Attention for Multi-Ontology Concept Recognition

Lorcán Pigott-Dix[1,*], Robert P. Davey[1]

[1]Earlham Institute, Norwich Research Park, Colney Lane, Norwich NR4 7UZ, United Kingdom

Abstract

The increasing scale of scientific output necessitates the use of machine-based tools to index, interpret, and allow scientists to digest the expanding volumes of data and literature. These tools depend on rich machine-readable ontology-based metadata. The scale of this task renders manual annotation infeasible. This work compares multi-ontology deep learning-based models for identifying ontology concepts in the natural language text of scientific literature. An existing convolutional neural net (CNN) architecture was improved and compared with two attention-based variants, a CNN with a Squeeze-and-Excite (SAE) mechanism, and a self-attention architecture that had been adapted for limited training data. The models were assessed against a gold-standard dataset of 228 PubMed abstracts, annotated with Human Phenotype Ontology terms. All models exceeded the previous state-of-the-art, with the SAE model promising to be the best candidate for multi-domain ontology concept extraction.

Keywords

Deep Learning, Ontology, Named Entity Recognition, Concept Recognition

1. Introduction

The amount of scientific literature published is increasing exponentially - with volumes doubling every 15 years [1]. This surge in productivity has been accompanied by an ever-increasing deluge of data. From genetic sequencing data to satellite imagery, manually sifting through this heterogeneous data is an insurmountable task. Computer-based tools that can rapidly find and interpret data promise a remedy. However, they depend on rich metadata to describe these data [2].

Domain experts in the life sciences have been codifying their expertise into ontologies [3, 4]. These ontologies describe these domains using formal, unambiguous, machine-readable "structured vocabularies" [5]. Annotating data with ontology terms attempts to make implicit human-inferable context explicit [6]. Explicit contextualisation allows computer-agents to behave as if they can "interpret" the semantics of a dataset. While funding bodies have begun to require data stewardship plans in grant applications, efforts to generate data far exceed those to curate it. The scale of the task necessitates computer tools to automate, or expedite, the annotation process.

SWAT4HCLS 2023: The 14th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences

*Author to whom correspondence should be addressed.

lorcan.pigott-dix@earlham.ac.uk (L. Pigott-Dix); robert.davey@earlham.ac.uk (R. P. Davey)

0000-0003-3120-5423 (L. Pigott-Dix); 0000-0002-5589-7754 (R. P. Davey)

2. Related work

2.1. Concept recognition

Previous tools for ontology-based concept recognition have largely been rule-based [7, 8]. They typically identify potential concepts within text using string matching, coupled with heuristics to refine the candidate concepts. These methods tend to have high precision but low recall scores, as they are less able to identify synonyms for concepts if they are not represented as found in the ontology.

Recently ontology-based concept recognition methodologies have begun to incorporate neural nets, typically employing recurrent neural nets (RNNs), as RNNs can learn dependencies between words in sequences. These methods employ word embeddings which help to address the synonym-gap as they represent meaning rather than lexical structure. However, these methods rely on substantive manual annotation or noisy heuristic data generation. For example, one [9] used an ontology to heuristically label a training corpus, while another [10] relied on manual annotation carried out by medical specialists.

Arbabi *et al.* [11] created a method that exploits word embeddings and only requires an ontology for training: the Neural Concept Recognizer (NCR). NCR is a "neural dictionary" that uses a convolutional neural net (CNN) to learn associations between sequences of word embeddings and embeddings representing ontology concepts. The neural dictionary converts input text into the representation space of the concepts and finds the most similar concept embeddings. It only requires an OBO format ontology, and no heuristic annotation. When the NCR model was evaluated against text annotated with concepts from the Human Phenotype Ontology (HPO), it achieved both micro and macro F1 scores of 70.2% and 73.9% respectively.

"PhenoTagger" [12] combined a string matching approach with a neural classifier for the task of HPO concept recognition. A dictionary of concepts names, synonyms, and their lemmatised forms, was created from an ontology. The dictionary was used to label a distantly supervised training set, which in turn, was was used to fine-tune a pre-trained BioBERT model. In deployment, PhenoTagger combines the outputs from the string-matching dictionary and BioBERT. PhenoTagger achieves a "document-level" f-score of 75.7% – the current state-of-the-art (SOTA) for neural dictionary methods.

2.2. Attention

In recent years, models containing attention mechanisms have become the SOTA for many natural language processing (NLP) and computer vision tasks, from machine translation [13] to object detection [14]. In neural networks, attention mechanisms are specific trainable weights, that learn to modify the model to increase the signal of task-relevant features and diminish less relevant features. For example, the Squeeze-and-Excite (SAE) attention mechanism [15] emphasises or diminishes feature signals by modelling dependencies between convolutional filters in a CNN. A more sophisticated attention mechanism, multi-headed self-attention (MHSA), explicitly models the semantic dependencies between words in a sequence, in order to compute updated word embeddings.

MHSA is an integral part of SOTA transformer models [13]. However, these models require large volumes of training data to be effective. Guo *et al.* [16] argues that this is because self-

attention models have a poor inductive bias, and instead rely heavily on these large volumes in order to generalise well. This becomes a problem for models that are trained on limited datasets, such as the text provided by an ontology. Arbabi *et al.* [11] tried attention mechanisms as an alternative to the CNN used in their model, but they were not effective. This may be explained by either the poor inductive bias, the limited training data, or the relative efficacy of CNNs at local feature extraction.

Guo *et al.* [16] describe an alternative configuration of MHSA, called Scale-Aware Self-Attention (SASA). With SASA, each attention head attends to a variable scale. The scaling restricts attention to within a certain neighbourhood of each sequence position. The intuition here is that words that are in close proximity within a sentence are more likely to have more relevance to each other. This scaling forces the attention heads to attend to a smaller set of features, so the relative differences between the remaining features are more pronounced, improving the inductive bias. SASA models exceed the SOTA, or are competitive, for a number of low-resource NLP tasks – requiring far fewer training examples than typical MHSA models.

Other methods have been developed to improve the performance of MHSA models. Zhou *et al.* [17] randomly drop entire attention heads during training to prevent a minority of heads from dominating the model, improving the model's ability to generalise. Wu *et al.* [18] applied dropout at various points within the attention mechanism: to the attention weights and activation layer; to the query, key, and value matrices; and to the output features prior to the linear transformation. They also randomly removed a proportion of tokens from the input sequence. This was found to improve the performance of MHSA models without additional training data or computational power.

3. Contribution

We achieve a new SOTA for neural ontology-based concept recognition. The improved performance is obtained by incorporating an attention mechanism in the CNN architecture and by using higher quality word embeddings. Unlike previous neural dictionaries, it can incorporate multiple domain ontologies at once. We found those trained using a combination of diverse ontologies performed the best. This work also demonstrates that transformer-based architectures can be modified, with variable attention scaling, to perform competitively with CNNs in situations where training data is substantially limited. All of the models tested are available here: https://github.com/lorcanpd/adorNER.

4. Methods

4.1. Neural Concept Recognition (NCR) adapted to use ELMo Word Embeddings

The NCR classifier comprises two parts: the concept embeddings and the CNN classifier. The concepts are represented by a matrix of randomly initialised embeddings, which share information between related concepts via multiplication by an ancestry matrix. The CNN classifier passes filters over the sequence of word embeddings (obtained from a pre-trained

ELMo model [19]) representing the natural language descriptions of the concepts, extracts semantic signals, and outputs them into the representational space of the concept embeddings. As the model trains, it learns to reduce the distance between the CNN output and the correct concept's position in representational space.

To perform concept recognition, input sentences are split into all of the possible n-grams they contain, where $n \in \{1, ..., 7\}$. Each n-gram is passed to the classifier, returning candidates with the highest match confidence score above a given threshold. Heuristics then resolve overlap in the text, favouring longer – likely more specific – terms.

4.2. Squeeze-and-Excite (SAE)

The CNN model is augmented to include an SAE mechanism. Here, the methodology described in Hu *et al.* [15] is adapted for a one-dimensional CNN. Average-pooling is applied to each of the non-zero elements of each feature map produced by the convolutional layer. This reduces the feature maps into a single vector, $\mathbf{z} \in \mathscr{R}^F$ where F is the number of feature maps. Each element of $\mathbf{z}$ is calculated like so:

$$z_f = \frac{1}{N_f} \sum \mathbf{u}_f \tag{1}$$

Where z_f is the statistic for the f-th filter, $\mathbf{u}_f$ is the f-th feature map's vector, and N_f is the number of non-zero elements in that vector. The vector of all the feature map statistics is then compressed and decompressed, as follows:

$$\mathbf{s} = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z})) \tag{2}$$

Where $\mathbf{s}$ is the vector of filter weights, $\mathbf{W}_1 \in \mathscr{R}^{\frac{F}{r} \times F}$ the parameter weights of the compression transformation, $\mathbf{W}_2 \in \mathscr{R}^{F \times \frac{F}{r}}$ the parameter weights of the decompression transformation, σ and δ are the sigmoid and ReLU activation functions respectively, and r is the compression ratio. The max-pooled feature maps are then scaled element-wise by $\mathbf{s}$. As the model is trained, the SAE mechanism learns to model dependencies between feature maps.

4.3. Multi-Scale Self Attention (MSSA)

Architecture Here, the CNN architecture is replaced entirely by an encoder based upon Guo's SASA transformer [16]. Given an input of word embeddings $\mathbf{X} \in \mathscr{R}^{N \times D}$, where N represents the number of embeddings and D their dimensionality, each scale-aware attention head can be described as follows:

$$\text{head}(\mathbf{X}, \omega)_{i,j} = \text{softmax}(\frac{\mathbf{Q}_{ij} C_{ij}(\mathbf{K}, \omega)^T}{\sqrt{\frac{D}{h}}}) C_{ij}(\mathbf{V}, \omega) \tag{3}$$

Where ω is the scale parameter, i corresponds to the i-th head, and j to the j-th element of the sequence. $\mathbf{K}$, $\mathbf{Q}$, $\mathbf{V}$ are the projections of $\mathbf{X}$ into $N \times \frac{D}{h}$ subspaces, with h being the number of heads. $\mathbf{X}$ is projected into these subspaces by multiplying it by the parameter matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ (all $\mathbf{W} \in \mathscr{R}^{D \times \frac{D}{h}}$).

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q,\ \mathbf{K} = \mathbf{X}\mathbf{W}_K,\ \mathbf{V} = \mathbf{X}\mathbf{W}_V \tag{4}$$

$C_{ij}(\mathbf{x}, \omega)$ is the context-extraction function, which is defined as:

$$C_{ij}(\mathbf{X}, \omega) = [\mathbf{x}_{i,j-\omega}, \ \dots, \ \mathbf{x}_{i,j+\omega}] \tag{5}$$

When the scale parameter exceeds the range of the sequence, the context-extraction function pads the sequence with zeros of the appropriate dimensions.

The h heads are incorporated into a Multi-Scale Multi-headed Self-Attention (MSMSA) block. This block consists of the scale aware attention layer and a feed-forward network and is computed as follows:

$$\text{MSMSA}(\mathbf{X}, \Omega) = \text{norm}([\text{head}_1(\mathbf{X}, \omega_1), \ \dots, \ \text{head}_h(\mathbf{X}, \omega_h))]\mathbf{W}^O + \mathbf{X}) \tag{6}$$

Where $\Omega \in \{\omega_1, \ \dots, \ \omega_h\}$ is the set of scale parameters, $\mathbf{W}^O$ is a parameter matrix and norm is the layer normalisation function. The output is then passed to a feed forward (FF) layer. MSMSA blocks can be stacked multiple times, with varying sets of scale parameters. The final output is reduced into a single vector $\mathbf{z}$ by summing each output vector element-wise, normalised by the square-root of the sequence length:

$$\mathbf{z} = \frac{\sum_{j=1}^{D} \mathbf{X}_{\cdot j}}{\sqrt{N}} \tag{7}$$

$\mathbf{z}$ is then passed onto a final feed-forward layer that converts the sentence representation into the representation space of the concept embeddings. This layer comprises two consecutive linear transformations, each with GELU activations and l2 normalisation, followed by a final linear transformation with no activation layer.

Dropout regime For each input batch, there was 50% chance of the dropout function being applied to the entire batch. If applied, entire embeddings were dropped from a sequence with a probability of 20%. Sequences shorter than three tokens in length were excluded from this, as there was a chance the input signal would be too degraded. Each attention-head had a 25% chance of an attention head being dropped out. A subsequent dropout was applied after the activation function within the attention block's feed forward network, where random elements were dropped from the matrix with a 10% probability. This dropout is also applied within the final feed-forward layer. Inside the attention-heads, a further dropout was applied to the iteratively extracted sections of the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ matrices. A final dropout was applied to the attention scores prior to being scaled and the softmax function being applied.

Scale regimes The scale regimes for SASA were originally designed for much larger sequences, whereas these models have a maximum input size of ten tokens. Figure 1 illustrates the different combinations of self-attention head scaling parameters tested.

4.4. Ontologies

This work also explored combining multiple ontologies from various domains of knowledge, rather than a single ontology. Table 1 displays an overview of the ontology combinations, the

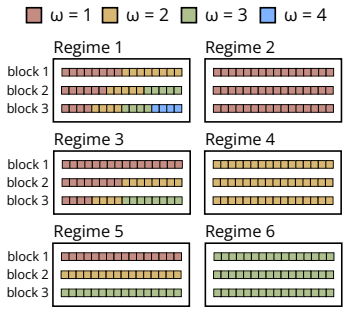

Figure 1: The combinations of scaling parameters used in different block regimes. Each regime was tested separately as just the first block alone, the first and second blocks together, and finally all of the blocks. This was to understand the effect changing the scaling parameters had upon model performance as the depth of the model increased.

Table 1
Ontology combinations used to train models.

Ontologies	Unique concepts	Training examples
Human Phenotype Ontology	16 059	35 969
Human Phenotype Ontology and Mammal Phenotype Ontology	29 370	75 298
Human Phenotype Ontology, Cell Ontology, and Ontology of Host-Pathogen Interactions	29 662	59 175

number of unique concepts represented, and their total number of training examples. One set contained the HPO alone, another both the HPO and the semantically similar Mammal Phenotype Ontology (MPO). The last set comprised three semantically distinct ontologies: the HPO, the Cell Ontology (CLO), and the Ontology of Host-Pathogen Interactions (OHPI). Unique concept IDs in common between ontologies were combined into single concepts.

4.5. Training

In total 81 models were trained (78 MSSA, and 3 NCR) in a python 3.6.13 environment using TensorFlow 2.2.0 [20]. An unresolvable compiler incompatibility prevented the use of FastText and Tensorflow 2 together, as was used in Arbabi *et al.* [11]. Instead, the word embeddings were obtained from the pre-trained ELMo model (v3) using TensorFlow Hub. Nine models (all SAE) were trained in a python 3.9.12 environment with Tensorflow 2.9.1. A bug in the earlier version of Tensorflow prevented the calculation of the means of only non-zero elements for each feature map.

All models were trained with a batch size of 256. After five epochs with no improvement in the training loss the NCR and SAE models would cease training and revert to the best performing parameter weights. For the MSSA models, after five epochs of no improvement, the model parameters revert to the previous best parameter weights, and training resumed with $1/5^{th}$ of the previous learning rate. After the 5^{th} learning rate change, when there was no improvement for five epochs, training was stopped with the model reverting to the best scoring parameter weights. Both the SAE and NCR models had an initial learning rate of $1/512$. The MSSA models have a warm-up period where the learning rate increased linearly from zero to $1/512$ over

Table 2
The evaluation metrics for each NCR and SAE model. The highest scores for each metric are indicated with bold font.

Ontology	Model	Filters	Threshold	Micro Precision	Recall	F-score	Macro Precision	Recall	F-score
HPO	NCR	1024	0.5	78.72	**72.75**	75.62	82.16	74.78	78.29
	SAE	1024	0.55	77.63	72.60	75.03	80.46	**75.04**	77.65
		1536	0.55	80.81	70.89	75.53	83.07	73.49	77.99
		2048	0.7	80.25	70.81	75.24	82.81	73.65	77.96
+ MPO	NCR	1024	0.5	74.12	66.12	69.89	77.38	70.75	73.91
	SAE	1024	0.85	82.28	52.20	63.87	84.28	57.09	68.07
		1536	0.5	76.17	65.23	70.28	78.68	68.90	73.46
		2048	0.5	74.62	61.73	67.56	76.78	66.19	71.10
+ CLO + OHPI	NCR	1024	0.75	**84.04**	64.71	73.12	86.01	66.61	75.08
	SAE	1024	0.6	83.05	70.07	**76.01**	85.13	72.82	**78.50**
		1536	0.5	80.55	69.69	74.73	82.18	72.07	76.79
		2048	0.65	82.94	65.15	72.98	**86.12**	66.96	75.34

Table 3
The evaluation metrics for the best performing MSSA models for each ontology combination.

Ontology	Scale regime	Blocks	Threshold	Micro Precision	Recall	F-score	Macro Precision	Recall	F-score
HPO	4	2	0.45	75.04	71.63	73.30	78.16	74.08	76.06
+ MPO	3	2	0.4	75.00	65.00	69.64	76.89	68.05	72.20
+ CLO + OHPI	2	1	0.5	80.37	68.58	74.01	83.21	70.72	76.46

20 000 iterations.

4.6. Evaluation

Once trained, the models were calibrated and then assessed against an annotated gold-standard 228 PubMed abstract corpus [21] (available here: https://github.com/lasigeBioTM/IHP), an update of the bench-marking dataset created by Groza *et al.* [22]. To calibrate, the models were used to annotate 40 randomly selected abstracts using $n \in \{0.05, 0.10, \dots, 0.95\}$ confidence thresholds. The thresholds with the highest sum of both macro and micro F-scores for each model were then used as the threshold for annotating the remaining 188 abstracts.

5. Results

Table 2 displays the performance metrics for the NCR and SAE models and Table 3 shows the results of best performing MSSA models for each ontology combination. The best performing score for each metric is highlighted in bold.

When trained solely using the HPO, ELMo embeddings improved the two f-scores of the NCR model by at least 4%, compared with the FastText NCR model reported in Arbabi *et al.* [11]. However, the performance declined when additional ontologies were combined with the HPO. The SAE model, trained with diverse domain ontologies, achieved a new SOTA, while only slightly increasing the total number of model parameters.

The MSSA model scores were competitive with the other models, with performance increasing

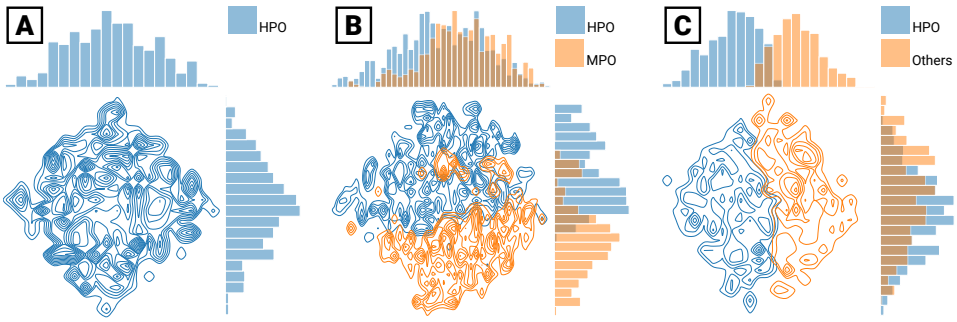

Figure 2: Density-contour plots of the concept embeddings from the best-performing SAE models trained using **A** the HPO (1536 filters), **B** the HPO and MPO (1536 filters), and **C** the HPO, CLO, and OHPI (1024 filters). Principle component analysis reduced dimensions from 1024 to 50, then a tSNE to reduce to two dimensions. The explained variance for each set of embeddings' principal components were 76.02%, 56.05%, and 73.72% respectively.

with the inclusion of additional domain ontologies in the training set. Both Table 2 and 3 show that the models trained using a combination of the HPO and MPO did not perform as well as those with the HPO alone, or in combination with the CLO and OHPI. Figure 2 contains density-contour plots of the concept embeddings for the best performing SAE models for each ontology combination.

6. Discussion

In addition to the SAE model setting a new SOTA for concept recognition, this work reinforces the findings that scaled attention architectures can be competitive with CNNs in low-resource settings.

Training the SAE and MSSA models with diverse domain ontologies resulted in better performance extracting HPO terms. The additional ontologies carry general English-language semantics that may improve the inductive bias. Conversely, when the domains of the ontologies have significant overlap, performance is reduced. In Figure 2 panel **B**, HPO and MPO terms are less clearly separated compared to the more clearly demarcated ontologies in **C**. Although HPO and MPO contain similar concepts with similar natural language names, their ancestry structure is different, which separates these parallel concepts in representational space. To discriminate between them, the models need to attend to more features. This is reflected in the explained variance percentages in Figure 2. Further research regarding how the specific ontology composition and features impact model performance is needed, and assessment may pose a challenge. Currently, we are unaware of any other benchmark dataset annotated with terms from another domain ontology. While the performance of each model will depend upon the language, we expect the scaled attention architecture to be particularly sensitive to syntactical variation. Specific efforts will need to be carried out by fluent native or multi-lingual experts.

Qualitative analysis of our model predictions found that the heuristics, preventing overlapping matches, lead to a not-insignificant number of false negatives. Lobo *et al.* [21] found that 26% of annotated concepts in the gold standard corpora overlap. As a result, Luo *et al.* [12] altered their

heuristics to allow overlap. Indeed, concept overlap is required for multi-domain annotation.

Acknowledgments

Thank you to the reviewers for their kind feedback. This work was funded as part of the Norwich Research Park Biosciences Doctoral Training Partnership, grant number BB/M011216/1, reference code 2243628.

References

[1] S. Fortunato, C. T. Bergstrom, K. Börner, J. A. Evans, D. Helbing, S. Milojević, A. M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi, A. Vespignani, L. Waltman, D. Wang, A.-L. Barabási, Science of science, Science 359 (2018) eaao0185. URL: https://www.science.org/doi/abs/10.1126/science.aao0185. doi:10.1126/science.aao0185.

[2] B. Ali, P. Dahlhaus, The Role of FAIR Data towards Sustainable Agricultural Performance: A Systematic Literature Review, Agriculture 12 (2022) 309. doi:10.3390/agriculture12020309.

[3] L. Cooper, R. L. Walls, J. Elser, M. A. Gandolfo, D. W. Stevenson, B. Smith, J. Preece, B. Athreya, C. J. Mungall, S. Rensing, M. Hiss, D. Lang, R. Reski, T. Z. Berardini, D. Li, E. Huala, M. Schaeffer, N. Menda, E. Arnaud, R. Shrestha, Y. Yamazaki, P. Jaiswal, The Plant Ontology as a Tool for Comparative Plant Anatomy and Genomic Analyses, Plant and Cell Physiology 54 (2013) e1–e1. doi:10.1093/pcp/pcs163.

[4] S. Jupp, T. Burdett, C. Leroy, H. E. Parkinson, A new Ontology Lookup Service at EMBL-EBI., in: SWAT4LS, 2015, pp. 118–119.

[5] B. Eine, M. Jurisch, W. Quint, Ontology-Based Big Data Management, Systems 5 (2017). URL: https://www.mdpi.com/2079-8954/5/3/45. doi:10.3390/systems5030045.

[6] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The FAIR Guiding Principles for scientific data management and stewardship, Scientific Data 3 (2016). doi:10.1038/sdata.2016.18.

[7] E. Tseytlin, K. Mitchell, E. Legowski, J. Corrigan, G. Chavan, R. S. Jacobson, NOBLE–Flexible concept recognition for large-scale biomedical natural language processing, BMC Bioinformatics 17 (2016). doi:10.1186/s12859-015-0871-y.

[8] C. Jonquet, N. Shah, C. Youn, C. Callendar, M.-A. Storey, M. Musen, NCBO Annotator: Semantic Annotation of Biomedical Data, in: International Semantic Web Conference, Poster and Demo session, volume 110, Washington DC, USA, 2009.

[9] E. Batbaatar, K. H. Ryu, Ontology-Based Healthcare Named Entity Recognition from Twitter Messages Using a Recurrent Neural Network Approach, International Journal of Environmental Research and Public Health 16 (2019) 3628. doi:10.3390/ijerph16193628.

[10] X. Dong, S. Chowdhury, L. Qian, Y. Guan, J. Yang, Q. Yu, Transfer bi-directional LSTM RNN for named entity recognition in Chinese electronic medical records, in: 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), IEEE, 2017, pp. 1–4. doi:10.1109/HealthCom.2017.8210840.

[11] A. Arbabi, D. R. Adams, S. Fidler, M. Brudno, et al., Identifying Clinical Terms in Medical

Text Using Ontology-Guided Machine Learning, JMIR Medical Informatics 7 (2019) e12596. doi:10.2196/12596.

[12] L. Luo, S. Yan, P.-T. Lai, D. Veltri, A. Oler, S. Xirasagar, R. Ghosh, M. Similuk, P. N. Robinson, Z. Lu, PhenoTagger: A Hybrid Method for Phenotype Concept Recognition using Human Phenotype Ontology, Bioinformatics 37 (2021) 1884–1890. doi:10.1093/bioinformatics/btab019.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-End Object Detection with Transformers, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), European Conference on Computer Vision, Springer International Publishing, Cham, 2020, pp. 213–229. doi:10.1007/978-3-030-58452-8_13.

[15] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141.

[16] Q. Guo, X. Qiu, P. Liu, X. Xue, Z. Zhang, Multi-scale Self-Attention for Text Classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 7847–7854. doi:10.1609/aaai.v34i05.6290.

[17] W. Zhou, T. Ge, K. Xu, F. Wei, M. Zhou, Scheduled DropHead: A Regularization Method for Transformer Models, arXiv preprint arXiv:2004.13342 (2020). doi:10.48550/arXiv.2004.13342.

[18] Z. Wu, L. Wu, Q. Meng, Y. Xia, S. Xie, T. Qin, X. Dai, T.-Y. Liu, UniDrop: A Simple yet Effective Technique to Improve Transformer without Extra Cost, arXiv preprint arXiv:2104.04946 (2021). doi:10.48550/arXiv.2104.04946.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv:1802.05365 (2018). doi:10.48550/arXiv.1802.05365.

[20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, 2016. URL: https://www.tensorflow.org/. doi:10.48550/arXiv.1603.04467, software available from tensorflow.org.

[21] M. Lobo, A. Lamurias, F. M. Couto, Identifying Human Phenotype Terms by Combining Machine Learning and Validation Rules, BioMed Research International 2017 (2017). doi:10.1155/2017/8565739.

[22] T. Groza, S. Köhler, S. Doelken, N. Collier, A. Oellrich, D. Smedley, F. M. Couto, G. Baynam, A. Zankl, P. N. Robinson, Automatic concept recognition using the Human Phenotype Ontology reference and test suite corpora, Database 2015 (2015). doi:10.1093/database/bav005.