Neural Methods for Biomedical Synonym Discovery and Concept Alignment

Leonor Fernandes
leonorcsjfernandes@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2021

Abstract

In the biomedical domain, the identification of synonymous concepts is highly challenging, due to vocabulary heterogeneity, lexical variations, and non-uniform coverage across standardized terminologies. This work tackles this particular challenge, arguing that concept alignment can be made through approximate string similarity using deep neural networks. In particular, this work extends recent studies that assessed string-matching methods in non-biomedical fields, i.e. using bi-directional recurrent neural networks or transformer models to encode and match pairs of strings. The models were trained with biomedical data collected from Wikidata, and tested on 15 datasets built from different biomedical ontologies, representing specific domains. The tests assessed aspects such as the influence of positional encodings together with the inputs, the size of the training dataset or the contribution of model fine-tuning with specific in-domain data. The experimental results show that deep neural networks consistently performed better than traditional string similarity approaches, particularly with larger amounts of training data. In most of the tests, models based on Transformers also performed better than models based on recurrent neural networks.

Keywords: Biomedical Concept Alignment, String-Matching, Supervised Machine Learning, Recurrent Neural Networks, Transformer Networks

1. Introduction

The standardisation of biomedical terminology and interoperability of electronic health systems are enabled by controlled vocabularies and ontologies. Controlled vocabularies are expressed by cataloged concepts and terms [26] whilst biomedical ontologies provide a formal definition of biomedical concepts and relationships between them [2]. This type of resources, including well-known examples, such as SNOMED[1] or the International Classification of Diseases (ICD)[2], usually contain synonyms for most terms, with Natural Language Processing (NLP) methods leveraging terminological resources, but they seldom cover all potential synonyms. This hinders tasks such as integrating clinical notes across different authors and domains or identifying new or rare terms that are not presented in standardized terminology [31]. Moreover, there are many ontologies covering overlapping domains, which leads to the same concept being defined in different ontologies with different terms [25]. The non-uniform coverage across subjects or languages motivates the development of methods for automatically performing alignments between multiple existing specialized terminology resources, in order to link together synonymous concepts across different vocabularies [10].

Concept alignments have been extensively studied within the context of ontology alignment. Most existing methods correspond to heuristic approaches combining multiple types of similarity metrics [11]. In the clinical-medical domain, similar concepts can be lexically similar (e.g. *dilated RA* and *dilated RV*), but also highly dissimilar (e.g. *cerebrovascular accident* and *stroke*). Thus, using similarity metrics based on matching character subsequences can be especially challenging for medical synonym discovery. As an alternative, recent studies have successfully explored deep learning approaches for synonym discovery in various contexts [39, 18, 21, 30, 4].

This article explores and proposes neural methods for concept alignment in the biomedical domain. Extending previous studies in the area, this work specifically assesses methods based on recurrent neural networks or Transformer. For this, a generic dataset is used to train the models, and

[1] https://www.snomed.org/

[2] https://www.who.int/standards/classifications/classification-of-diseases

the models are tested in 1 in-domain and 14 cross-domain datasets.

The remainder of this article is organized into the following sections: Section 2 contains related work on biomedical ontology and concept alignment, as well as deep neural networks used for similar tasks in other fields; Section 3 details the proposed models; Section 4 presents information on the datasets used in this work together with the experimental protocol and the main obtained results; finally, Section 5 draws the main conclusions and advances proposals for future work.

2. Related Work

In order to understand the baseline of the current work, this section introduces neural network models used in NLP and reviews related work on biomedical ontology and concept alignment as well as on string/matching methods in other fields.

2.1. Neural Network Models for Natural Language Processing

In NLP applications, various machine learning algorithms have been applied, with deep neural networks being a prominent choice nowadays. Specifically, Recurrent Neural Networks (RNNs) are currently present in many state-of-the-art approaches. RNNs are neural networks designed to recognize patterns in sequences of data such as character strings [29]. These time-dependent neural networks compute a hidden state vector h_t at each time step t. The hidden state is obtained by a non-linear transformation that receives as inputs the previous hidden state h_{t-1} and the current input x_t

$$h_t = f(h_{t-1}, xt). \tag{1}$$

At a certain time-step t, the hidden state h_t is a function of the input at the same time step x_t, modified by a weight matrix W. This result is added to its own hidden-state-to-hidden-state matrix U, also known as a transition matrix, and multiplied by the hidden state of the preceding time step h_{t-1}. The weight matrices are essentially filters that determine how much importance should be given to both the present input and the past hidden state

$$h_t = \phi(Wx_t + U_h ht - 1). \tag{2}$$

According to previous research, modeling long sequences is challenging for standard RNNs. As a result, several extensions have been explored to handle this problem [30]. Well-known examples include Gated Recurrent Units (GRUs) [6] and Long Short-Term Memory Networks (LSTMs) [16]. GRUs involve two gates: a reset gate r, that determines how to combine the new input with the previous memory and an update gate z that defines how much of the previous memory is kept and how much new information is added. Mathematically, these models can be defined by Equations 3 to 6:

$$z_t = \varphi_g(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z), \tag{3}$$

$$r_t = \varphi_g(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r), \tag{4}$$

$$\tilde{h_t} = \varphi_h(W_h \cdot x_t + U_h \cdot (r_t \odot h_{t-1}) + b_r), \tag{5}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t, \tag{6}$$

where x_t refers to the input vector at a certain time step t, $\odot$ is the Hadamard product (i.e. the entry-wise product of two matrices) and parameters W, U and b denote different weight matrices and biases that are adjusted when training the model through back-propagation. LSTMs are similar to GRUs but have more parameters (e.g. an extra gate). These networks apply different gating mechanisms, more specifically a forget gate f_t that controls how much of the previous gate will be kept, an input gate i_t that controls how much of the proposed gate g_t should be kept, and an output gate o_t that controls the output at time t. Another relevant extension is the use of bi-directional RNNs (BiRNNs) [32]. BiRNNs are composed of two RNNs and read an input sequence in both directions, therefore getting information from past and future states simultaneously. The forward RNN reads the input from left to right, hence capturing unbounded left side context, and the backward RNN reads the input from right to left, therefore capturing the unbounded right side context. The hidden states for each of the RNNs are concatenated according to Equation 7, where h_t^f and h_t^b are respectively the forward and backward hidden states and $\oplus$ is the concatenation operator

$$h_t = h_t^f \oplus h_t^b. \tag{7}$$

Besides RNN models, other neural architectures are also commonly employed to model sequential data, including Convolutional Neural Networks (CNNs) [19, 13] and Transformer models [9]. The Transformer model architecture was proposed by Vaswani et al. [37] as a way to use an attention scheme to model input and output dependencies without needing recurrence or convolutions. The scaled dot product attention mechanisms, in which numerous attention heads are applied in parallel, enabling the model to attend to distinct representation sub-spaces at different points, are the foundations of this model. A Transformer encoder has two sub-layers within each layer. The first sub-layer has a Multi-Head Attention module that aggregates the embeddings (V) of a collection of keys (K) to compute the output embeddings for a set of queries (Q) (Equations 8, 9 and 10). The second sub-layer corresponds to a position-wise fully-connected feed-forward network.

$$\begin{aligned}\text{MultiHead}(Q,K,V) = \\ \text{Concat}(head_1,...,head_h)W^O,\end{aligned} \tag{8}$$

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{9}$$

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V. \tag{10}$$

In the previous expressions, W_i^Q, $W_i^K$and W_i^V are matrices that linearly project queries, keys and values into the attention space of the ith head, while W^O is a matrix that linearly transforms the concatenation of the outputs of all heads. Transformer models have also been extended. A relevant example is the extension to the R-Transformer Model [40] that combines a local RNN with the Transformer attention mechanism.

2.2. Biomedical Ontology and Concept Alignment

When considering ontology matching in the biomedical domain, it is important to be aware of the rich lexical component of biomedical vocabulary and consider a variety of annotations per class, instead of considering only the primary name of each class within each biomedical ontology. Faria et al. [11] have shown that it is more effective to use all available synonyms for a certain concept, and only by doing so can biomedical ontologies from different communities be effectively bridged.

Several attempts have been made to build machine-learning algorithms based on binary classification for ontology matching [23]. These approaches include, among others, classifiers based on decision trees [3], Support Vector Machines [23], and Logistic Regression [1].

Many state of the-art-approaches rely on contextual or external information to aid on identifying medical synonyms or performing biomedical ontology alignment. Wang et al. [39] and Jiang et al. [18] both propose supervised ontology alignment methods through neural networks - a siamese multi-layer perceptron with a sigmoid function [39] and a LSTM based method enhanced with a char-embedding technique [18]. On the other hand, Scumaster et al. [31] and Kolyvakis et al. [21] present unsupervised learning methods to tackle these challenges. Schumaster et al. [31] developed a neural network that makes use of contextual information from surrounding text or patient information to build synonym representations and perform the task of synonym discovery. Kolyvakis et al. [21] describe a network based on embedding ontological terms in a high-dimensional Euclidean space to perform ontology alignment, relying on a similarity function whose measurement is higher in the cases where vectors of words that appear in the same sorts of context.

Although these methods have shown to be effective there is an underlying struggle on identifying the most suitable and useful sources of background knowledge [11], which in itself has also been a topic of several studies. [12, 14, 33].

Another challenge related to concept alignment is that controlled vocabularies are not accessible in all languages and often lack complete definitions. Rahimi et al. [28] focused on aligning a controlled vocabulary - the Unified Medical Language System (UMLS) to Wikipedia, whose health related articles can contribute with content and multilinguality, through a neural ranking model.

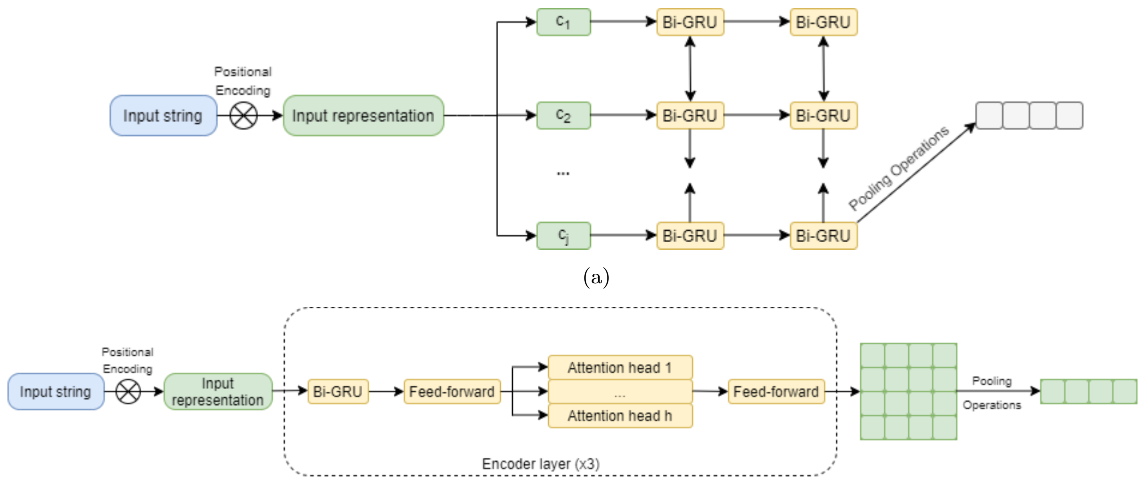
2.3. String-Matching through Deep Neural Networks

String matching is the task of identifying character strings that represent the same real-world entity or concept [4] (i.e. determining whether two strings s_1 and s_2 refer to the same concept).

Traditionally, string similarity metrics can be used as a method to determine if two string correspond to the same concept. These metrics can be based on character operations (e.g. Levenshtein [22] and Jaro-Winkler [5] distances), vector-space representations (e.g. Jaccard [17] and cosine-similarity [7] distances) or hybrid methods of the aforementioned metrics (e.g. Monge and Elkman [24] distance). A more recent character operation metric (I-Sub) has also been used in many approaches, since it was specially developed for ontology alignment [34].

Recently, deep learning approaches have been successfully explored as an alternative to standard string similarity metrics in various domains. Many of these deep learning models models that accept a certain vector x as input. Hence, it is useful to represent textual information in a vector form. A well-known way to do this is to rely on one-hot vectors, where each instance of a given vocabulary with dimension V is represented in the vector with value of 1 [27], whilst all other vocabulary instances are represented by 0. Recently, Wang et al. [38] explored the addition of positional embeddings to these representations. The author showed that in a variety of tasks, including positional encoding at a feature level, consistently obtained better results, which inspired the extension proposed in this dissertation.

Conneau et al. [8] advanced a generic architecture for determining, from a premise sentence, if a given hypothesis sentence can be inferred. This architecture has also been used for string-matching [30, 4]. In these cases both strings were encoded by a RNN, creating a representation of each vector that were then matched in some way (e.g. through



(a)

(b)

Figure 1: (a) The string encoder proposed by Santos et al. [30] with the extensions proposed by Borges et al. [4] (average and max-pooling operations) and the positional encoding proposed in the present work. (b) Transformer extension proposed by Borges et al. [4] with the positional encoding proposed in the present work.

the vector difference, through a concatenation of the vectors, and/or through an element-wise product), fed into a set of fully-connected layers, and finally processed through a feed-forward layer with a sigmoid activation, that generates a binary decision.

Santos et al.'s architecture [30] took inspiration on the previously described generic architecture. This neural network is a siamese RNN (i.e., a network that has an architecture where different parts have their parameters tied) that receives as input a sequence of one-hot vectors (corresponding to the characters of the input string). The string encoder, that consists of a stack of two bi-directional Gated Recurrent Unit (bi-GRU), then outputs the final hidden state of the second bi-GRU, which serves as a vector summary of the input string. Hence, if $X = [x_1, ..., x_L]$ is a sequence of one-hot vectors corresponding to the byte representations of the characters that compose an input string, with length L, we can denote by H the sequence of hidden states output by the second Bi-GRU, and by s the final representation of the input string, as described by Equations 11 and 12.

$$H = \text{BiGRU}(\text{BiGRU}(X)) = (h_i, ..., h_L) \qquad (11)$$

$$s = h_L \qquad (12)$$

After obtaining the two embedding vectors from the two layers of bi-GRUs, these are combined into a a single representation by concatenating them and by calculating the element-wise product and the difference between them. This representation is finally fed into two feed forward layers to produce the final output. These layers consist of a simple combination of the inputs in addition to a Rectified Linear Unit (i.e. a nonlinear activation function) and a sigmoid activation function.

Borges et al. [4] proposed several extensions to this model. Instead of considering the final hidden state of the second bi-GRU as the input vector representation the authors considered: the use of max-pooling and average-polling operations over the hidden states from the second bi-GRU to create the representation of the whole input string; the use of an inter-attention (i.e., alignment) layer, allowing the model to learn to attend and align different pairs of characters between the two input strings. The substitution of the activation functions of the GRU cell with a penalized hyperbolic tangent activation function was also advanced. Moreover, Borges et al. [4] proposed another extension using the R-Transformer model [38] for string matching. In this case, instead of two bi-GRU the encoder layers use a single and first bi-GRU followed by multi-headed scaled dot product attention mechanism to capture interactions between the two input strings. In order to obtain the vector representation for the input, these layers are followed by max-polling and average-pooling operations to aggregate the outputs of the final encoder layer. The remaining layers (specifically, the two feed-forward layers) maintain themselves the same as in the previous architecture. The bi-GRU with the max-pooling and average-pooling extension presented better results in most cases.

The string matching problem can also be formulated as retrieval-based ranking problem where, given a string, the goal is to rank a set of other similar strings, with the most similar placed on top. Gan et al. [13], Traylor et al. [36] and Tam et al. [35] all presented deep neural networks to

Table 1: Testing Datasets Description

Dataset	Source	Total	Positive	Dissimilar	
				Matches	Non-Matches
Wikidata (train)	Wikidata	1 250 000	625 000	1.02%	0.00%
Wikidata (test)	Wikidata	43 214	21 607	0.99%	0.00%
Orphanet Rare Diseases Ontology (ORDO)	OLS	116 860	58 430	33.35%	0.10%
Human Disease Ontology (HDO)	OLS	98 494	49 247	7.55%	0.38%
Foundational Model of Anatomy Ontology (FMA)	OLS	198 306	99 153	2.28%	0.00%
Uber-anatomy ontology (Uberon)	OLS	300 322	150 161	7.07%	0.30%
Human Phenotype Ontology (HPO)	OLS	350 234	175 117	8.93%	0.14%
Mammalian Phenotype Ontology (MPO)	OLS	691 680	345 840	1.44%	0.01%
National Cancer Institute Thesaurus (NCIT) subset	OAEI	7 592	3 796	0.09%	0.01%
Mouse adult gross anatomy (MA)	OAEI	768	384	0.00%	0.00%
FMA + NCIT subset - 1	OAEI	26 752	9 229	0.00%	0.00%
FMA + NCIT subset - - 2	OAEI	20 000	7 082	0.00%	0.01%
MA + NCIT subset - - 1	OAEI	6 705	1 744	0.00%	0.01%
MA + NCIT subset – 2	OAEI	6 000	1 596	0.00%	0.01%
SNOMED CT	SNOMED CT	3 988	1 994	0.00%	0.00%
NCBI disease entities	NCBI Disease Corpus	15 541	7 770	0.76%	0.00%

address this problem. Gan et al. [13] proposed a string encoder combined with a ranking component. The string encoder was represented either by a CNN, where final vector representations for the input string were obtained by concatenating results of a max-pooling operation over the outputs of the three convolution layers, or a bi-directional LSTM, where vector representation was defined as the last hidden state of the neural network. The ranking component ranked the candidates based on their cosine similarity with the query representations. Traylor et al. [36] and Tam et al. [35] also used bi-directional LSTMs to encode pairs of strings, but considered the whole sequence of vectors output by the bi-LSTM for the input strings. In the first case, a CNN with max-pooling was applied to an alignment matrix (obtained by multiplying both string representations), whilst in the second case a transport plan matrix (the conversion of the encoding of one string to the encoding of the other string) is multiplied element-wise by a similarity matrix (the inner product of both string representations) and its result is fed to a three layer CNN. Both approaches output the desired score through a final linear layer.

3. Proposed Approach

This work tackles the string matching problem within the biomedical domain. Leveraging the neural network models proposed by Santos et al. [30] and Borges et al. [4] (described in Section 2), I propose to extend RNN and R-Transformer models with positional encodings in order to classify the strings as matching or non-matching. More specifically the positional encoding is added to the RNN model from Borges et al. [4] with max-pooling and average-pooling operations and to the Transformer model. The implementation relied mostly on *Pytorch Lightning*[3] deep learning library. The datasets, trained model and source code are available in a public github repository[4].

3.1. Positional Encoding

The proposed positional encoding is added to the aforementioned models so that, given a certain pair of strings, the input representation includes information on each character's position, instead of containing information only on which character it represents (i.e., a one-hot vector).

A trigonometric position embedding [37] was added to the input representation where each position embedding is selected as:

$$\begin{aligned} PE_{2k}(\cdot, pos) &= sin(pos/10000^{2k/d_{model}}), \\ PE_{2k+1}(\cdot, pos) &= cos(pos/10000^{2k/d_{model}}). \end{aligned} \tag{13}$$

In the previous expressions, pos is the position index, $2k$ and $2k+1$ are the dimension index and d_{model} is the dimension size of embedding.

The overall models are instances of the generic approaches illustrated in Figure 1.

4. Experimental Methodology

This section describes the experimental evaluation of the methods described in the previous section, first detailing the datasets considered in training and evaluating the proposed approaches and then presenting the obtained results.

4.1. Description of the Datasets

The proposed neural networks were trained with a dataset featuring instances retrieved from Wikidata, and they were tested on 15 datasets corresponding to pairs of strings from different biomedi-

[3] https://www.pytorchlightning.ai/

[4] https://github.com/LeonorFernandesIST/BiomedicalConceptAlignment.git

Table 2: Results with the Levenshtein and Jaro-Winkler metrics

Testing Dataset	Levenshtein ($\alpha = 0.1$)				Jaro-Winkler ($\alpha = 0.1$)			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Wikidata	48.97	**49.53**	97.96	**65.80**	**49.00**	49.49	**98.03**	65.78
ORDO	43.23	46.37	86.47	60.37	**45.56**	**47.68**	**91.13**	**62.61**
SNOMED	50.00	50.00	100.00	66.67	50.00	50.00	100.00	66.67
HDO	**48.95**	49.47	97.90	65.73	49.69	**49.85**	**99.39**	**66.40**
NCBI	49.41	49.76	98.82	66.19	**49.63**	**49.81**	**99.25**	**66.33**
HPO	49.29	49.64	98.60	66.03	**49.48**	**49.74**	**98.98**	**66.20**
Uberon	49.75	49.87	99.50	66.44	**49.84**	**49.92**	**99.68**	**66.53**
FMA	49.85	49.93	99.70	66.54	**49.93**	**49.97**	**99.86**	**66.61**
NCIT subset	**49.72**	49.86	**99.45**	66.42	49.70	**49.87**	99.39	66.42
MA	49.61	49.80	99.22	66.32	**49.70**	**49.87**	**99.39**	**66.42**
MPO	49.37	49.68	98.75	66.11	**49.70**	**49.87**	**99.39**	**66.42**
FMA + NCIT subset - 1	**34.49**	34.49	**100.00**	51.29	34.48	**34.51**	99.97	**51.31**
FMA + NCIT subset - 2	**35.41**	35.41	**100.00**	52.30	35.40	**35.43**	99.96	**52.32**
MA + NCIT subset - 2	26.60	26.60	100.00	42.02	26.60	**26.65**	100.00	**42.09**
MA + NCITsubset - 1	26.01	26.01	100.00	41.28	26.01	**26.06**	100.00	**41.34**

cal ontologies or collected from biomedical text corpora. In general, a positive instance corresponds to the case where both strings in a pair correspond to the same concept.

The training dataset is a generic balanced dataset with 1 250 000 pairs of strings (and, therefore, with 625 000 positive instances). In order to obtain a large and generic dataset, concepts were retrieved from Wikidata[5], belonging to the following classes: *physiological condition, biological component, health science, biology, group or class of chemical substances, zootomy, veterinary medicine, comparative medicine, biomolecular structure, biological region, anatomical entity, biological system, general anatomical term and phenotype.* Positive instances correspond to pairs of concepts that are presented as synonyms in this platform (e.g. *induced miscarriage* and *abortion*) whereas negative instances were generated with randomly selected concepts that were not synonyms or generated with the replacement of words in a given list by there antonyms (e.g., concepts with *anterior* replaced by *posterior*). A significant portion of the non-matching pairs are not completely dissimilar, so that the dataset is representative and challenging for automated classification (e.g., *complex global pain syndromes* and *complex regional pain syndromes* are non-matching pairs).

A testing dataset was also collected from the same Wikidata classes. This dataset does not have any pair equal to the ones in the training dataset and is to be considered as a validation dataset from the same domain. The remaining datasets were retrieved from the following sources:

- Ontology Lookup Service (OLS)[6], i.e. a repository for biomedical ontologies that aims to provide a single point of access to latest ontology versions. Positive instances correspond to identified cross-reference concepts.

- Ontology Alignment Evaluation Initiative (OAEI)[7], whose major purpose is to openly compare systems and algorithms on an equal basis so that anybody is allowed to make informed decisions regarding the best matching techniques. Datasets retrieved from OAEI are from either ontologies/subsets used as sources for ontology alignments, or from already performed alignments in anatomy tracks. In the first case, positive instances correspond to pairs of strings belonging to the same name set (i.e. group of synonyms and main labels of a class). In the second case, positive instances correspond to mapped synonyms in the performed alignment. In both cases, negative instances all have an ISub similarity ≥ 0.7.

- Systemized Nomenclature of Medicine – Clinical Terms (SNOMED-CT), i.e. a standardized, international, multilingual core set of clinical healthcare terminology that can be used in electronic health records. Positive instances correspond to English terms with the same SNOMED-CT code.

- National Center for Biotechnology Information (NCBI) disease corpus[8], a resource for disease name recognition and normalization. Positive instances correspond to strings marked as entities in the text with the same SNOMED-CT code.

[5] https://www.wikidata.org/wiki/

[6] https://www.ebi.ac.uk/ols/index

[7] http://oaei.ontologymatching.org/

[8] https://www.ncbi.nlm.nih.gov/research/bionlp/data/disease

Table 3: Results with the proposed RNN and R-Transformer models

Testing Dataset	Proposed RNN				Proposed R-Transformer			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Wikidata	89.87	89.48	90.34	89.87	**93.00**	**92.33**	**93.80**	**93.03**
ORDO	86.82	90.14	82.65	86.19	**90.15**	**93.23**	**86.55**	**89.73**
SNOMED	82.95	76.24	**95.88**	84.91	**84.34**	**79.57**	92.55	**85.54**
HDO	81.76	78.59	87.32	82.67	**84.41**	**82.02**	**88.16**	**84.92**
NCBI	75.48	73.43	79.62	76.34	**76.46**	**74.36**	**80.73**	**77.34**
HPO	68.97	67.72	72.48	69.94	**73.69**	**73.57**	**73.95**	**73.68**
Uberon	68.28	65.73	76.39	70.58	**71.77**	**69.24**	**78.38**	**73.45**
FMA	71.94	71.94	**75.97**	72.95	**74.29**	**75.90**	71.20	**73.39**
NCIT subset	69.83	67.43	76.73	71.73	**70.03**	**66.68**	**80.02**	**72.69**
MA	63.93	62.72	67.62	65.02	**67.32**	**65.21**	**74.22**	**69.42**
MPO	65.82	66.28	64.39	65.23	**68.33**	**69.62**	**65.09**	**67.18**
FMA + NCIT subset - 1	**69.48**	**53.58**	85.95	**65.93**	66.05	50.46	**86.57**	63.66
FMA + NCIT subset - 2	**68.84**	**53.81**	86.07	**66.11**	64.94	50.38	**86.15**	63.45
MA + NCIT subset - 2	**68.42**	**45.21**	91.22	**60.36**	63.90	41.77	**92.52**	57.44
MA + NCIT subset - 1	**68.75**	**44.77**	90.88	**59.88**	64.51	41.76	**92.08**	57.34

All datasets are presented in the Table 1 detailing their source, the total number of pairs, the total number of positive instances, the percentage of pairs with matching concepts that are completely dissimilar and the percentage of pairs with non-matching concepts that are completely dissimilar. Non-matching concepts correspond to cases when the pairs had a Jaro-Winkler similarity of 0.

Apart from the OAEI datasets derived from the alignment between FMA or MA with the NCIT subset, all other datasets are balanced (i.e. half of the instances correspond to matching concepts). The four imbalanced datasets have more non-matching than matching concepts. The existence of totally dissimilar matches occurs with an incidence higher than 1% only in 8 of the datasets and from these, only the ORDO dataset has an incidence superior to 10%. None of the datasets present a high percentage of totally dissimilar non-matches (they are all below 1 percent).

Additionally, it is important to refer that none of the datasets present cross-language pairs. The English language was considered in all cases.

4.2. Evaluation Methodology

The proposed approach was compared to baseline methods over all the testing datasets. Specifically, the considered baselines consist of individual string similarity metrics, with a threshold value α tuned for optimal F1 score on average over all the datasets. Results were measured in terms of accuracy, precision, recall, and the F1 measure.

In terms of hyper-parameter choices and model training strategies, the tests with models leveraging RNNs used an RNN hidden layer size of 60, a hidden layer size of 120 in the dense layer processing the result from the interaction between the string representations, a batch size of 32, and the Adam [20] optimizer with a learning rate of 0.001. Regarding experiments with the R-Transformer encoder, 3 layers of dimensionality 512 were considered, leveraging 8 attention heads in parallel.

Model training is performed for a maximum of 20 epochs over the training dataset with early stopping being activated when the training loss does not decrease after 3 epochs. Other experiments were conducted in regards to the assessment of the impact of the training dataset size or the contribution of model fine-tuning with specific in-domain data. The in-domain experiments were done with two-fold cross validation, according to a stratified sampling procedure.

4.3. Results

Tables 2, 3, and 4 present the obtained results.

Table 2 details the results over each dataset with baseline methods. Although experiments were conducted with 6 traditional string similarity measures, only the results with Levenshtein and Jaro-Wrinkler metrics are presented, since they obtained better overall results and can, therefore, fairly be compared with the proposed neural network results. It is also important to notice that the threshold value α was tuned for optimal F1 scores and has, in both cases, the value of 0.1. Hence, it is highly likely that most pairs are considered positive instances, which results in the high recall values observed.

Table 3 presents the proposed neural network methods. The results show that neural methods outperform traditional string similarity measures in terms of accuracy and F1 score in all testing datasets. In most cases, the proposed R-Transformer model obtains better scores than the bi-GRU model.

The 4 imbalanced datasets (FMA + NCIT subset 1 & 2 and MA + NCIT subset 1 & 2) correspond

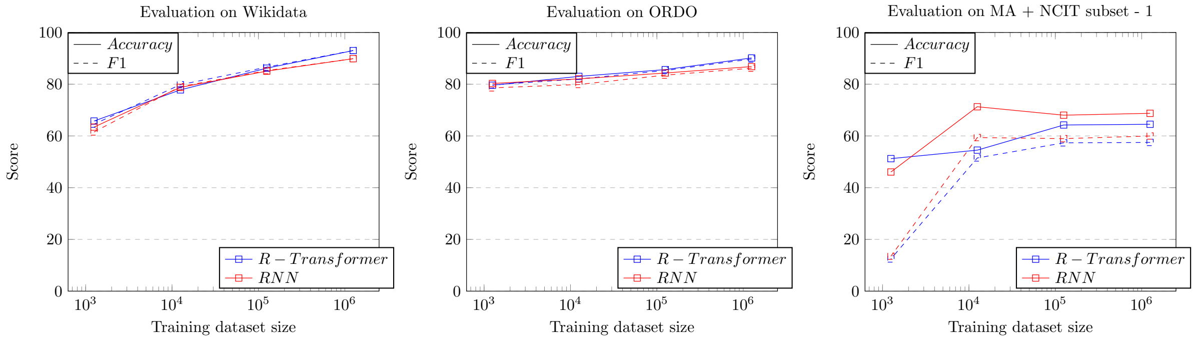


Figure 2: Results changing the training dataset size

to the worst accuracy and F1 scores with the R-Transformer model. Additionally, they correspond to the cases where the RNN outperforms the R-Transformer model. It is also interesting to notice that the recall is always high (above 85 %) in these cases, whilst precision is rather low (below 54%). These imbalanced datasets not only have more negative instances than positive, but are also datasets where pairs of strings that corresponded to negative instances presented a high similarity between them (ISub $\geq$ 0.7), thus, contributing to the identification of false positives.

Datasets with pairs retrieved from disease related ontologies (ORDO, HDO, NCBI) or general health terms (SNOMED CT) had better results than datasets related to phenotypes and anatomy terms. This can be related to the classes from which the Wikidata training set was retrieved (i.e., anatomy and phenotype terms are underrepresented in relation to disease related terms).

Another interesting note is that the percentage of totally dissimilar pairs that are synonyms does not seem to have an influence on the results. The ORDO dataset presents the highest percentage of totally dissimilar matches (33.35%) and is also the testing dataset with the highest scores (excluding the Wikidata validation set). Furthermore, the datasets that do not present any dissimilar matches do not perform necessarily better or worse (e.g. comparing the SNOMED-CT and MA + NCIT subset - 1 datasets' results one can infer that this does not have a direct influence).

In order to assess the impact of the training dataset size, experiments were conducted in which the size of the training dataset was reduced (using stratified folds) and the effect on the evaluation metrics evaluated. This was conducted for the following testing datasets: Wikidata, which is considered from the same domain; ORDO and MA + NCIT subset - 1 which obtained better and worse results in the previous tests, respectively. Figure 2 illustrates the results for this set of experiments.

Both the ORDO and Wikidata datasets show that a greater amount of training data leads to better results, independently of the model being used. The R-Transformer model continues to outperform the RNN in most cases, although the difference between the two models' scores is smaller with smaller training sets. On the other hand, experiments with the MA + NCIT subset- 1 dataset were different than expected. Although, in the R-Transformer model the pattern maintains itself (higher scores for bigger training sets), for the RNN model we have that the 12 500 size dataset presented the better scores. Moreover, in this case the difference between the dataset scores is bigger when using smaller training sets (with the R-Transformer model outperforming the RNN model in the extreme of the smallest training set). For a small training generic dataset (1 250 pairs of strings) the results are rather discouraging, obtaining in both MA + NCIT subset - 1 and Wikidata worst scores than some of the traditional approaches. Hence, the proposed models are considered a good alternative when using a generic training dataset, if it is big and representative enough of the biomedical domain.

The neural model that performed better in most cases (the R-Transformer model) was chosen to design an additional set of experiments to evaluate the effect of how the training domain (i.e. the Wikidata training dataset) affected performance when evaluating the results in same-domain settings. On the one hand, fine-tuning experiments were conducted, where the training set included data from the ontology or domain being tested. I opted for fine-tuning the pre-trained model instead of training it from scratch every time since these are time consuming and resource intensive processes. On the other hand, an experiment where a single 2-fold training was executed with the training dataset including data from all ontologies at once was also conducted. Since the Wikidata testing set was used as reference for the same domain as the training dataset (whilst the remaining datasets were considered cross-domain experiments), I expected that the scores obtained in both cases to become more similar to the ones in the Wikidata dataset. In each

Table 4: In-domain or in-ontology results for the R-Transformer model

Testing Dataset	Fine-tuning				Training with all ontologies			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
ORDO	**95.87**	**97.19**	**94.47**	**95.81**	92.22	96.47	87.66	91.85
SNOMED	**98.76**	**97.56**	**100.00**	**98.77**	91.02	88.44	94.49	91.36
HDO	**91.80**	**95.04**	**88.22**	**91.50**	89.96	91.63	87.94	89.74
NCBI	**88.45**	**97.39**	**79.04**	**87.26**	82.90	84.24	80.75	82.45
HPO	88.94	93.60	85.28	89.25	**89.33**	**94.35**	**85.74**	**89.84**
Uberon	82.38	**89.70**	73.15	80.59	**87.59**	86.60	**88.95**	**87.76**
FMA	**97.04**	96.37	**96.37**	**97.06**	94.67	96.72	92.47	94.55
NCIT subset	81.04	**86.44**	73.61	79.51	**85.45**	87.60	**82.59**	**85.02**
MA	74.80	79.43	67.66	73.07	**89.55**	**90.58**	**89.50**	**90.03**
MPO	**90.22**	**96.30**	**83.66**	**89.54**	89.24	93.71	81.10	88.29
FMA and NCIT subset - 1	**87.63**	**86.51**	76.17	**81.01**	84.38	73.96	**84.46**	78.87
FMA and NCIT subset - 2	**86.40**	**84.88**	75.15	**79.72**	83.62	73.44	**84.28**	78.49
MA and NCIT subset - 2	**90.35**	**82.10**	74.02	77.85	87.02	69.91	**90.00**	**78.69**
MA and NCIT subset - 1	**89.82**	**83.21**	76.04	**79.46**	86.93	69.04	**78.62**	73.52

test it was ensured that the dataset being evaluated was not present at train time. Table 4 presents the obtained results for these tasks, where one can observe that in both cases the scores obtained were better in the 4 evaluation metrics for all datasets. For the R-Transformer model, accuracy scores were all above 80% and F1 scores all above 70%.

Fine-tuning involves initializing the deep learning process with weights of the pre-trained model, and training it with the new data. The model is, therefore, adjusting its weights to the new data. In the case of imbalanced datasets, this can be extremely important, since it can perform a class reweighting (i.e. to take into account asymmetry of cost error directly during the training of the classifier). In my experiments, results for the imbalanced datasets all improved significantly, increasing at least 21% in terms of accuracy and 16% in terms of F1 score. These datasets also presented better results in the fine-tuned models than when training with all ontologies, as expected. Seeing that there is less adjustment to the imbalanced classes, recall maintains itself higher in the model trained from scratch. It is also interesting to notice that the datasets which improve less with fine-tuning are the ORDO, in terms of accuracy, and the MA, in terms of F1 score. In the first case it may be due to the fact that the model already obtained high scores without fine-tuning, and hence the rare disease ontology was probably already well represented in the model. In the second case, it is important to refer that the mouse ontology is the smallest dataset, with only 768 pairs of strings. Consequently, when fine-tuning the model with each fold of 384 pairs the available data might not be enough to adjust the weights significantly to the domain.

Concerning the results obtained from the model trained with a dataset including data from all ontologies at once, it is significant to note that I am not only adding these domains and ontologies to the training set but also enlarging it significantly (2 193 272 instances in each fold). As shown previously, the size of the training dataset also influences the outcome and, therefore training with a dataset that is approximately 1.75 times larger than the original training dataset contributes to the generally better obtained scores. In particular, this experiment showed that the ORDO dataset presented worse results than with the initial training; the rest of diseases or general medical terms related datasets all improved in comparison to initial training, but performed worse than fine-tuning the model. Most of the balanced datasets with anatomy or phenotype related terms obtained improved the scores. Assuming that, as mentioned before, the initial Wikidata training dataset was underrepresented in terms of anatomy of phenotype related concepts, then these results support this idea. These datasets benefit from each other being in the training dataset, and the percentage of anatomical and phenotype representation increases. Disease or generic related datasets perform better with fine-tuning because the initialized weights already benefited them and are then adjusted in-domain, not needing each other to perform better. Hence, it is seems to be extremely challenging to find a large and generic dataset that not only includes all relevant biomedical classes but that also contains them proportionally so that it is applicable with all ontologies.

5. Conclusions and Future Work

This article describes extensions of the neural string matching methods developed by Santos et al. [30] and Borges et al. [4], augmenting the proposed architecture to include positional embeddings, and assessing its performance in cross-domain settings (the main goal for biomedical concept alignment), in in-domain settings, and when varying the amount of training data. The proposed models were tested on different datasets, covering several biomedical ontologies and domains (i.e., disease, anatomy and phenotype related ontologies).

A comparison was also performed between the proposed neural models against classical string similarity metrics where the proposed neural network models consistently outperformed the other techniques. Except for the 4 imbalanced datasets (out of 15 testing datasets), the R-Transformer model outperformed the one based on RNNs.

Regarding the experiments in which the size of the training datasets was varied, both models performed better with a larger number of training instances. It is possible to observe that when the training dataset was very small (i.e. only 1250 instances) the results were discouraging leading, in some cases, to a worse score than traditional approaches.

After identifying the neural architecture that consistently obtained better results, I focused on assessing how well the proposed models performed when considering data of the same ontology or domain. The results showed that cases where training included in-ontology terms performed better than the initial experiments. However, in the biomedical context, it would be relevant that in-ontology training was not needed. This would be useful to perform concept alignment even when the ontology is not known. Moreover, it can enhance synonym discovery and identification between several authors and contexts. All things considered, a large training dataset that considers and represents the most biomedical categories it can should be aimed for. Wikidata, still seems to be a good option for this data retrieval since it is a large-scale collaborative ontological medical database.

There are several potential paths for future research based on the findings given in this article. A straightforward future experiment would be to retrieve a larger amount of data from Wikidata for the training dataset, including a larger amount of classes in order to obtain more anatomy and phenotype related terms. However, it is also important to notice that these models are time and resource consuming, so there should be a balance between enlarging the dataset and computational effort. Fine-tuning pre-trained models has showed to be a successful alternative for this balance (e.g. using the already trained model and adding more data). In what regards to the positional encoding, it would be interesting to generalize the word and character embeddings as continuous functions over a variable (position) instead of being defined as independent vectors. Recently Wang et al. [38] demonstrated this to be more efficient. Keeping in line with recent advances in natural language processing, I believe other extensions to the Transformer model might be beneficial to the task in hand (e.g. considering the residual attention layer Transformer [15]). Finally, testing with cross-language datasets would also be an interesting experiment, wherefore contributing to the multilinguality challenge mentioned in Section 2.

Acknowledgements

I would like to express my gratitude to Professors Bruno Martins, Daniel Faria and Cátia Pesquita, who co-supervised this work and the elaboration of the thesis underlying it.

References

[1] N. Alboukaey and A. Joukhadar. Ontology matching as regression problem. *Journal of Digital Information Management*, 16(1), 2018.

[2] B. Aldosari, A. Alanazi, and M. S. Househ. Pitfalls of ontology in medicine. *Studies in health technology and informatics*, 2017.

[3] S. Amrouch, S. Mostefai, and M. Fahad. Decision trees in automatic ontology matching. *International Journal of Metadata, Semantics and Ontologies*, 11(3):180–190, 2016.

[4] L. Borges and B. Martins. Evaluating neural methods for approximate string matching and duplicate detection. Technical report, Instituto Superior Técnico, 2019.

[5] P. Christen. A comparison of personal name matching: Techniques and practical issues. In *Procedings of workshop at IEEE International Conference on Data*, 2006.

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] W. W. Cohen, P. Ravikumar, S. E. Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the 2003 International Conference on Information Integration on the Web*, 2003.

[8] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] J. Euzenat, P. Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

[11] D. Faria, C. Pesquita, I. Mott, C. Martins, F. M. Couto, and I. F. Cruz. Tackling the challenges of matching biomedical ontologies. *Journal of Biomedical Semantics*, 9(1):1–19, 2018.

[12] D. Faria, C. Pesquita, E. Santos, I. F. Cruz, and F. M. Couto. Automatic background knowledge selection for matching biomedical ontologies. *PloS one*, 9(11):e111226, 2014.

[13] Z. Gan, P. Singh, A. Joshi, X. He, J. Chen, J. Gao, and L. Deng. Character-level deep conflation for business data analytics. In *Proceding of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[14] A. Gross, M. Hartung, T. Kirsten, and E. Rahm. Mapping composition for matching large life science ontologies. In *Proceding of the 2nd International Conference on Biomedical Ontology*, 2011.

[15] R. He, A. Ravula, B. Kanagal, and J. Ainslie. Realformer: Transformer likes residual attention. *arXiv preprint arXiv:2012.11747*, 2020.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] P. Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[18] C. Jiang and X. Xue. Matching biomedical ontologies with long short-term memory networks. In *Proceding of the IEEE international conference on bioinformatics and biomedicine*, 2020.

[19] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] P. Kolyvakis, A. Kalousis, B. Smith, and D. Kiritsis. Biomedical ontology alignment: an approach based on representation learning. *Journal of Biomedical Semantics*, 9(1):1–20, 2018.

[22] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[23] M. Mao, Y. Peng, and M. Spring. Ontology mapping: as a binary classification problem. *Concurrency and Computation: Practice and Experience*, 23(9):1010–1025, 2011.

[24] A. E. Monge, C. Elkan, et al. The field matching problem: Algorithms and applications. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[25] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. Storey, C. G. Chute, et al. Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research*, 37(suppl_2):W170–W173, 2009.

[26] T. B. Patrick, H. K. Monga, M. C. Sievert, J. H. Hall, and D. R. Longo. Evaluation of controlled vocabulary resources for development of a consumer entry vocabulary for diabetes. *Journal of medical Internet research*, 3(3):e24, 2001.

[27] D.-H. Pham and A.-C. Le. Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis. *International Journal of Approximate Reasoning*, 103:1–10, 2018.

[28] A. Rahimi, T. Baldwin, and K. Verspoor. Wikiumls: Aligning umls to wikipedia via cross-lingual neural ranking. *arXiv preprint arXiv:2005.01281*, 2020.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[30] R. Santos, P. Murrieta-Flores, P. Calado, and B. Martins. Toponym matching through deep neural networks. *International Journal of Geographical Information Science*, 32(2):324–348, 2018.

[31] E. Schumacher and M. Dredze. Learning unsupervised contextual representations for medical synonym discovery. *JAMIA open*, 2(4):538–546, 2019.

[32] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[33] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, et al. The obo foundry: coordinated evolution of ontologies to

support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255, 2007.

[34] G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In *Procedings of the International Semantic Web Conference*, 2005.

[35] D. Tam, N. Monath, A. Kobren, A. Traylor, R. Das, and A. McCallum. Optimal transport-based alignment of learned character representations for string similarity. *arXiv preprint arXiv:1907.10165*, 2019.

[36] A. Traylor, N. Monath, R. Das, and A. McCallum. Learning string alignments for entity aliases. In *Proceding of the Workshop on Automated Knowledge Base Construction*, 2017.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceding of the Annual Meeting on Advances in neural information processing systems*, 2017.

[38] B. Wang, D. Zhao, C. Lioma, Q. Li, P. Zhang, and J. G. Simonsen. Encoding word order in complex embeddings. *arXiv preprint arXiv:1912.12333*, 2019.

[39] L. L. Wang, C. Bhagavatula, M. Neumann, K. Lo, C. Wilhelm, and W. Ammar. Ontology alignment in the biomedical domain using entity definitions and context. *arXiv preprint arXiv:1806.07976*, 2018.

[40] Z. Wang, Y. Ma, Z. Liu, and J. Tang. R-transformer: Recurrent neural network enhanced transformer. *arXiv preprint arXiv:1907.05572*, 2019.