Automated and explainable ontology extension based on deep learning: A case study in the chemical domain

Adel Memariani[1], Martin Glauer[1], Fabian Neuhaus[1,2], Till Mossakowski[1] and Janna Hastings[1,3]

[1] *Otto von Guericke University Magdeburg, Germany*

[2] *Free University of Bozen-Bolzano, Italy*

[3] *University College London, UK*

Abstract

Reference ontologies provide a shared vocabulary and knowledge resource for their domain. Manual construction enables them to maintain a high quality, allowing them to be widely accepted across their community. However, the manual development process does not scale for large domains. We present a new methodology for automatic *ontology extension* and apply it to the ChEBI ontology, a prominent reference ontology for life sciences chemistry. We trained a Transformer-based deep learning model on the leaf node structures from the ChEBI ontology and the classes to which they belong. The model is then capable of automatically classifying previously unseen chemical structures. The proposed model achieved an overall F1 score of 0.80, an improvement of 6 percentage points over our previous results on the same dataset. Additionally, we demonstrate how visualizing the model's attention weights can help to explain the results by providing insight into how the model made its decisions.

Keywords

ontology extension, ontology generation, ontology learning, chemical ontology, Transformers, automated classification, transfer learning, multi-label classification

1. Introduction

Ontologies represent knowledge in a way that is both accessible to humans and is machine interpretable. Reference ontologies provide a shared vocabulary for a community, and are successfully being used in a range of different domains. Examples include the OBO ontologies in the life sciences [1], the Financial Industry Business Ontology for the financial domain [2], and the Open Energy Ontology in the energy domain [3]. While these ontologies differ in many respects, they share one important feature: they are manually created by experts using a process by which each term is manually added to the ontology including a textual definition, relevant axioms, and ideally some additional documentation. Often, this process involves extensive discussions about individual terms. Hence, developing such ontologies is a time-intensive and expensive process. This leads to a challenge for ontologies that cover a large domain.

For example, the ChEBI (Chemical Entities of Biological Interest) ontology [4] is the largest and most widely used ontology for the domain of biologically relevant chemistry in the public

3rd International Workshop on Data meets Applied Ontologies, September 2021 - https://daoxai.inf.unibz.it

domain. It currently (as of June 2021) contains 59,122 fully curated classes, which makes it large in comparison to other reference ontologies. ChEBI is largely manually maintained by a team of expert curators. This is an essential prerequisite for its success, because it enables it to capture the terminology and classification logic shared by chemistry experts. However, the number of chemicals covered by ChEBI is dwarfed by the 110 million chemicals in the PubChem database [5], which itself is not comprehensive. The manually curated portion of ChEBI only grows at a rate of around 100 entries per month, thus will only ever be able to cover a small fraction of the chemicals that are in its domain.

ChEBI tries to navigate this dilemma by extending the manually curated core part of the ontology automatically using the ClassyFire tool [6]. This approach has tripled ChEBI's coverage to 165 000 classes (as of June 2021). However, there are limitations to this approach. Firstly, ClassyFire uses a different underlying classification approach to ChEBI (e.g. conjugate bases and acids are not distinguished), thus, mapping to ChEBI loses classification precision. More importantly, ClassyFire is rule-based and while the extension of the ontology is automated, the creation and curation of the ClassyFire's rules is not. This limits the scalability of this approach.

Somewhat inspired by ChEBI's workflow, we suggest navigating the ontology scaling dilemma by using a new kind of approach to *ontology extension*, which transfers the design decision of an existing ontology analogously to new classes and relations. Our starting point is an existing, manually curated reference ontology. We suggest the use of machine learning methods to learn some of the criteria that the ontology developers adopted in the development of the ontology, and then use the learned model to extend the ontology to entities that have not been covered by the manual ontology development process yet. We will illustrate this approach in this paper for the chemistry use case by training an artificial neural network (with a Transformer-based architecture) to automate the extension of ChEBI with new classes of chemical entities. The approach has several benefits: since it builds on top of the existing ontology, the extension will preserve the manually created consensus. Moreover, the model is trained solely on the content of the ontology itself and does not rely on any external sources. Finally, as we will see, the chosen architecture allows explanation of the choices of the neural network, and, thus to validate the trained model to some degree by manual inspection.

In the next two sections we discuss related work and the overall methodology that we are using to train a model for classifying new classes of chemical entity as subclasses of existing classes in ChEBI.

2. Related Work

In this paper, we present a methodology for ontology extension, which can be considered as a kind of ontology learning. Ontology learning has been an active area of research for more than two decades [7, 8, 9, 10, 11] and a number of automated ontology generators have been developed. A recent publication [11] defined a list of six *desirable goals* for ontology learning methods: they should support expressive languages, require small amount of time and training data, require limited or no human intervention, support unsupervised learning, handle inconsistencies and noise, and their results should be interpretable.

The fundamental make-up of the resulting ontologies varies largely – in part due to different notions of what constitutes an ontology. A survey-based study by Biemann [9] defines three classes of ontologies: *formal*, *prototype-based* and *terminological* ontologies. Most early and data-driven approaches resulted in prototype-based ontologies, in which concepts are not defined in natural language or by logical formulae, but solely by their members. New concepts are often derived from metric-based aggregations such as hierarchical clustering [12]. The quality of the resulting classification depends strongly on the chosen representation of individuals and the criteria for similarity, and may not agree with distinctions that are used by domain experts.

Advances in natural language processing led to a different class of approaches - the *terminological* ontologies. Here, artificial intelligence is used to analyse corpora of relevant literature in order to extract important terms and their relations. Yet, these approaches reflect rather than resolve the inherent ambiguities and differences in language use that exist within different communities of domain experts or even within single communities. The resolution of these ambiguities is an essential part of the ontology development process that involves extensive in-depth communication with and between domain experts [3]. Finally, formal ontologies place a strong emphasis on definitions distinguishing entities, and a rich logical axiomatisation that yields a powerful foundation for reasoning and data integration [13, 14, 15].

While the majority of existing approaches in ontology learning focus on creating new ontologies from scratch, the ones that are dedicated to ontology extension use the ontology as a seed to identify terms that are important for the target domain [16, 17, 18, 19, 20]. These are used to guide approaches that are similar to those that are applied to learn ontologies 'from scratch'. Hence, the resulting extensions are not necessarily based on the principles that have been employed to develop the ontology in the first place, and may potentially introduce biases from the literature into the ontology. Some approaches involve several manual steps, in which experts evaluate concepts and related phrases to sort out these potential issues [21]. Involving human experts has the advantage of providing quality control, but is labour-intensive and costly.

Our approach differs from the existing work in that it employs machine learning techniques but does not rely on text corpora. Rather, it relies only on the content of the ontology that is being extended, in particular on structured annotations.

Our specific application domain is chemical ontology. One characteristic of chemical ontologies is the fact that many classes of chemical entities are annotated with information about their chemical structure. Particularly important for our purposes are annotations in the Simplified Molecular-Input Line-Entry System (SMILES) [22], which is used to represent chemical entities as a linear sequence of characters. The SMILES notation is analogous to a language to describe atoms and their bonds within a chemical entity.

In our approach, we train a deep learning classifier on an existing chemical ontology with structural annotations. The learning method biases the classifier towards the ontology's internal structure, yielding a model that is in line with the domain experts' conceptualisation as represented in the existing ontology. The resulting model is then used to integrate previously unseen classes into the ontology.

This is a novel approach to the problem of chemical classification, which task has historically been approached in multiple different ways [23]. Solutions that involve deep-learning methods were successfully employed for many other applications in chemistry [24], such as the prediction of properties of chemicals [25] or reaction behaviour [26]. Yet, the automated classification of chemicals using deep learning according to an existing ontology has been largely unexplored.

The ClassyFire tool [6] is at the time of writing the most comprehensive method for structure-based automated chemical ontology extension. However, it uses a rule-based and algorithmic implementation that is cumbersome to maintain and is not able to adapt as the underlying ontology changes.

In our previous work [27], we have evaluated several classifiers for this task, including a long short-term memory (LSTM) model which was the best-performing overall. The results of this effort were satisfactory as a whole, but several specific limitations were identified. In particular, the model failed to provide any prediction for a subset of input molecules, and the system as a whole offered no explainability. The current contribution harnesses a Transformer-based architecture and describes how the attention weights of the resulting model can provide insights into how the model made its decisions. Furthermore, by using transfer learning, a broader applicability of this data- and compute-hungry method becomes computationally more feasible.

3. Methodology

Our goal is to train a system that automatically extends the ChEBI ontology with new classes of chemical entities (such as molecules) based on the design decisions that are implicitly reflected in the structure of ChEBI. Thus, for our work we take the 'upper level' of the ontology, which contains generic distinctions, as given. Our focus is the extension of the ChEBI ontology with classes of chemical entities that may be characterised by a SMILES string, i.e., are associated with a specific chemical structure. (These classes are not necessarily leaf nodes in the ontological hierarchy, but nevertheless tend to be in the 'lower' part of the hierarchy.) The learning task for ontology extension may, thus, be characterised as follows: Given a class of chemical entities (characterised by a SMILES string), what are its optimal direct superclasses in ChEBI?

While our goal is – from an ontological point of view – to extend the ChEBI ontology with new classes (i.e., adding new subsumptions), from a machine learning perspective we turned this problem into a classification task, for which we prepare an appropriate learning dataset from the ontology.

Hierarchical chemical classifications should group chemical compounds in a scientifically valid and meaningful way [23, 28]. Each chemical entity has many structural features which contribute to its potential structure-based classification and structures that determine different classes may occur in a single molecule. Thus, ChEBI contains classes that overlap (i.e. share members). The ChEBI ontology provides two separate classification hierarchies for the chemical entities: one based on their structures and another based on their functions or uses. In the current work, we focus on the structure-based sub-ontology. Entities in the structure-based sub-ontology are often associated with specifications of their molecular structures, particularly – but not exclusively – the leaf nodes within the classification hierarchy. In ChEBI, a chemical entity with a defined structure can be the classification parent for another structurally defined entity, since all entities are classes according to the ChEBI ontology, and there can be different levels of specificity even amongst structurally defined classes. To formulate a *supervised* machine learning problem, however, we need to create a distinction between those entities with chemical structures that form the input for learning, and the chemical classes that they belong to that form the learning target. This distinction is created by sampling structurally defined entities

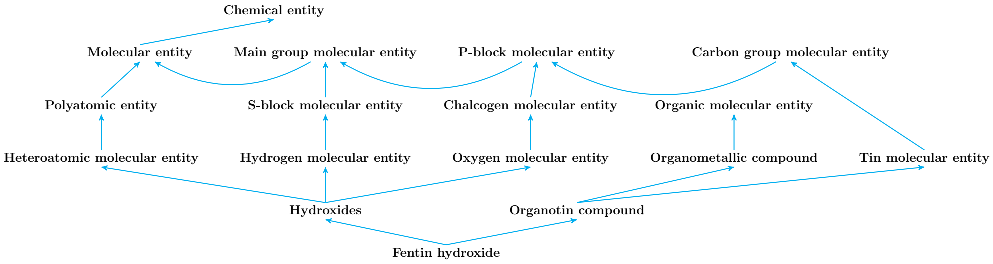


Figure 1: *Fentin hydroxide* and its hierarchical classes. Blue lines indicate the *sub-class* relationships.

only from the ontology leaf nodes.

As mentioned above, the SMILES notation is analogous to a language to describe atoms and their bonds within a chemical structure. Intuitively, this leads to a correspondence between the processing of chemical structures in this type of representation, and natural language processing [29]. Therefore, architectures that have been successfully applied to language-based problems can also be employed for this multi-label prediction task. One of these successful architectures is *Bidirectional Encoder Representations from Transformers* (BERT) [30] – a precursor of the RoBERTa architecture that our approach is based on. The BERT architecture offers a learning paradigm that enables pre-training the model on unlabeled data and then fine-tuning it for the ultimately desired task. Fine-tuning can be done by adding one additional layer to the pre-trained model, without requiring major modifications to the model's architecture. BERT is pre-trained on two unsupervised tasks: A Masked Language Modeling (MLM), in which some tokens are randomly removed from the input sequences and the model will train to predict the masked tokens, and Next Sentence Prediction (NSP), a binary classification task that predicts whether or not the second sentence in the input sequence follows the first sentence in the original text. The RoBERTa model is an extension of the BERT model and it offers several improvements with minor changes in the pre-training strategy. The *Robustly optimized BERT* (RoBERTa) [31] model does not include the NSP part of BERT, and it employs a dynamic masking approach as a replacement of the original masking scheme of BERT. While the original BERT model only applies masking once during data preprocessing, the Roberta model dynamically changes the masking pattern on each training sequence in every epoch. As a result, the model gets exposed to different versions of the same input data with masks on various locations.

Since chemical structures in ChEBI typically belong to several ontology classes, the problem of automated chemical entity categorization can be viewed as a *multi-label* prediction task. Figure 1 shows the *fentin hydroxide* molecule and its parents in the ChEBI ontology: *organotin compound* and *hydroxides*.

Our approach[1] pre-trains a RoBERTa model on SMILES strings, and then predicts multiple chemical class memberships. The overall architecture is shown in Figure 3. This architecture is similar to the one that was used for molecular property prediction in Chithrananda et al. [32].

[1] https://github.com/adelmemariani/chebi-roberta

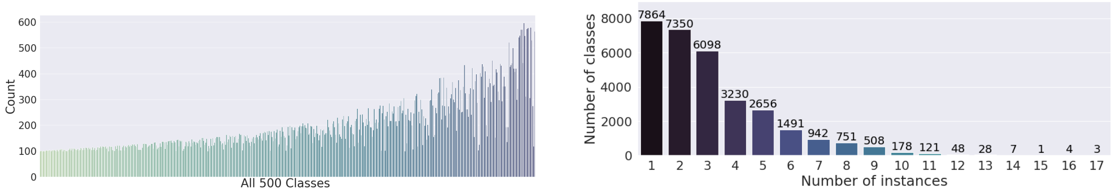


Figure 2: Left: class counts in the dataset. Right: Number of members per number of assigned classes

3.1. Dataset

To use the existing ontology classification as input to the learning task, the ontology first has to be transformed into an appropriate form. The ontology classification is inherently unbalanced, as different classes have different numbers of members and are partially overlapping. It is therefore necessary to define a *sampling strategy* to select leaf node entities and classes to minimize the impact on the training. In order to be able to compare our results to our earlier findings, we have used the same dataset[2] and sampling strategy as was used in Hastings et al. [27]. Using only the hierarchical sub-class relations in the ChEBI ontology, this dataset was created by randomly sampling leaf node molecular entities from higher-level classes that they are subclasses of, using an algorithm that aimed to minimize (as far as possible) class overlap, described in Section 3 of [27]. The resulting dataset contained a total of 500 molecule classes and 31.280 molecules. Despite these balancing measures, it still suffers from certain imbalances. Figure 2 (left) illustrates the number of times each class has appeared in the training and test datasets. As illustrated, some of the classes appeared more frequently than others. Figure 2 (right) shows the number of members per number of associated classes. For example, 7,864 members have just one assigned class, whereas three members have 17 classes assigned. To train, validate and test our model, we divided the dataset into three subsets; a training set containing 21,896 molecules, a validation set of 2,815 molecules, and a test set of 6,569 molecules.

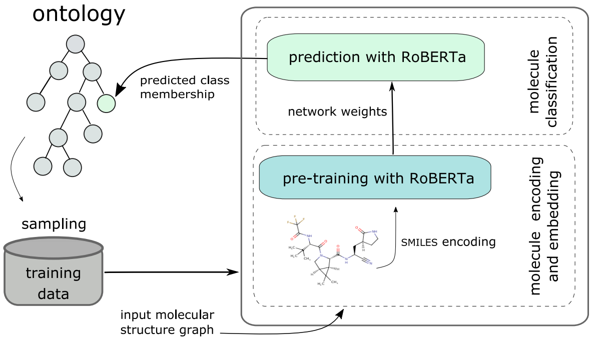
3.2. Input encodings

Tokenization is a pre-processing step used to create a vocabulary from textual data. It is applicable at the character, word, or sub-word level. Pre-trained large-scale word embeddings such as Word2Vec [33] and GloVe [34] employ word tokenization to generate vector representations for words that can encapsulate their meanings, semantic connections, and the contexts in which they are used. Transformer-based models rely on a subword tokenization algorithm that counts the occurrences of each character pair in the dataset and incrementally adds the most frequently occurring pairs to the vocabulary. In our previous work, Hastings et al. [27], we used two strategies to encode the input sequences for the LSTM model: a character-level tokenization and an atom-wise tokenization, where letter combinations that represent an atom were encoded as a token. In the current work, we use the Byte Pair Encoding (BPE) algorithm as a sub-word tokenization method with a RoBERTa architecture.

[2] https://doi.org/10.5281/zenodo.4519815

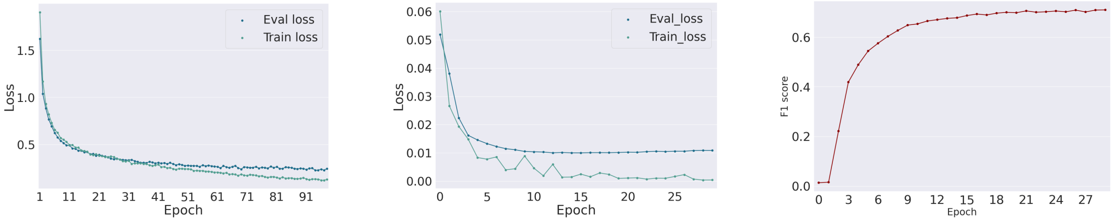
Parameter	Value
Number of attention heads	12
Number of hidden layers	6
Dropout for attention probabilities	0.1
Activation function in the encoder	gelu
Activation for the classification layer	sigmoid
Number of epochs in pre-training	100
Number of epochs in fine-tuning	30
Masked language modeling probability	%15
Batch size	4
Loss function for pre-training	BCELoss
Loss function for fine-tuning	BCEWithLogitsLoss
Optimizer	Adam with weight decay
Number of vocabularies (tokens)	1395
Number of trainable parameters	45,577,728
Tokenizer	BPE

Table 1
Hyper-parameters of the RoBERTa model



ontology
predicted class membership
prediction with RoBERTa
network weights
molecule classification
pre-training with RoBERTa
SMILES encoding
molecule encoding and embedding
sampling
training data
input molecular structure graph

Figure 3
Architecture of our ontology extension approach



(a) (b) (c)

Figure 4: Train and validation loss: (a): pre-training (masked language modeling). (b): fine-tuning (class prediction). (c): F1 score for the validation dataset, during the fine-tuning step.

3.3. Experiment

To train the model, we used a single GPU. Table 1 shows the hyper-parameters for our model. We firstly pre-trained our model based on masked language modelling for 100 epochs *(unsupervised)*. The pre-training step allows the model to discover common patterns in the SMILES strings by attempting to predict the masked tokens using the unmasked tokens. As discussed in Section 3, the pre-trained model provides a proper starting point for training a model on a related desired task. This starting point incorporates the trained weights of the model. Furthermore, We validated the model on a separate dataset after each training epoch. The validation during training has no effect on model's trained weights, nevertheless, it helps in adjusting the model's hyper-parameters. Figure 4 (a) illustrates the loss values for the train and validation sets during the pre-training phase. For the final multi-label classification task, we loaded the pre-trained model and trained it for 30 epochs with the class labels *(supervised)*. Figure 4 (b) shows the train and validation loss during the fine-tuning step. Similarly, Fig. 4 (c) shows the F1 score for the validation dataset during the fine-tuning.

4. Results and Evaluation

For our evaluations during and after training, we used the F1-score as the main measure. The F1 score may be computed in different ways depending on the averaging scheme: (1) *samples*:

calculates the F1 score for each molecule in the test dataset and then computers their average. (2) *micro*: collects the total number of true positives, false positives, and false negatives and calculates the overall F1 score. (3) *macro*: calculates the F1 score for each class and then computes their average. (4) *weighted*: this averaging scheme is similar to the macro F1 score, but it calculates a weight for each class based on the number of true members in each class. Fig. 5 compares the results of our proposed model with the previously obtained results by the LSTM model. The raw output values of our model are the probabilities of a sigmoid function. Therefore, a threshold value must be applied to these probabilities to produce a binary vector, indicating the final classifications. These results are based on the threshold value of 0.5. The precision – in our classification task – shows the ability of the model to not wrongly assign a label to a molecule, while the recall score reflects the ability of the model to discover all labels that were assigned to a molecule.

	Samples		**Macro**		**Micro**		**Weighted**	
	LSTM	**RoBERTa**	**LSTM**	**RoBERTa**	**LSTM**	**RoBERTa**	**LSTM**	**RoBERTa**
F1	0.66	0.76	0.71	0.77	0.74	0.80	0.73	0.79
Recall	0.66	0.75	0.68	0.76	0.70	0.78	0.70	0.78
Precision	0.67	0.77	0.77	0.80	0.79	0.82	0.79	0.82
ROC-AUC	0.83	0.87	0.84	0.89	0.85	0.88	0.85	0.89

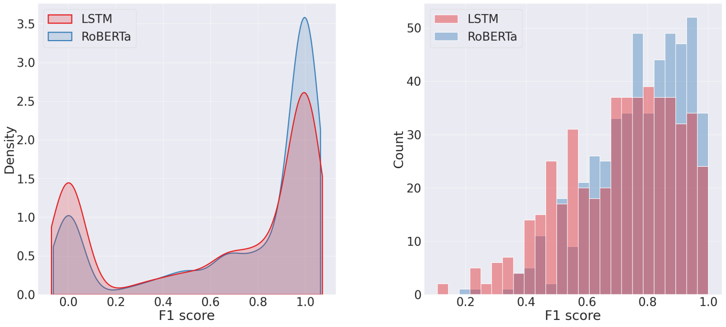


Figure 5: F1 score on test dataset. Left: Kernel density diagram based on the samples (molecules). Right: Histogram diagram based on the labels (classes in the ontology).

Self-attention in Transformer-based models enables the model to explore several locations in the input sequence to produce a better embedding for the tokens. As a result, the embeddings encode different contextual information for the same token in different positions (and different sequences). The architecture of the RoBERTa model contains a stack of Transformers' encoders, each consisting of multiple attention heads. Since the attention heads do not share parameters, each head learns a unique set of attention weights. Intuitively, attention weights determine the importance of each token for the embeddings of the next layers [35]. In this sense, visualizing the attention weights of Transformer-based models helps to interpret the model with respect to the relative importance of different input items for making classifications [36]. While the benefit of attention visualization may be limited in explaining particular predictions, depending on the task, attention can be quite useful in explaining the model's predictions overall [37, 38, 39]. In fact, attention heads can reveal a wide variety of model behaviors and some of these heads may be more significant for model interpretation than others [36]. Our proposed model comprises six layers, each with twelve heads, producing a total of 72 unique attention mechanisms. We examined how attention corresponds to different chemical structural elements, at both the token and molecule level. Figure 6 shows the averaged attention weights of all heads in the last encoder of the model. The most attended sub-structures for each molecule are highlighted with green circles in the molecular graphs. It can be observed that often, the most attention (darker green) is given to

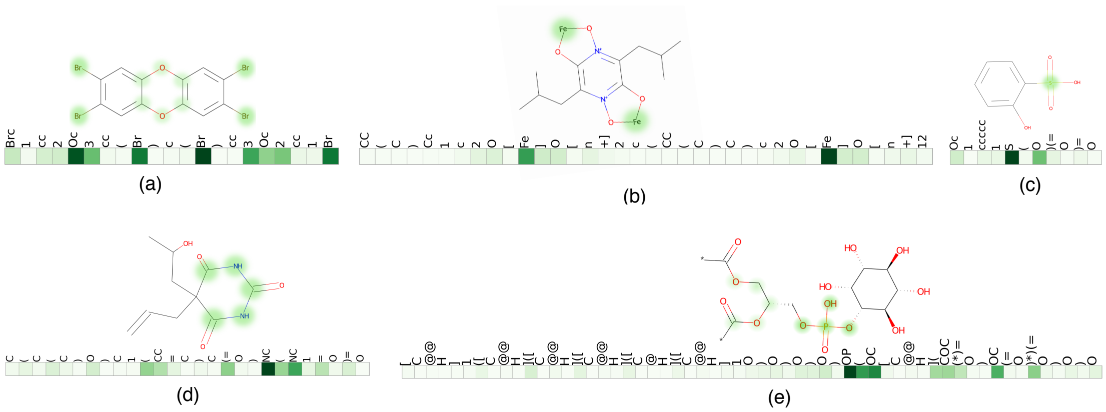


Figure 6: The model predicted class labels for these molecules by attending to influential sub-structures (highlighted in green): (a) organobromine compound (b) iron molecular entity (c) arenesulfonic acid (d) barbiturates (e) phosphatidylinositol

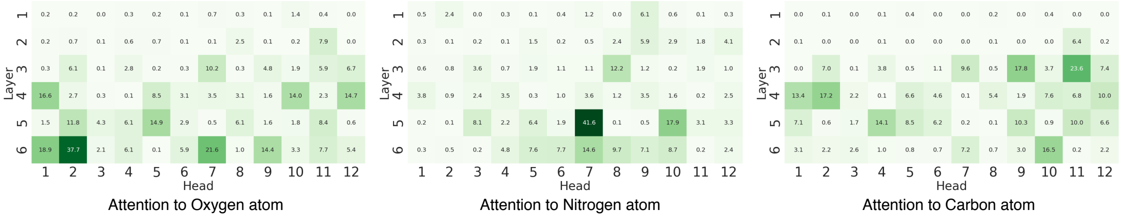


Figure 7: Each cell represents the percentage of all attentions (by each head) that was given to the corresponding token. For example, head 5-6 in (b) dedicated 41.6% of its attention to the Nitrogen atom.

the heaviest atoms, for example *bromine*, *iron* and *sulfur* in Fig. 6 (a), (b) and (c) respectively.

This corresponds to the broad principles of classification in organic chemistry as captured in ChEBI. The predicted classes in Fig. 6 demonstrate that the model learned to assign appropriate labels to the chemical compounds. As illustrated in Fig. 6 (d), the model assigned the *barbiturates* class to the corresponding molecule, which class refers to the family of chemicals that contain a six-membered *ring* structure, which was also the structural element given the most attention. Similarly, Fig. 6 (e) shows that the model focused most on the *phosphate* substructure when assigning the *phosphatidylinositol* class to the molecule.

The presented model takes a given class of molecules, represented by a SMILES string, and assigns the corresponding superclasses from the CHEBI ontology. ChEBI already makes use of an automated tool to extend its coverage beyond the manually curated core, namely ClassyFire. The model can be integrated into the ChEBI development process in the same way. The resulting system can then be used to integrate the given class into the ontology and translate the classification results into subsumption relations. Figure 8 shows the result of this process. The resulting workflow, depicted in 3, allows for the fully automated extension of the ChEBI ontology.

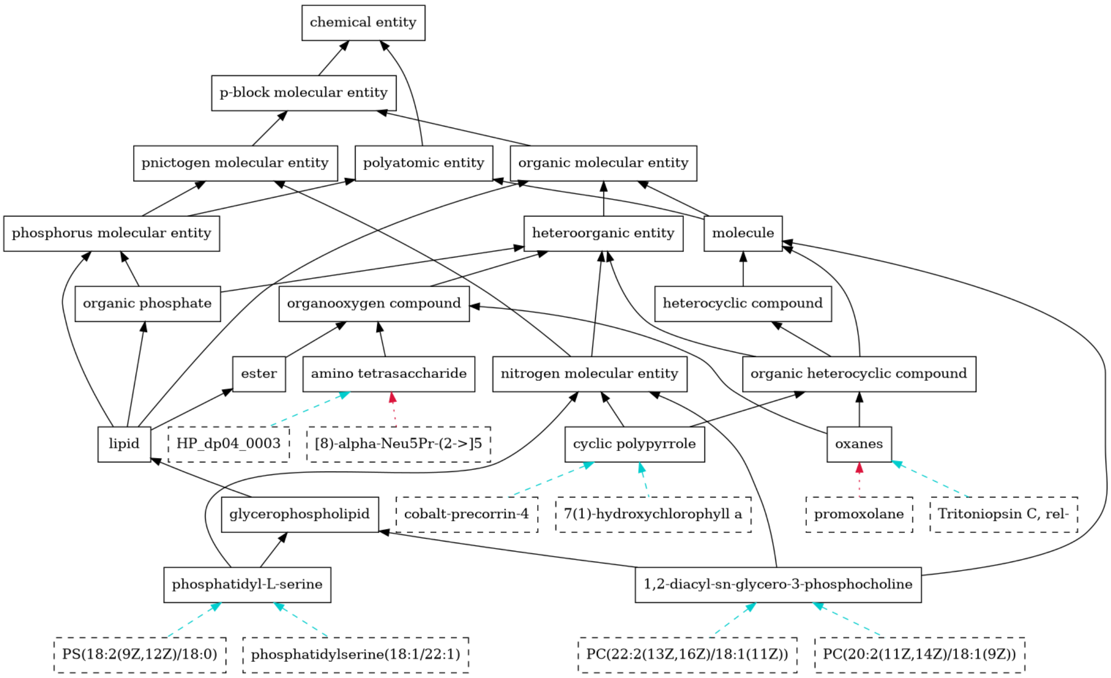


Figure 8: The extended ontology. Existing subsumption relations (black) have been enriched with new subclasses, shown with dashed borders. Correct subclass predictions are depicted with cyan, dashed arrows, while red, dotted arrows indicate misclassifications.

5. Discussion

ChEBI uses ClassyFire, a rules-based system to extend its manually curated reference ontology to chemicals that are not yet covered. This approach has limitations, notably that ClassyFire is structured around a different chemical ontology with only a partial mapping to ChEBI, and ClassyFire's rules are manually maintained. The deep-learning-based approach that we presented can overcome the limitations of rules-based approaches by allowing dynamic creation of classifiers based on a given existing ontology structure. Yet, for optimal applicability, the approach must meet certain quality criteria. Ozaki [11] defined six goals for ontology extension, which we use to structure our discussion of our results.

Handling of inconsistencies and noise Our model is trained on information that originated solely from the ontology itself. This design decision eliminates this external source of inconsistencies and noise. The comparison of the F1 scores in the table in Fig. 5 shows that this classification outperforms the current state-of-the-art approaches - including the formerly leading LSTM-based model. In particular, for those chemical classes that were the most challenging in the previous approach, the current approach performed almost twice as well as shown in Figure 5. It should be noted that there nevertheless remain some chemical classes that perform worse than others. For example, classes that are based on cyclic structures pose challenges, as their information may be scattered around the respective SMILES strings. Alternative input formats and network architectures may be explored in the future to better handle these structures.

The model may also benefit from a larger amount of data. The distribution of class memberships depicted in Figure 2 indicates that the dataset features some classes far more often than others. These classes are more prominent, often by virtue of being higher in the ontology subclass hierarchy and, therefore, represent broader classes of chemicals that may share members with other classes. Such an imbalance can skew the training in favour of those classes. Different sampling and regularization techniques may be explored in the future to address this issue.

Unsupervised learning The presented approach is a variant of ontology extension. The ontology is therefore a mandatory input, from which the information that is needed for the ontology extension is extracted. The resulting dataset does include labels for each molecule. Strictly speaking, it is thus a supervised learning approach. However, these labels are extracted fully automatically from the input – the ontology. Therefore, no additional annotation by experts or other manual data pre-processing is necessary.

Human interaction As the ontology is extended automatically, no interaction is required.

Expressivity The system extends the given ontology using the same ontology language that has been used to build it. ChEBI is developed as an OWL ontology, which comes with expressive OWL-DL semantics.

Interpretability The formerly best classifier was based on an LSTM architecture. This approach outperformed ClassyFire, but this performance came with a disadvantage: The reason for a specific classification was not transparent. This is problematic, because the experts that check the ontology extension need insights into the system's decision processes in order to evaluate the classifications. An explainable approach is therefore crucial. The attention mechanism of the RoBERTa architecture that has been used in the present approach helps to address this issue. Attention weights can be seen as a measure of how much focus is put on an individual token. A homogenous distribution of attention shows that nothing has been focused in particular, whilst high attention on a head shows that a particular token had a high impact. Figure 7 shows that carbon atoms, which are very common in organic chemistry, trigger a low general focus. At the same time, a high focus is put on oxygen atoms, that often indicate functional groups of high classificatory relevance, such as carboxy groups. Figure 6 shows which parts of a particular molecule have been focused on during the classification process. This information can be used to explain the decisions made by the model, raise trust in the prediction system, and aid the experts during the ontology extension process.

Efficiency In [11] 'efficiency' is defined as the time it takes to build the ontology. Once the model is fully trained, the classification which leads to the ontology extension only takes a few minutes. As an example, classification of 6,569 chemical entities in our test dataset took around 10 minutes. While extending the ontology itself is fast, the training of the model requires more time. Training is divided in pre-training and fine-tuning. The pre-training with 100 epochs took around 10 hours. This time is only invested a single time and once a model has been pre-trained, it can be fine-tuned repeatedly for several large sets of molecules and their corresponding classes comparatively quickly. Our final fine-tuning for 30 epochs took around 2 hours.

This analysis shows that the presented approach achieves the goals of ontology learning stipulated in [11]. One additional issue that needs to be addressed is *applicability*. At the heart of the presented approach is a neural network that is trained based on the annotations of the ontology. In the same way as any text analysis approach to ontology generation is dependent on the existence of suitable text corpora, our approach requires that the ontology contains enough information to train a model to predict the superclasses of a new class. ChEBI is an ideal use case, because SMILES annotations provide rich, structured information that we could use for training the model. Another potential application domain for our approach in biology are proteins, which are also classified based on structures, features of which can be annotated in the relevant ontology. Moreover, our approach is not limited to ontologies with structural information represented in annotations. E.g., for ontologies in material science one could consider training the model based on the physical properties (e.g., density, hardness, thermal conductivity), which are typically represented as data properties. In short, our approach to ontology extension is applicable to reference ontologies that associate classes with sufficient information that a neural network may learn the classification criteria that the ontology developers are using.

6. Conclusion and Future Work

We have presented a novel approach to the problem of ontology extension, applied to the chemical domain. Instead of extending the ontology using external resources, we created a model using the ontology's own structured annotations. This transformer-based model can not only classify previously unseen chemical entities (such as molecules) into the appropriate classes, but also provides information about relevant aspects of its internal structure on which the decision is based. At the same time, it was able to outperform previously existing approaches to ontology-based chemical classification in terms of predictive performance.

However, the trained model still struggles with several chemical classes that depend on specific structural features. E.g, classes that exhibit cyclic structures are often found in the lower quantile of classification quality. This behaviour can be traced back to the way molecules are encoded into the SMILES notation. This weakness might be addressed by using architectures that operate directly on the molecular structures, such as Graph Neural Networks [40].

We have illustrated our approach by applying it to the chemical domain, but as we discussed in Section 5, the approach is applicable to any ontology that contains classes that are annotated with information that is relevant to their position in the class hierarchy.

While our approach supports an automatic extension of an ontology, it can also be used in a semi-automated fashion to help ontology developers in their manual curation of the ontology. Since the model is trained based on the content of a manually curated ontology, improving and extending this ontology will lead to better quality training data and, thus, enable better predictions. Hence, there is a potential for a positive feedback loop between manual development and the AI-based extension.

One limitation of our current approach is that it does not use most of the logical axioms of the ontology during the learning process. One strategy to address this gap would be to represent the axioms in the form of Logical Neural Networks [41] in order to detect possible inconsistencies already in the learning process and to penalise them accordingly. Overall, there is still a great need for research in the field of (semi-)automatic ontology extension. Here, the growing field of neuro-symbolic integration can serve as the interface between formal ontologies and the potent solutions of deep learning. This may further the understanding of the inner workings of artificial intelligence and, therefore, raise trust in these systems.

References

[1] 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 (2007) 1251–1255.

[2] D. Allemang, P. Garbacz, P. Grądzki, E. Kendall, R. Trypuz, An analysis of the debate over structural universals, in: F. Neuhaus, B. Brodaric (Eds.), Formal Ontology in Information Systems - Proceedings of the 11th International Conference, FOIS 2021, Bozen-Bolzano, Italy, Frontiers in Artificial Intelligence and Applications, in print.

[3] M. Booshehri, L. Emele, S. Flügel, H. Förster, J. Frey, U. Frey, M. Glauer, J. Hastings, C. Hofmann, C. Hoyer-Klick, et al., Introducing the open energy ontology: Enhancing data interpretation and interfacing in energy systems analysis, Energy and AI 5 (2021) 100074.

[4] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, C. Steinbeck, ChEBI in 2016: Improved services and an expanding collection of metabolites., Nucleic Acids Research 44 (2016) D1214–D1219. doi:doi:10.1093/nar/gkv1031.

[5] Y. Wang, J. Xiao, T. Suzek, J. Zhang, J. Wang, S. Bryant, PubChem: a public information system for analyzing bioactivities of small molecules, Nucl Acids Res 37 (2009) W623–W633. doi:doi:10.1093/nar/gkp456.

[6] Y. Djoumbou Feunang, R. Eisner, C. Knox, L. Chepelev, J. Hastings, G. Owen, E. Fahy, C. Steinbeck, S. Subramanian, E. Bolton, R. Greiner, D. S. Wishart, ClassyFire: automated chemical classification with a comprehensive, computable taxonomy, Journal of Cheminformatics 8 (2016) 61. URL: https://jcheminf.biomedcentral.com/articles/10.1186/s13321-016-0174-y. doi:doi:10.1186/s13321-016-0174-y.

[7] H. Assadi, Construction of a Regional Ontology from Text and its Use within a Documentary System, in: FOIS'98 - 1st International conference on Formal Ontology in Information Systems, volume 46 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, Trento, Italy, 1998, pp. 236–252. URL: https://hal.archives-ouvertes.fr/hal-01617868.

[8] A. Maedche, S. Staab, Ontology learning for the semantic web, IEEE Intelligent systems 16 (2001) 72–79.

[9] C. Biemann, Ontology learning from text: A survey of methods., in: LDV forum, volume 20, 2005, pp. 75–93.

[10] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, H. M. Abbasi, A survey of ontology learning techniques and applications, Database 2018 (2018). URL: https://doi.org/10.1093/database/bay101. doi:doi:10.1093/database/bay101. arXiv:https://academic.oup.com/database/article-pdf/doi/10.1093/database/bay101/27329264/bay101.pdf, bay101.

[11] A. Ozaki, Learning description logic ontologies: Five approaches. where do they stand?, KI-Künstliche Intelligenz 34 (2020) 317–327.

[12] L. Karoui, M.-A. Aufaure, N. Bennacer, Contextual concept discovery algorithm., in: FLAIRS conference, 2007, pp. 460–465.

[13] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyaschev, Ontology-based data access: A survey, in: Proceedings of the Twenty-Seventh International

Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 5511–5519. URL: https://doi.org/10.24963/ijcai.2018/777. doi:doi:10.24963/ijcai.2018/777.

[14] J. Hastings, Primer on Ontologies, in: C. Dessimoz, N. Škunca (Eds.), The Gene Ontology Handbook, volume 1446, Springer New York, New York, NY, 2017, pp. 3–13. URL: http://link.springer.com/10.1007/978-1-4939-3743-1_1. doi:doi:10.1007/978-1-4939-3743-1_1, series Title: Methods in Molecular Biology.

[15] R. Fikes, P. Hayes, I. Horrocks, Owl-ql—a language for deductive query answering on the semantic web, Journal of Web Semantics 2 (2004) 19–29.

[16] W. Liu, A. Weichselbraun, A. Scharl, E. Chang, Semi-automatic ontology extension using spreading activation, Journal of Universal Knowledge Management (2005) 50–58.

[17] S. Althubaiti, Ş. Kafkas, M. Abdelhakim, R. Hoehndorf, Combining lexical and context features for automatic ontology extension, Journal of biomedical semantics 11 (2020) 1–13.

[18] Y. Zhou, L. Zhang, S. Niu, The research of concept extraction in ontology extension based on extended association rules, in: 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS), IEEE, 2016, pp. 111–114.

[19] P. H. Barchi, E. R. Hruschka, Never-ending ontology extension through machine reading, in: 2014 14th International Conference on Hybrid Intelligent Systems, IEEE, 2014, pp. 266–272.

[20] A. Schutz, P. Buitelaar, Relext: A tool for relation extraction from text in ontology extension, in: International semantic web conference, Springer, 2005, pp. 593–606.

[21] H. Li, R. Armiento, P. Lambrix, A method for extending ontologies with application to the materials science domain, Data Science Journal 18 (2019) 1–21.

[22] D. Weininger, Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules, Journal of chemical information and computer sciences 28 (1988) 31–36.

[23] J. Hastings, D. Magka, C. Batchelor, L. Duan, R. Stevens, M. Ennis, C. Steinbeck, Structure-based classification and ontology in chemistry., Journal of cheminformatics 4 (2012) 8. doi:doi:10.1186/1758-2946-4-8.

[24] A. C. Mater, M. L. Coote, Deep learning in chemistry, Journal of chemical information and modeling 59 (2019) 2545–2559.

[25] G. B. Goh, N. O. Hodas, C. Siegel, A. Vishnu, Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties, arXiv preprint arXiv:1712.02034 (2017).

[26] C. W. Coley, W. Jin, L. Rogers, T. F. Jamison, T. S. Jaakkola, W. H. Green, R. Barzilay, K. F. Jensen, A graph-convolutional neural network model for the prediction of chemical reactivity, Chemical science 10 (2019) 370–377.

[27] J. Hastings, M. Glauer, A. Memariani, F. Neuhaus, T. Mossakowski, Learning chemistry: exploring the suitability of machine learning for the task of structure-based chemical ontology classification, Journal of Cheminformatics 13 (2021) 1–20.

[28] C. Bobach, T. Böhme, U. Laube, A. Püschel, L. Weber, Automated compound classification using a chemical ontology, Journal of Cheminformatics 4 (2012) 1–12.

[29] P. Schwaller, T. Gaudin, D. Lanyi, C. Bekas, T. Laino, "found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence

models, Chemical science 9 (2018) 6091–6098.

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

[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[32] S. Chithrananda, G. Grand, B. Ramsundar, Chemberta: Large-scale self-supervised pretraining for molecular property prediction, arXiv preprint arXiv:2010.09885 (2020).

[33] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[34] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[35] J. Vig, A. Madani, L. Varshney, C. Xiong, N. Rajani, Bertology meets biology: Interpreting attention in protein language models. arxiv 2020, arXiv preprint arXiv:2006.15222 (2020).

[36] J. Vig, A multiscale visualization of attention in the transformer model, arXiv preprint arXiv:1906.05714 (2019).

[37] P. Moradi, N. Kambhatla, A. Sarkar, Interrogating the explanatory power of attention in neural machine translation, arXiv preprint arXiv:1910.00139 (2019).

[38] D. Pruthi, M. Gupta, B. Dhingra, G. Neubig, Z. C. Lipton, Learning to deceive with attention-based explanations, arXiv preprint arXiv:1909.07913 (2019).

[39] S. Serrano, N. A. Smith, Is attention interpretable?, arXiv preprint arXiv:1906.03731 (2019).

[40] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE transactions on neural networks 20 (2008) 61–80.

[41] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, et al., Logical neural networks, arXiv preprint arXiv:2006.13155 (2020).