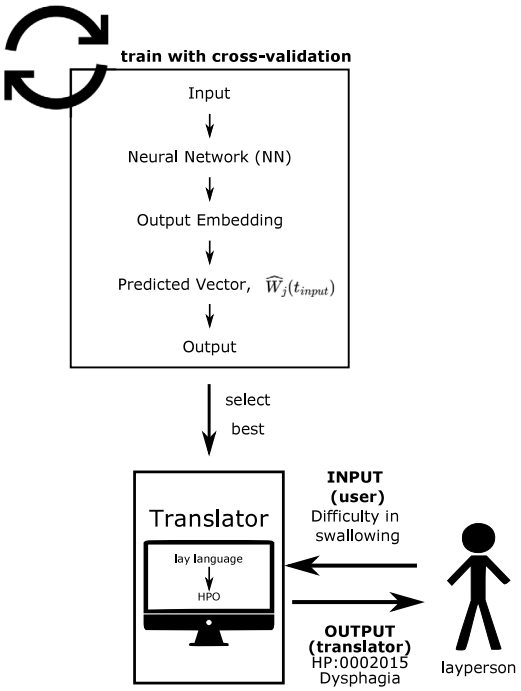

Fig. 1. Methodological framework. We use the Human Phenotype Ontology (HPO) to build a dataset of 30,000 words and sentences, each associated with an HPO term in the 'Phenotypic abnormality' subontology; this dataset is used to train the models with a cross-validation scheme in which the dataset is split in 29,400 samples for the training set and 600 samples for the test set. We tested different model and parameter configurations. The best model was then selected to build an automatic translator that can be used by patients and caregivers to map lay language to HPO terminology. The models consist of a neural network that is fed with a fixed-length input vector. The output from the neural network is mapped into an embedding space encoding semantic information about phenotypic abnormalities. The output from the translator model will be the closest vector in that space.

for specific knowledge domains: (1) a **generic embedding** trained on extensive corpora e.g. Wikipedia, PubMed® (2) a **domain-specific embedding** based on a specific corpus, (3) a **combined embedding** created using generic and domain-specific embeddings (e.g. Sarma et al., 2018). To better understand the behaviour of the embeddings we implement and test the three different strategies.

3.2.1. Generic embeddings

We use a word embedding pretrained on medical texts from MEDLINE®/PubMed®, provided by McDonald, Brokos, and Androutsopoulos (2018). Since most HPO terms are concepts described by a sentence or few words –e.g. *Atrial septal defect*, *Abnormal mitral valve morphology*–, the vectors of the word embedding should be combined to get a numeric representation of each HPO term. Let $|HPO|$ be the number of terms contained in HPO, $R_\tau(i) = (w_1, w_2, \dots, w_n)$, $i \in (1, \dots, |HPO|)$, the list of words (lemmatized, without neither stop words nor punctuation) included in the ith HPO term, $\mathbf{w}$ the vector representation of word w in the McDonald et al. (2018) word embedding, and $W_j(HPO_i)$ the new vector representation of the ith HPO term. We then define the following HPO term representations:

- W_{G1}, defined as the sum of the vectors in $R_\tau(i)$:

$$W_{G1}(HPO_i) = \sum_{w_j \in R_\tau(i)} \mathbf{w}_j$$

- W_{G2}, calculated with the sum of the vectors in $R_\tau(i)$ weighted by the *term frequency-inverse document frequency* index ($tfidf$) of each word (Salton & Buckley, 1988):

$$W_{G2}(HPO_i) = \sum_{w_j \in R_\tau(i)} \mathbf{w}_j \cdot tfidf(w_j)$$

- W_{G3}, defined as the sum of the vectors multiplied by a decay factor of the words in the extended sorted list R^*_t. Given:

$$f(w_j) = \begin{cases} 1 & if \ \ w_j \in R_\tau(i) \\ e^{-\lambda \cdot j} & if \ \ w_j \notin R_\tau(i) \end{cases},$$

with $\lambda = 0.2$ (Pilehvar & Collier, 2016), the embedding is defined as:

$$W_{G3} = \sum_{w_j \in R^*_\tau(i)} \mathbf{w}_j \cdot f(w_j)$$

R^*_τ is a list extending the words in R_τ with words linked to the HPO term extrapolated from Wikipedia and sorted based on how specific each term is (this is similar to Pilehvar and Collier, 2016).

3.2.2. Domain-specific embedding (W_{LSA})

We build the domain-specific word embedding via latent semantic analysis (LSA) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). We firstly create a collection of documents related to each HPO term extracting its name, description, parents, and synonyms. After removing stop words and punctuation, we concatenate the words in a unique document. Then we proceed to construct a matrix in which each row represents a document, and each column a word in the corpus. Each element in the matrix contains the tf–idf of each word, representing its importance in a document with respect to the entire corpus. In the last step, we reduce the dimensionality of the matrix via singular-value decomposition (SVD).

3.2.3. Combined (W_{SVD})

There are several strategies for combining word embeddings. In this work, we use a meta-embedding proposed by Yin and Schütze (2016), herein W_{SVD}. We first concatenate the vectors from W_{G1} and W_{LSA} into a unique higher dimensional vector. Then, as in the domain-specific embedding, we reduce the dimension of the word embedding through SVD.

In addition to the previous embeddings, a random word embedding (W_{rand}) is also created using a Gaussian mixture model to randomly produce embedding features based on W_{G1}.

3.3. Models

Once the HPO-specific embedding is obtained, the problem of mapping layperson phenotype terms and short textual descriptors into HPO terms can be modelled as a text classification problem. In this work, we explore three distinct architectures using different parameter configurations (see Fig. 2): an encoder model with a dense linear layer to map the output vector in the HPO space (LSTM-D), an encoder model applying a convolutional layer to the embedding vector (C-LSTM-D), and a model that combines these two approaches into a framework whose goal is to classify the input terms into the parent branches of the phenotypic abnormality node in the ontology (*"HP:0000118: Phenotypic abnormality"*), prior to classification (LSTM-P).

3.3.1. LSTM-D

This strategy is inspired in the classical encoder–decoder model used in machine translation (Sutskever, Vinyals, & Le, 2014), but it is modified at the output to map the result into the HPO embedding space. The architecture uses a long–short term memory layer (LSTM) (Hochreiter & Schmidhuber, 1997) for capturing long-range dependencies between the input words. This model takes a layperson term or short text (τ_{input}) coded as a vector $v \in \mathbb{R}^{l_{LAY}}$ of length l_{LAY} as input. Then a word embedding for the input space is created, and trained with the model. This word embedding is different from the HPO-specific embedding at the output. This way, given N_W the dimension of W, we obtain a matrix $\mathbf{M} \in \mathbb{R}^{l_{LAY} \times N_W}$ embedding the input text. The output of the LSTM layer is a vector of length N_{LSTM}. This vector is mapped with a linear fully-connected layer into the HPO-specific embedding space, obtaining a

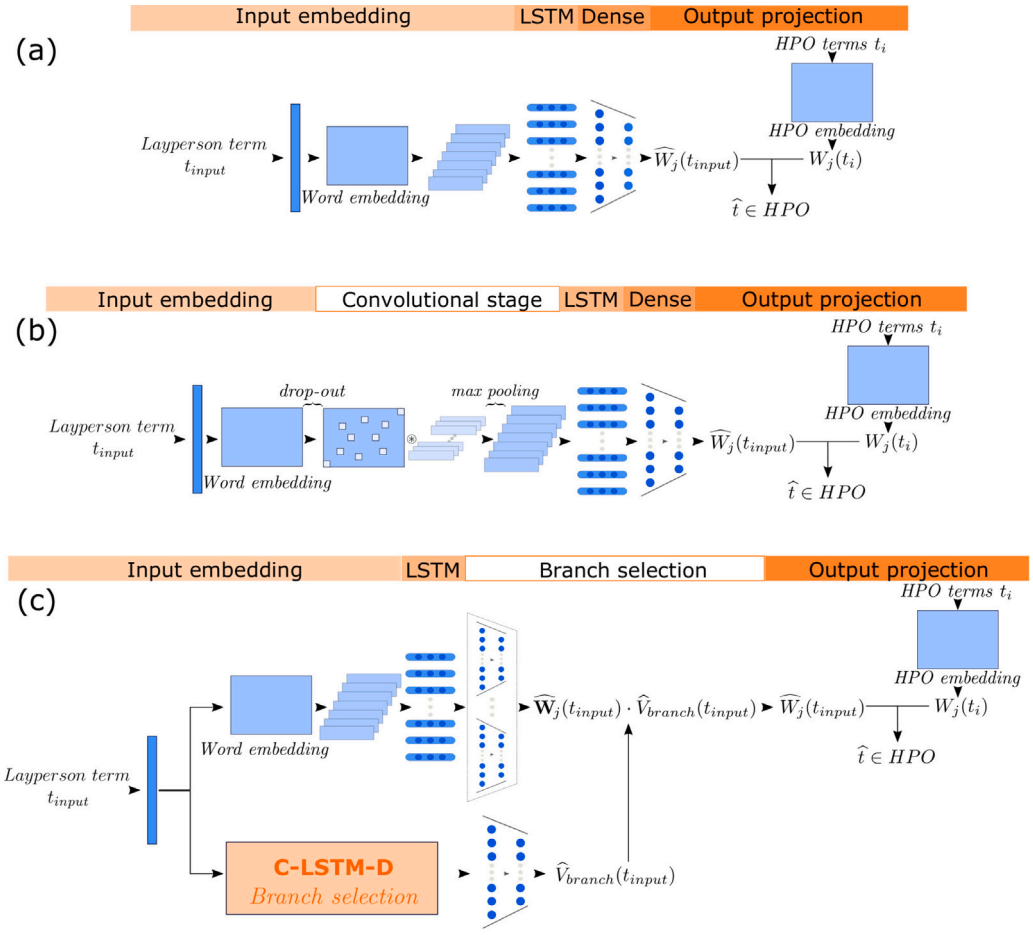

Fig. 2. Three different model architectures were implemented and tested, namely LSTM-D, C-LSTM, and LSTM-P. (a) *LSTM-D* is based in the classical encoder–decoder model used in machine translation, modified at the output to map the result in the HPO embedding space; (b) *C-LSTM-D* involves two steps: (1) extracting features from the input text using a convolutional layer, and (2) capturing long-range dependencies in the feature maps with a LSTM layer; (c) *LSTM-P* may benefit from the semantic structure of the HPO-specific embedding at the output, in which terms are mapped in subspaces that approximately represent the categories in the `Phenotypic abnormality` sub-ontology. This model performs two tasks in parallel: (1) branch selection, to reduce the output space to a subregion of the HPO embedding, and (2) layperson translation incorporating the information about the predicted branch to map the term in that subregion.

predicted vector $\widehat{W}_j(\tau_{input})$, $j \in \{LSA, G1, G2, G3, SVD\}$. Finally, the prediction $\hat{\tau}$ is chosen as the closest HPO vector:

$$\hat{\tau} = \min_{i=1,\dots,|HPO|} dist\left(\widehat{W}_j(\tau_{input}), W_j(\tau_i)\right),$$

where $dist(a, b)$ is the Euclidean distance between vectors a and b.

3.3.2. C-LSTM-D

Based on a model proposed by Zhou, Sun, Liu, and Lau (2015), C-LSTM-D contains two steps: (1) exploring a convolutional layer to get features from the input text, and (2) capturing long-range dependencies in the feature maps using a LSTM layer. The output of the input word embedding is passed through a convolutional layer with ReLU units ($N_{filters} = 600$, $filter\,dim. = 5$) to extract features in the embedded representation with a max-pooling block that reduces the dimensionality of the feature space ($max\ pool = 15$). This step produces a matrix that is used as the input for an LSTM layer and the output is computed as in LSTM-D.

3.3.3. LSTM-P

This model may benefit from the semantic structure of the HPO-specific embedding at the output, in which terms are mapped in subspaces that approximately represent the categories in the `Phenotypic abnormality` sub-ontology. This model performs two tasks in parallel: (a) branch selection, to reduce the output space to a subregion of the HPO embedding, and (b) layperson translation incorporating the information about the predicted branch to map the term in that subregion. The branch detector and C-LSTM-D share some similarities: However, there are two dense layers, one that predicts the HPO embedding from the LSTM as in C-LSTM-D, and the second that predicts the branches, mapping $\widehat{W}_j(\tau_{input})$ to a one-hot vector in $\mathbb{R}^{26}$, $\hat{V}_{branch}(\tau_{input})$ representing the different categories below `Phenotypic abnormality` and the other sub-ontologies grouped as a single category (`Frequency`, `Mode of inheritance`, `Clinical course` and `Clinical modifier`). The second task (the actual translation) is accomplished by a part of the model that works similarly to LSTM-D, but after the LSTM layer there is not a single dense layer but 26 layers working in parallel, each one with the specific task of mapping the input vector to a specific region of the embedding space that represents the whole HPO. Hence, given d being the size of $\widehat{W}_j$, this last layer produces a matrix $\widehat{\mathbf{W}}_j(\tau_{input}) \in \mathbb{R}^{26\times d}$. Each row of this matrix is a different prediction given the input, according to the relevant branch. The last step is to multiply the outputs of the two tasks to get the actual prediction:

$$\widehat{W}_j(\tau_{input}) = \widehat{\mathbf{W}}_j(\tau_{input}) \cdot \hat{V}_{branch}(\tau_{input}).$$

The prediction $\hat{t}$ is chosen as in the previous models.

3.4. Model evaluation

In order to evaluate the performance of the different models, we use the Jiang and Conrath (1997) similarity function modified for this

Table 1
Variables in the explanatory model.

	Architectures		LSTM	Word embedding					Compression			
	LSTM-D	LSTM-P		$G1$	$G2$	$G3$	LSA	SVD	1	2/3	1/2	1/3
Regressors	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9			

specific ontology as proposed by Seco, Veale, and Hayes (2004). The information content of a term τ_i is defined as:

$$ic(\tau_i) = 1 - \frac{log\left(hypo(\tau_i) + 1\right)}{log(|HPO|)},$$

with $hypo(\tau_i)$ being the number of parent (i.e. more specific) terms τ_i has. The Resnik similarity function (Resnik, 1995) of two terms $\tau_1, \tau_2 \in HPO$ is defined as:

$$sim_{res}(\tau_1, \tau_2) = \max_{\tau \in S(\tau_1, \tau_2)} ic(\tau).$$

where $S(\tau_1, \tau_2)$ is the set of terms that subsume τ_1 and τ_2. Then the similarity function used by Seco et al. (2004) is:

$$sim(\tau_1, \tau_2) = 1 - \frac{1}{2}\left(ic(\tau_1) + ic(\tau_2) - 2sim_{res}(\tau_1, \tau_2)\right),$$

4. Experiments

Terms in HPO are often enriched with a list of synonyms and with a brief description of one or more sentences. In the HPO release used here,[5] the ontology contains more than 17,500 synonym terms, 45% of which are classified as "layperson term" and are linked to 35% of the HPO terms. Despite 43% of the terms not having a list of synonyms, almost 90% of them are represented in the train set by means of the describing sentences, the synonyms, or both.

By retaining only sentences shorter than 55 words and removing longer ones, we split the descriptions into sentences. LSTM can accept variable length inputs. However, this is not the case for convolutional layers, whose inputs should always have the same dimensionality. Since more than 99% of the sentences in our input sets are shorter than 55 words, and the remaining ones are considerably longer (> 60 words), we choose this threshold as cut-off number in order to reduce sparsity in the input. For each sentence or synonym, we convert the numbers to strings (e.g. "1st" to "first", "4-layered" to "four-layered") and remove punctuation and stop words. Sentences and layperson terms are then mapped to $\mathbb{R}^{54}$ using a tokenizer based on word frequencies. Zero-padding is applied when needed. In this way, the first layer of each model is always fed with fixed length vectors. After these preprocessing steps, we obtain a list of more than 30,000 words and sentences, each associated with an HPO term and represented by a fixed length vector that we use as inputs to the models in the training and test sets.

For the generic and domain specific embeddings we test vectors of dimension 400 and 200 (indicated by A and B, respectively). For W_{SVD}, we test two possible combinations: (1) v_1, of dimension 200, obtained with 300-dimensional W_{LSA} and W_{G1} of dimension 200, and (2) v_2 of dimension 400, obtained with 600-dimensional W_{LSA} and 200-dimensional W_{G1}.

The models are trained using a cross-validation scheme, with 29,400 terms for the training phase and 600 for the testing phase, at each iteration. Keras (Chollet et al., 2015) was used to build and train the models, using mean squared error as loss function and the Adam optimization algorithm (Kingma & Ba, 2014).

5. Results

We fit a logistic regression as an explanatory model using R and the stats package (R Core Team, 2018) to assess the impact of different model configurations in the median output similarity, $sim_{res}(\tau_1, \tau_2)$, as follows:

$$log\left(\frac{y}{1-y}\right) = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_9 \cdot x_9,$$

with $y = sim_{res}(\tau_1, \tau_2)$. The regressors x_i are designed to reflect the following aspects of the models described in Section 3.3 (see Table 1): disposition of layers and structure of the models i.e. model architecture ($x_1, x_2 \in \{0, 1\}$); LSTM dimensionality ($x_3 \in \{0, 1\}$, 0 corresponds to 400 units, 1 to 600 units); type of output word embedding ($x_4, x_5, x_6, x_7, x_8 \in \{0, 1\}$); output–input size ratio i.e. compression at the output ($x_9 \in \{1/3, 1/2, 2/3, 1\}$). The null model was selected as the worst performing hyperparameter configuration such that changing the hyperparameters had a positive impact on the performance of the model and the results were easier to interpret. The configuration of the null model is as follows:

$$x_i = 0 \ \forall \ i \in \{1, \ldots, 8\},$$

and

$$x_9 = 1.$$

Thus, the null model is C-LSTM-D with a random output word embedding, a 400-dimensional LSTM layer, and no difference between input and output size. The results are presented in Table 2. The intercept represents the null model, and each column represents setting the corresponding hyperparameter of the null model to that value. The estimates represent the value of the logit function, y is the estimate transformed to $sim_{res}(\tau_1, \tau_2)$, and p-value indicates the significance of the effect of each of the variables in the output of the model. The results of the explanatory model highlight the relative relevance of the word embeddings in the performance. In general models show similar performance, even if the results for LSTM-P combined with the random output space suggest that this architecture contributes the most to the correct translation of layperson terms, although its effect is not statistically significant ($p = 0.37$). In addition, the table shows that the regressors with stronger effect on the performance are the word embeddings, overshadowing the impact of any other hyperparameter. Furthermore, higher input–output size ratio is beneficial for model performance, although this seems to have a negligible impact.

The results and analysis presented in the following are obtained with a fixed configuration of the LSTM-P model. The independent variables are the different word embeddings used as the output embedding spaces. The statistical analysis is performed using Python 3.7 (Van Rossum & Drake, 2009) and the scipy package (Virtanen et al., 2020).

The main results are reported in Table 3 for a specific combination of parameters and in Tables 1, 2, and 3 in the Supplementary Material. Here, sim indicates the similarity between the predicted and the true HPO term ($sim = 1$ means exact prediction, see Table 4 for an example of predictions in validation). In general, the model performance for different embeddings is similar, although four of them show slightly better results overall ($G1$ and SVD word embeddings, with different dimensions), with $median(sim) = 1$ and more than 80% of the terms identified with high similarity ($sim > 0.7$).

Using the random output embedding (W_{rand}) the models could identify the exact HPO term or get close in the semantic space in more than 50% of the cases. This shows that, although the output space has a random structure, the architecture of the model is contributing relevant features of the input space, as observed in the explanatory model. In addition, a suitable representation for HPO is crucial in improving model performance, as described by the explanatory model and shown in Fig. 2: Whereas there is no significant difference between the results using different neural network configurations, the analysis shows that the use of an structured output word embedding (e.g. W_{G1}) instead of W_{rand} is beneficial for model performance (Mann–Whitney U Test, $\alpha = 0.05$).

[5] 2018-12-21.

Table 2
Results of the explanatory model. Null model (intercept): C-LSTM-D, 400-unit LSTM, random word embedding, no compression.

	Null model	Architectures		LSTM	Word embedding					Compression		
		LSTM-D	LSTM-P		$G1$	$G2$	$G3$	LSA	SVD	2/3	1/2	1/3
Estimate	0.19	0.48	0.68	0.19	2.78**	1.98**	1.48*	2.48**	2.96**	0.08	0.11	0.22
y	0.55	0.66	0.70	0.59	0.95**	0.90**	0.84*	0.94**	0.96**	0.57	0.57	0.60

*Indicate the level of significance for the variables ($\alpha = 0.05$); $0.05 < p \leq 0.1$.
**Indicate the level of significance for the variables ($\alpha = 0.05$); $0.01 < p \leq 0.05$;

Table 3
Results for different word embeddings for model LSTM-P with 600-unit LSTM, no compression. *(e: $sim = 1$, s: $0.7 < sim < 1$, w: $sim \leq 0.7$).*

W	e, exact (%)	s, similar (%)	w, wrong (%)	$e \cup s$ (%)	median(sim)
$LSA\ A$	47.06	33.63	19.31	80.69	0.964
LSA B	49.91	32.92	17.17	**82.83**	0.986
G1 A	50.83	30.28	18.89	81.11	**1.0**
G1 B	50.29	30.35	19.36	80.64	**1.0**
$G2\ A$	43.42	37.02	19.56	80.44	0.951
$G2\ B$	42.73	37.28	19.99	80.01	0.949
$G3\ A$	35.44	41.96	22.6	77.4	0.909
$G3\ B$	37.06	41.66	21.28	78.72	0.918
SVD v1	**51.45**	30.38	18.17	81.83	**1.0**
SVD v2	51.28	30.28	18.44	81.56	**1.0**
W_{rand}	22.97	31.08	45.95	54.05	0.779
Random choice	$7 \cdot 10^{-3}$	0.88	99.1	0.89	0.31

Table 4
Example of predictions in validation. The similarity is computed between the top predicted term and the true term.

Input	Top prediction	True term	Similarity
Absent kidney on one side	Unilateral renal agenesis	*Unilateral renal agenesis*	1
Cystic abnormalities of the ovaries	Abnormality of the ovary	*Ovarian cyst*	0.95
Urgency frequency syndrome	Bradyphrenia	*Urinary urgency*	0.32

The distribution of the similarity between predicted and true HPO terms shows a similar behaviour for the best embeddings ($G1\ A$, $G1\ B$, $SVD\ v1$, and $SVD\ v2$). There are no major differences in median similarity, neither significant differences on the proportions of exact (e, $sim = 1$), similar (s, $0.7 < sim < 1$), and wrong (w, $sim \leq 0.7$) prediction bins (see Tables 4–7 in Supplementary Material). The only exception is for $G1\ B$ vs $SVD\ v1$ in the case of the w similarity bin ($p_w = 0.018$) and in the distribution of the similarities ($p_s = 0.03$, see Table 8 and Figure 1 in Supplementary Material); the difference is not significant in the e bin and is essentially negligible in the s bin ($p_e = 0.072$; $p_s = 0.978$). Together, these results suggest that the tail of the distribution of output similarities is thicker for $G1\ B$ than for $SVD\ v1$ but the mappings at the output that could be considered good enough ($0.7 < sim < 1$) are better overall for $G1\ B$ than for $SVD\ v1$. On the contrary, there are more differences on the proportions of the e, s, and w bins between $LSA\ B$ and both $SVD\ v1$ and $SVD\ v2$ ($p_e = 0.017$, $p_s = 2.2 \cdot 10^{-5}$, $p_w = 0.043$; $p_e = 0.034$, $p_s = 1.1 \cdot 10^{-5}$, $p_w = 0.011$). The differences are more significant for contrasts between $LSA\ B$ and both $G1\ A$ and $G1\ B$ when comparing the s and w bins ($p_s = 4.1 \cdot 10^{-6}$, $p_w = 3.1 \cdot 10^{-4}$; $p_s = 1.8 \cdot 10^{-5}$, $p_w = 1.1 \cdot 10^{-5}$). These results suggest that $LSA\ B$ has significantly fewer wrongly predicted terms in comparison with other output mapping spaces which have a higher number of exactly predicted terms. However, the distribution of output similarities in each bin is not sufficiently different (see Tables 8, 9 in Supplementary Material), except for $LSA\ B$ vs $G1\ B$ in the s bin in which, overall, $LSA\ B$ produces significantly worse results ($p_s = 0.003$, see Figure 2 in Supplementary Material).

The results presented so far epitomize the compromise between accuracy and precision in evaluating the performance of machine learning models. In the case at hand, when comparing $LSA\ B$ with $SVD\ v1$ and $SVD\ v2$ we can consider the latter, which are among the best performing models in terms of median similarity (together with $G1\ A$, $G1\ B$) to be most accurate in relative terms, as they contain a significantly higher proportion of results in the e bin. However, being less accurate, $LSA\ B$ is, at the same time, more precise than $SVD\ v1$ and $SVD\ v2$ in the $e \cup s$ set (i.e. contains a higher mass) where the exact matches and sufficiently good translations lie, as compared with the w bin, which includes the unacceptable results.

At this point, we note that the simplest generic embeddings ($G1\ A$, $G1\ B$), built from a sum of vectors without weighting, provide a more structured space than their weighted counterparts ($G2$, $G3$). In addition, the combination with the domain specific embedding to create $SVD\ v1$ and $SVD\ v2$ seems to be slightly beneficial for model performance, as the accuracy increases with marginal losses in the s bin, improving the precision in the $e \cup s$ bin (see Fig. 3).

In general, the more specific the term, the better the model performance (Fig. 4). Since the more specific a term is, the more unlikely a patient will be familiar with it, it is thus a promising behaviour that the models work better with these terms. Moreover, the specific terms are not overrepresented in the training set, i.e. the number of synonyms or descriptive sentences per term remain almost constant with increasing depth (see Figure 3 in Supplementary Material). At the same time, entries in the training set for specific terms do not contain more, or significantly different, words with respect to more generic terms in HPO ($p = 0.1282$; Chi-squared test, $\alpha = 0.05$), in general (see Figure 4 in Supplementary Material). Hence, since direct projections on word embeddings behave in opposite way (see Figure 5 in Supplementary Material), one may presume that these results are an improvement due to the models' effects.

Finally, we provide some examples of the potential of the model to bridge the terminological gap in the medical language using kidney health and related terminology as an example, and qualitatively explore some aspects of the output semantic space. According to a recent study, "chronic" and "acute" were considered very "medicalized" and obscure in kidney health communication (Tong et al., 2020). Trying to overcome this hindrance we explored input text including lay expressions that could provide mappings to two important kidney disorders, namely, chronic kidney disease and acute kidney injury. The

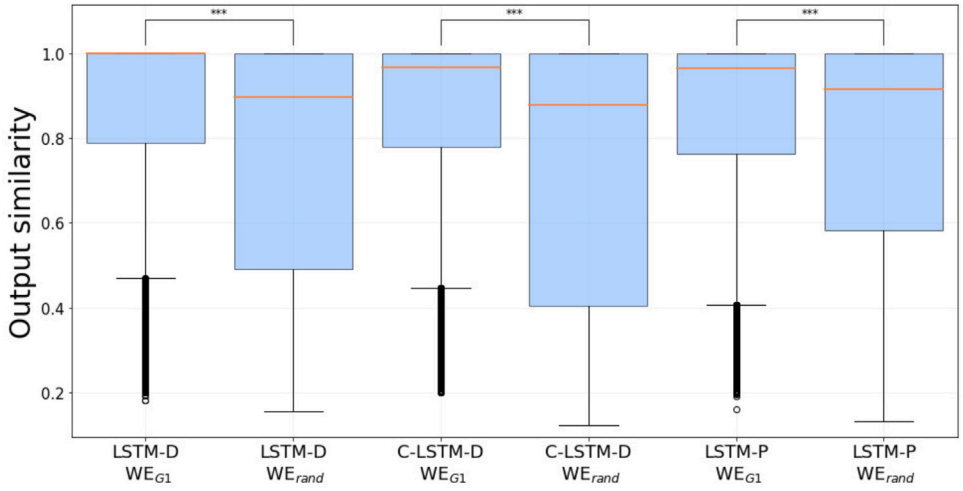

Fig. 3. Output similarity using W_{G1} output embedding with different architectures. There are no significant differences between different architectures; there are differences between the use of W_{G1} and W_{rand} (Mann–Whitney U Test, $\alpha = 0.05$).

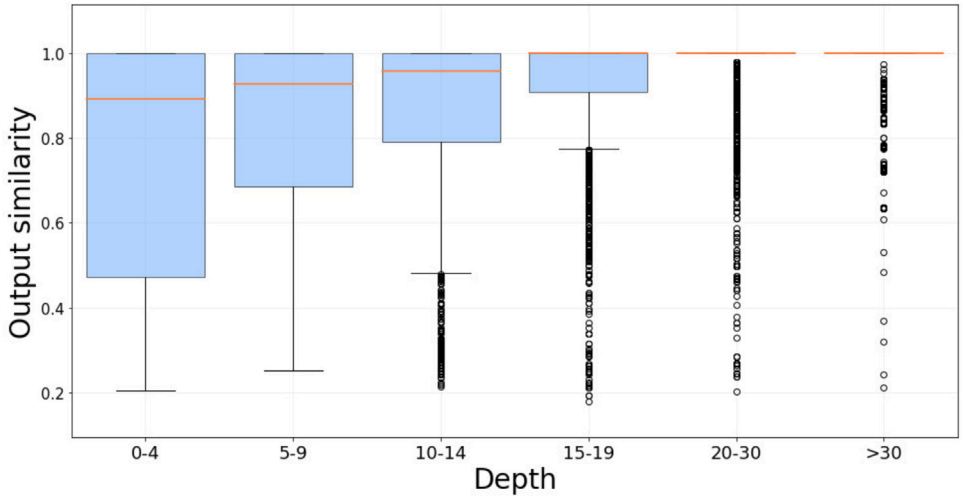

Fig. 4. Output similarity by depth level bins using W_{G1} output embedding and the LSTM-D model. Numbers in the x-axis indicate the length of the path between the terms in the bin and the phenotypic abnormality node in the ontology (*"HP:0000118: Phenotypic abnormality"*).

sentence "long lasting kidney failure" provides the HPO term "Chronic kidney disease" as the first option in the output. On the other side, "sudden kidney failure" provides the HPO term "Acute kidney injury" as the first option. Checking the data associated to those terms in HPO allows to interpret the results. In the case of "Chronic kidney disease" (see Figure 6 in Supplementary Material), none of the words in "long lasting" appear in the associated data: This fact suggests that the output space may have created a semantic region in which "long lasting" and "persisting for at least three months", or perhaps "progressive", are close between each other. In the case of "Acute kidney injury" (see Figure 7 in Supplementary Material), the word "sudden" appears in the description of the term, and it likely makes the connection with "sudden kidney failure" (which is none of the synonyms for the term) easier.

Furthermore, "renal" and "kidney" are used to described kidney health, but the term "renal" may be unfamiliar to patients and the public, preventing awareness and advocacy. In this regard, our model is able to make the "kidney" to "renal" mapping. For instance, providing "small kidney" as input to the model, it returns four potential candidates in the top most similar HPO terms, namely: "Renal hypoplasia", "Renal agenesis", "Renal insufficiency", and "Renal dysplasia". The terms "kidney" and "renal" are clearly overlapping in the semantic space, as both terms appear interchangeably in the associated data (see Figures 8–11 in Supplementary Material). In addition, "hypoplasia" and "small" appear to be close in the semantic space as well. In particular, it seems that "small"/"hypoplasia" is semantically close to "aplasia"/"agenesis"/"absence", "insufficiency"/"failure", and "dysplasia"/"adysplasia"/"dysplastic". However, the latter are not as close between them as it seems they are to "small"/"hypoplasia".

To conclude, limitations of this work must be highlighted. On the first place, we have explored a limited subset of the parameter space; we should assess the influence of each parameter on the outcome. Secondly, the test and validation sets described in Section 4 were built only with synonyms and short sentences inside HPO. This is a limiting aspect from two points of view: first, regarding the volume of data, limited to the content in HPO, and second, for the fact that definitions and synonyms associated to a term inside the ontology could not represent the extent of lay terminology. Finally, if words introduced to the translator are not in the training subset they will not be represented in the semantic space and therefore the results will not differ from entering random words, outside of the clinical realm, to the model.

6. Conclusions

In this work, we described a novel method for predicting specific phenotypes from generic text. We first created a vector representation

for HPO, and then mapped sentences in this space through a modified neural machine translation model. We tested different embeddings for HPO and evaluated our solution using the synonyms and descriptions present in HPO. The models showed similar performances with different embeddings. Although the model is central for the prediction of HPO terms, the choice of an embedding, irrespective of its dimensions, also had an impact. In the future, we expect to improve the model by exploring new configurations as well as different input and output spaces. We also plan to explore the performance of the model at different topological dimensions in the ontology. In addition, we would be interested in exploring the output semantic space further. The translator model is made available in a web application at this link: https://hpotranslator.b2slab.upc.edu. In the future we plan to improve this web application to make it more user-friendly and to help patients in the definition of their phenotypic profile through self-declaration.

CRediT authorship contribution statement

Enrico Manzini: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Jon Garrido-Aguirre:** Conceptualization, Formal analysis, Methodology, Validation, Writing – original draft, Writing – review & editing. **Jordi Fonollosa:** Supervision, Writing – review & editing. **Alexandre Perera-Lluna:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Spanish Ministry of Economy and Competitiveness (www.mineco.gob.es) TEC2014-60337-R, DPI2017-89827-R, Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), initiatives of Instituto de Investigación Carlos III (ISCIII), and Share4Rare project (Grant Agreement 780262). This work was partially funded by ACCIÓ (Innotec ACE014/20/000018). B2SLab is certified as 2017 SGR 952. The authors thank the NVIDIA Corporation for the donation of a Titan Xp GPU used to run the models presented in this article. J. Fonollosa acknowledges the support from the Serra Húnter program.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2022.117446.

References

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 238–247). Baltimore, Maryland: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-1023.

Baroni, M., & Siri, S. (2004). Using cooccurrence statistics and the web to discover synonyms in a technical language. In *(LREC'04), Proceedings of the fourth international conference on language resources and evaluation*.

Chollet, F., et al. (2015). Keras. https://keras.io.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407. http://dx.doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., et al. (2022). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, *3*(1), 1–23.

Hagiwara, M., Ogawa, Y., & Toyama, K. (2006). Selection of effective contextual information for automatic synonym acquisition. In *(ACL-44), Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics* (pp. 353–360). http://dx.doi.org/10.3115/1220175.1220220.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Ivanović, M., & Budimac, Z. (2014). An overview of ontologies and data resources in medical domains. *Expert Systems with Applications*, *41*, 5158–5166.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In K. Chen, C. Huang, & R. Sproat (Eds.), *Proceedings of the 10th research on computational linguistics international conference, ROCLING 1997, Taipei, Taiwan, August 1997* (pp. 19–33). The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Keselman, A., Smith, C., Divita, G., Kim, H., Browne, A., Leroy, G., et al. (2008). Consumer health concepts that do not map to the UMLS: Where do they fit? *Journal of the American Medical Informatics Association : JAMIA*, *15*, 496–505. http://dx.doi.org/10.1197/jamia.M2599.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980.

Köhler, S., Carmody, L., Vasilevsky, N., Jacobsen, J. O. B., Danis, D., Gourdine, J.-P., et al. (2018). Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources. *Nucleic Acids Research*, *47*(D1), D1018–D1027. http://dx.doi.org/10.1093/nar/gky1105.

Luo, J., Zheng, Z., Ye, H., Ye, M., Wang, Y., You, Q., et al. (2020). A benchmark dataset for understandable medical language translation. ArXiv, abs/2012.02420.

McDonald, R., Brokos, G., & Androutsopoulos, I. (2018). Deep relevance ranking using enhanced document-query interactions. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1849–1860). Brussels, Belgium: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D18-1211.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS'13, Proceedings of the 26th international conference on neural information processing systems - Volume 2* (pp. 3111–3119). Red Hook, NY, USA: Curran Associates Inc..

Pakhomov, S. V., Finley, G., McEwan, R., Wang, Y., & Melton, G. B. (2016). Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, *32*(23), 3635–3644. http://dx.doi.org/10.1093/bioinformatics/btw529.

Pérez, A., Gojenola, K., Casillas, A., Oronoz, M., & de Ilarraza, A. D. (2015). Computer aided classification of diagnostic terms in spanish. *Expert Systems with Applications*, *42*, 2949–2958.

Pilehvar, M. T., & Collier, N. (2016). Improved semantic representation for domain-specific entities. In *Proceedings of the 15th workshop on biomedical natural language processing* (pp. 12–16). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/W16-2902.

R Core Team (2018). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI'95, Proceedings of the 14th international joint conference on artificial intelligence - Volume 1* (pp. 448–453). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. http://dx.doi.org/10.1016/0306-4573(88)90021-0.

Sarma, P. K., Liang, Y., & Sethares, B. (2018). Domain adapted word embeddings for improved sentiment classification. In *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 2: Short papers)* (pp. 37–42). Melbourne, Australia: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P18-2007.

Seco, N., Veale, T., & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in WordNet. In *ECAI'04, Proceedings of the 16th European conference on artificial intelligence* (pp. 1089–1090). NLD: IOS Press.

Smith, C. A., Stavri, P. Z., & Chapman, W. W. (2002). In their own words? A terminological analysis of e-mail to a cancer information service. In *Proceedings / AMIA ... annual symposium. AMIA symposium*.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *NIPS'14, Proceedings of the 27th international conference on neural information processing systems - Volume 2* (pp. 3104–3112). Cambridge, MA, USA: MIT Press.

Tong, A., Levey, A. S., Eckardt, K.-U., Anumudu, S., Arce, C. M., Baumgart, A., et al. (2020). Patient and caregiver perspectives on terms used to describe kidney health. *Clinical Journal of the American Society of Nephrology*, *15*(7), 937–948.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Vasilevsky, N., Foster, E., Engelstad, M., Carmody, L., Might, M., Chambers, E., et al. (2018). Plain-language medical vocabulary for precision diagnosis. *Nature Genetics*, *50*, 474–476. http://dx.doi.org/10.1038/s41588-018-0096-x.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al., SciPy 1. 0 Contributors (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272.

Vydiswaran, V., Mei, Q., Hanauer, D. A., & Zheng, K. (2014). Mining consumer health vocabulary from community-generated text. In *Proceedings of the American medical informatics association annual symposium (AMIA)*.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., et al. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*, *87*, 12–20. http://dx.doi.org/10.1016/j.jbi.2018.09.008.

Weng, W.-H., Chung, Y.-A., & Szolovits, P. (2019). Unsupervised clinical language translation. In *KDD '19*, *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 3121–3131). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3292500.3330710.

Yin, W., & Schütze, H. (2016). Learning word meta-embeddings. In *Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 1351–1360). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P16-1128.

Zeng-Treitler, Q., Goryachev, S., Kim, H., Keselman, A., & Rosendale, D. (2007). Making texts in electronic health records comprehensible to consumers: A prototype translator. In *AMIA ... Annual symposium proceedings / AMIA symposium. AMIA symposium, Vol. 11* (pp. 846–850).

Zhang, J., Bolanos, L., Li, T. S., Tanwar, A., Freire, G., Yang, X., et al. (2021). Self-supervised detection of contextual synonyms in a multi-class setting: Phenotype annotation use case. In *EMNLP*.

Zhang, J., Zhang, X., Sun, K., Yang, X., Dai, C., & Guo, Y. (2019). Unsupervised annotation of phenotypic abnormalities via semantic latent representations on electronic health records. In *2019 IEEE international conference on bioinformatics and biomedicine (BIBM)* (pp. 598–603).

Zhou, C., Sun, C., Liu, Z., & Lau, F. C. M. (2015). A C-LSTM neural network for text classification. https://arxiv.org/abs/1511.08630.

Zielstorff, R. D. (2003). Controlled vocabularies for consumer health. *Journal of Biomedical Informatics*, *36*(4), 326–333. http://dx.doi.org/10.1016/j.jbi.2003.09.015.