Text-to-Ontology Mapping via Natural Language Processing Models

Uladzislau Yorsh[1], Alexander S. Behr[2], Norbert Kockmann[2] and Martin Holeňa[1,3,4]

[1]Faculty of Information Technology, CTU, Prague, Czech Republic
[2]Faculty of Biochemical and Chemical Engineering, TU Dortmund University, Germany
[3]Institute of Computer Science, Czech Academy of Sciences, Prague, Czech Republic
[4]Leibniz Institute for Catalysis, Rostock, Germany

Abstract

The paper presents work in progress attempting to solve a text-to-ontology mapping problem. While ontologies are being created as formal specifications of shared conceptualizations of application domains, different users often create different ontologies to represent the same domain. For better reasoning about concepts in scientific papers, it is desired to pick the ontology which best matches concepts present in the input text.

We have started to automatize this process and attack the problem by utilizing state-of-the-art NLP tools and neural networks. Given a specific set of ontologies, we experiment with different training pipelines for NLP machine learning models with the aim to construct representative embeddings for the text-to-ontology matching task. We assess the final result through visualizing the latent space and exploring the mappings between an input text and ontology classes.

Keywords

text analysis, language models, fastText, BERT, matching text to ontologies

1. Introduction

The FAIR (Findable, Accessible, Interoperable and Reusable) research data management needs a consistent data representation in ontologies, particularly for representing the data structure in the specific domain [1]. The application of ontologies varies from a domain-specific vocabulary and a translation reference up to an environment for logical reasoning and property inference.

Despite their purpose of standardizing the knowledge conceptualization, there still may exist several ontologies within the same domain [2]. Creating and managing an ontology is a manual process often performed by many domain experts. As each expert works on different problems, they also might have different conceptualizations of their respective knowledge. However, approaches to automate the knowledge conceptualization also face their challenges, as a machine cannot easily create semantics without human input (e.g. scientific theses, which are created by humans). A constant demand for a knowledge database expansion and utilizing of already available knowledge leads to the problem of ontology alignment and merging, which is a research field on their own.

Another problem faced by domain experts is how to choose a proper ontology for a certain task. Different ontologies can focus on different sub-domains as well as on different levels of abstraction. Choosing the ontology which best corresponds to an input text is an important step towards reasoning about it.

In the reported work in progress, we focus on the latter problem. One of the possible ways to address the task is to consider it as matching input texts with an existing text collection. Such a formulation allows to employ already existing rich text processing pipelines, as well as powerful pretrained models.

2. Related Work

2.1. Entity linking

The problem is closely related to the *concept normalization* and *entity linking* tasks. The algorithms encountered in this context include dictionary lookup [3, 4], conditional random fields and tf-idf vector similarity [5], word embeddings and syntactical similarity [6].

The vector similarity approaches either employ tf-idf vectors or dense word embeddings. The tf-idf vector is a document vector of the size of the considered vocabulary, where each element is the number of occurrences of the term in a document, multiplied by the logarithmized reciprocal value of the number of the documents where this term appears. These vectors are well-interpretable (high values indicate the rare term which appears in particular document often), but very sparse, which impedes the performance of machine learning algorithms. On contrary, word embeddings generated by representation

ITAT'22: Information technologies – Applications and Theory, September 23–27, 2022, Zuberec, Slovakia
yorshula@fit.cvut.cz (U. Yorsh);
alexander.behr@tu-dortmund.de (A. S. Behr);
norbert.kockmann@tu-dortmund.de (N. Kockmann);
martin@cs.cas.cz (M. Holeňa)

learning algorithms are dense, but provide no direct interpretation.

The mentioned systems share a common pipeline—at the first step, they use an external algorithm to find potential concepts in a scientific text. After that, they link proposals with concepts using retrieval techniques, such as dictionary lookup or vector distance.

2.2. Natural Language Processing

Entity linking techniques relying on vector similarity may either use tf-idf vectors or word embeddings. The latter may be beneficial due to the dense vector structure and an ability to be produced by high-capacity language models, trained on large corpora.

fastText [7] is a representation learning algorithm producing word-level embeddings. A neural network with a single hidden layer is being trained to predict a word given its context, and the learned word representations are then being used as word embeddings.

Another widely used representation learning algorithm is **BERT** [8]. A deep sequence processing neural network is trained on two objectives—predicting a masked word in a sentence and predicting the order of two given sentences.

Compared to the fastText, BERT embeds the whole input sequence at once and produces *contextual* embeddings for each token—the same token in different contexts will be embedded differently. This allows it to achieve state-of-the-art results in text classification [8] and named entity recognition [9] tasks. Another benefit of BERT is that its Transformer architecture demonstrates impressive transfer-learning capabilities [10], which can be useful for fine-tuning the model for tasks laying outside pretraining data distribution.

3. Matching Texts to Ontologies

3.1. Problem definition

Within the proposed framework, we define an ontology O as a directed attributed multi-graph, where vertices represent classes, edges represent relationships between them, and both vertices and edges can have attributes. Given a set of specific ontologies $K = \{O_1, \ldots, O_n\}$ and an input text $T \in \mathbf{T}$, the task is to predict the ontology that best matches the content of T. A predictor may be either a "hard" mapping $f : \mathbf{T} \mapsto K$ or a scoring function $f : \mathbf{T} \times K \mapsto \mathbb{R}$ which allows to order ontologies by relevance.

There are several complications of the task:

Given ontologies are the only source of supervision. No text-to-ontology mapping labels are provided. This is the key difference from many other works, which rely on ground-truth either for training or evaluation.

Ontologies may significantly differ in size. This can lead to very outbalanced datasets when generating them from ontologies.

These difficulties should be considered in the first place when choosing a solution method.

3.2. Text Similarity Strategy

Ontologies typically provide annotations for most of their classes and relations, potentially generating supervised datasets for ML algorithms. But before employing a text similarity approach, we have to make several strong assumptions:

- The distribution of input texts is the same as the distribution of annotation texts. It means that the input sentences should follow the same general structure, length and vocabulary as ontology annotations to avoid prediction skewing for irrelevant reasons.
- The best matching ontology is the one which provides annotations most similar to the input text. Since the considered methods are text-based, they will not rely on structures or hierarchies created by ontology classes and input text terms.

For methods mentioned below in this subsection, we will employ fastText and BERT models trained on texts from related domains, which will serve as a backbone for further processing. Following the notation introduced in the Subsection 3.1, we consider a "hard" mapping $f : \mathbf{T} \mapsto K$ directly to the space of ontologies of interest.

3.2.1. Zero-shot classification

The method consists of assigning an ontology considering a similarity between annotation embeddings and an embedding of an input text. The method is simple and does not require model fine-tuning, which allows to quickly establish a baseline for other experiments. The common choices of similarity measures are Euclidean or cosine distances – we choose the latter in our experiments. The reason is that for some embedding algorithms vector length may be influenced by the input text size, so vectors corresponding to semantically close texts may generally point in the same direction but be dissimilar in terms of Euclidean distance.

3.2.2. Supervised classification based on ontology annotations

This method relies on a supervision provided by ontology annotation attributes. Given an ontology set K, we

can generate a dataset of annotation-ontology label pairs and use it for supervised training. Under the aforementioned assumptions we can directly assign input texts to ontologies using the trained model.

3.2.3. Negative sampling

This method extends the method above by adding a "None" class, denoting that the input text does not relate to any of given ontologies. The annotation dataset is extended by:

- Sentences extracted from scientific papers from unrelated domains and labeled with the "None" label.
- Sentences extracted from papers from related domains with a different objective during training. For related input texts, instead of maximizing the model output scores for a ground truth class we minimize the output scores for the "None" class. This method is intended to partially counter the possible input distribution difference between ontology annotations and scientific texts.

4. Experiments

4.1. Setup

We conduct our experiments on a set of five ontologies related to the chemical domain (Table 1). The ontologies NCIT, CHMO and Allotrope are considered to be the closest to it, while Chemical Entities of Biological Interest (CHEBI) has only a subset of relevant entities. The SBO was selected as it contains some general laboratory and computational contexts, which can be seen as some kind of a test, whether the tools used can also identify ontologies not fitting to the text content.

We also selected 28 scientific papers as inputs for assessment, consisting of 25 research and 3 review papers. Those papers deal with the topic of methanation of CO2 and consist in sum of 1,3M symbols.

Table 1
Sizes of considered ontologies

Ontology	Classes	Annotations
CHEBI [11]	171058	51095
NCIT [12]	170300	133478
Allotrope [13]	2893	2677
CHMO [14]	3084	2895
SBO [15]	693	692

We use the pretrained fastText model by [16] and the `recobo/chemical-bert-uncased` [17] checkpoint of a BERT implementation [18] from the HuggingFace repository. For preprocessing we use spaCy [19] with a scispaCy [20] model `en_ner_bc5cdr_md`. For the remaining machine learning models, PyTorch implementations were used. For 3D visualization method Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [21], we used the implementation described in [22].

Due to the lack of ground truth matching data, we assess the performance primarily through inspecting the resulting input sentence-annotation pairs.

4.1.1. Text preprocessing

We employ the following text preprocessing pipeline before constructing input embeddings:

1. *Split an input text into sentences with a spaCy model.
2. *Filter valid sentences, which contain at least two nouns and a verb.
3. *Filter out sentences with non-paired parenthesis and ill-parsed formulas or composed terms.
4. (BERT) Tokenize with a tokenizer coming with the model.
4. (fastText) Convert to lowercase and split into words

The points marked with an asterisk are meant to be applied to new sentences from scientific papers only.

4.2. Text Similarity

Zero-shot setup. We start with representation learning of annotations using the fastText and BERT algorithms and inspecting the embeddings produced. For the dimensionality reduction, we use the UMAP algorithm with the number of neighbors set to 15, minimum distance 0.5 and cosine metric. We have found that 3-dimensional embeddings preserve substantially more information (allowing to separate clusters that may be inseparable in 2D). The result is illustrated in Figure 1, three example sentences together with annotations assigned to them by fastText and BERT are in Table 3.

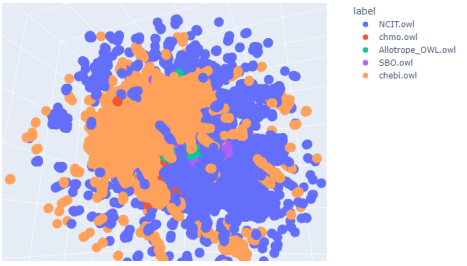
Table 2
Zero-shot statistics for the distances of sentences to the closest ontology annotations.

Embeddings	Closest distance mean	Closest distance standard deviation
fastText	0.846	0.086
BERT	0.605	0.038

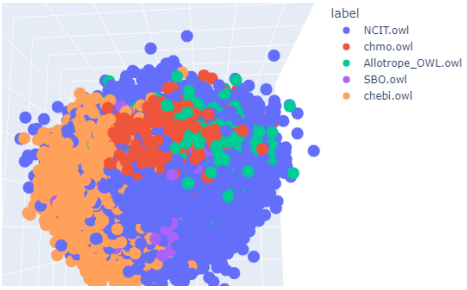
Those visualizations and Table 2 allow to suppose that the model embeds input papers separately from ontology annotations, which may indicate a distribution shift between sentences and annotations.

Table 3
Sentence pairs of a new sentence from the scientific papers and the closest ontology annotation. The "carbon dioxide" annotation was assigned by BERT to all three above example new sentences. While BERT embeddings are more discriminative for the ontology classification task, the assigned sentences and low-dimensional embeddings on Figure 3 indicate that this approach is more sensitive to the distribution shift problem.

New sentence	Also there is an upper limit of operation above which thermal decomposition will occur.	The difference is the main adsorption species during the reaction.	This enhancement of the Ni dispersion is very relevant because as reported in the literature [78] NiO sites [...]
fastText closest	An end event specification is an event specification that is about the end of some process.	Reaction scheme where the products are created from the reactants [...]	The name of the individual working for the sponsor responsible for overseeing the activities of the study.
BERT closest	Carbon dioxide gas is a gas that is composed of carbon dioxide molecules.	Carbon dioxide gas is a gas that is composed of carbon dioxide molecules.	Carbon dioxide gas is a gas that is composed of carbon dioxide molecules.



(a) fastText



(b) BERT

Figure 1: A 3-dimensional projection of annotation embeddings produced by fastText and BERT. In the case of fastText, SBO, Allotrope, and CHMO annotations are located in tiny areas, primarily close to the center of the image.

Ontology matching as text classification. As we mentioned in the Subsection 3.2, another potential strategy to solve the problem is to treat it as a classification task. If the distributions of input texts and corresponding ontologies are the same, we can train a classifier on ontology annotations and apply it on input texts.

We implement this by embedding ontology annotations with BERT and training over them a shallow fully-connected multilayer perceptron (MLP) with a single 768-dimensional hidden layer. Due to the significant difference in sizes between ontologies, we proportionally oversample minority data points. The classifier reaches 0.987 validation accuracy after the single-shot validation on the annotations from all the classes, which indicates their good separability for different ontologies, cf. Figures 2 and 3.

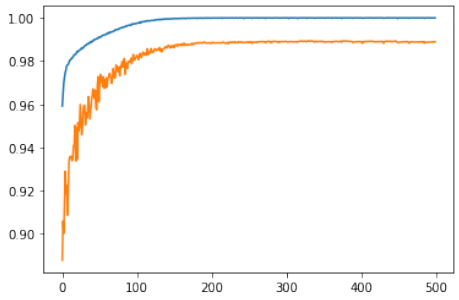
However, if we preprocess input texts and embed them in this way, the inspection will show that their distribution significantly differs from the distribution of ontology annotations. The visualizations in Figures 2 and 3 show a dense separate cluster of sentences parsed from scientific papers.

Negative sampling. As an attempt to counter the issue, we introduced scientific texts into training data. We sampled 400 scientific texts from the chemical domain (as positive examples) and 400 from unrelated domains (as negatives). During training, the model is being trained on two objectives:

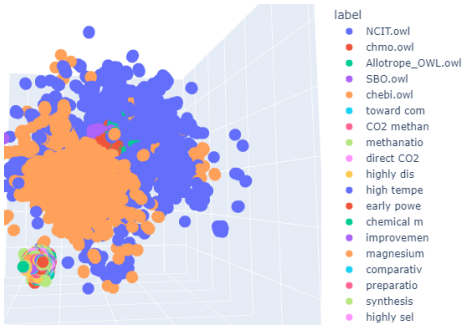
1. Cross-entropy loss if the input is an ontology annotation (same as before)
2. Binary cross-entropy loss if the input is a sentence from a scientific paper. The model minimizes the probability of a special "Negative" class output for a related scientific text, and maximises it for unrelated.

In this setting we train the head over BERT until convergence first, leaving the backbone frozen. Considering only ontology annotations and leaving aside sampled sentences, the model reaches 0.984 validation accuracy, which is very similar to the performance of the classifier described above.

After that, we fine-tune the whole BERT model. The model reaches 0.958 validation accuracy after single-shot validation on the combined annotation and paper sentence dataset, with the confusion matrix on Figure 4. As



(a) The progress of training and validation accuracy during training. The blue (above) and orange (below) lines indicate the training and validation accuracy respectively.



(b) Annotation from ontologies and sentences from 14 scientific papers embedded by BERT

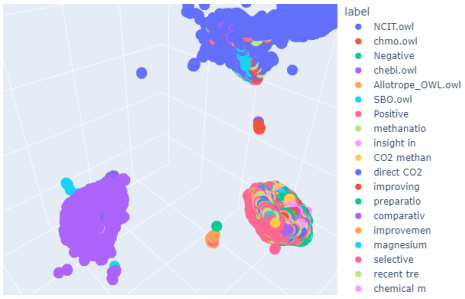
Figure 2: Training plot and a 3-dimensional projection of the embeddings produced by BERT in the classification approach. The visualization gives an intuition of the distribution gap between the scientific texts for which we would like to find the most relevant ontology, and ontology annotations.

we will show later, mixing sampled sentences in from both relevant and irrelevant scientific texts allowed to improve classification accuracy over the classifier on top of BERT.

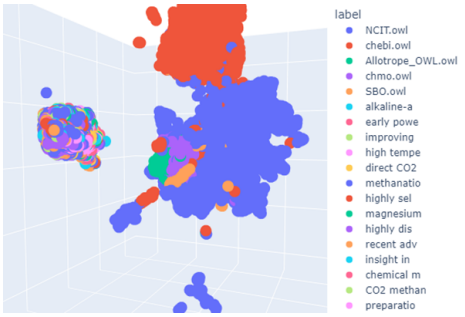
Despite the good separability of individual ontologies and the additional optimization criterion, the UMAP embeddings look similar to the previous setup in terms of clustering input sentences into a separate subspace.

It is worth to note that the classifier and negative sampling models produce softmax scores, which can be interpreted as a class probability distribution. However, neural networks tend to be overconfident in their outputs [23], so additional calibration is needed before using the outputs for relevance estimation.

Statistical results. To compare the models, we conduct the Friedman test first to check if the models perform the same. We perform a stratified split of the validation dataset with the ontology annotations into 50 samples



(a) 3-dimensional projection of the embeddings produced by the fine-tuned BERT



(b) 3-dimensional projection of the activities of the hidden layer of the MLP trained over BERT

Figure 3: Visualization of the BERT embedding phase and the MLP classification phase of the ontology classification task with the fine-tuned BERT in the negative sampling setting.

and test the following hypothesis:

Hypothesis H_1 (Null)**:** All the six models perform the same on the validation splits.

The Friedman test resulted in the null hypothesis rejection on the significance level of 5%. To further compare the models, we perform the Wilcoxon signed-rank test on each pair of models. We make the following assumptions about the algorithms:

- For a larger k the kNN classifier can work the same or better than the 1NN.
- The neural network model can fit training data the same or better than the kNN.
- The negative sampling results in a non-decrease or an improvement in the model generalization.

Hypothesis H_2 (Null for kNN models)**:** The 10NN models perform the same as their 1NN variants.

While the 1NN is a common setting for many NLP systems, it may produce complex decision boundaries and lead to overfitting. We test a larger k versus one to determine whether this is an issue in our setup.

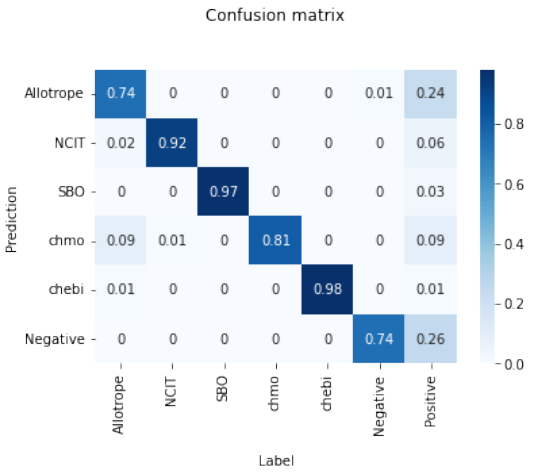


Figure 4: Confusion matrix of the MLP classification over fine-tuned BERT for a dataset consisting of the annotations from all five considered ontologies and the sentences of the additional 400 related and 400 unrelated scientific papers..

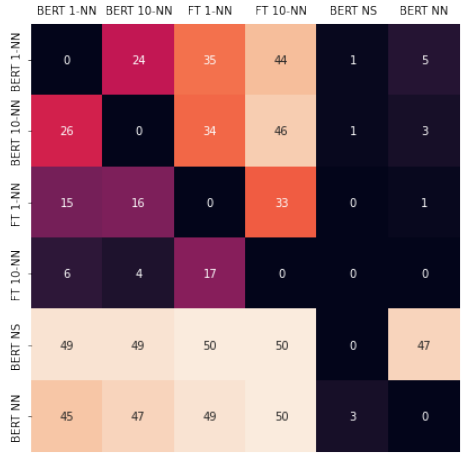


Figure 5: The comparison matrix of the six considered models. The i, j-th element indicates an amount of splits where the i-th model performed better than j-th. Except the one- and ten-nearest-neighbor over BERT embeddings, all the models demonstrate statistically significant differences. BERT NN denotes a neural network classifier trained over BERT embeddings.

Hypothesis H_3 (Null for neural network classifier)**:** The NN classifier performs the same as the kNN models both on BERT/fastText embeddings.

The assumption behind this hypothesis is that a neural network as a universal approximator can fit data better than a nearest-neighbour classifier.

Hypothesis H_4 (Null for the fine-tuned model with negative sampling)**:** The fine-tuned BERT with negative sampling performs the same as other considered models.

We suppose that additional sampled sentences would allow to improve the model performance and help to avoid overfitting when fine-tuning the whole model instead of head only.

Hypothesis H_5 (Null for the rest)**:** In each remaining pair, both models have the same performance.

We indicate the relative model performance on Figure 5. Considering the 5% significance level, the test rejected all the null hypotheses except the H_2, which was rejected only for the fastText embeddings. To explain that, we can note that there is a relatively sharp boundary between individual classes on UMAP embeddings. If it holds so for the original space, larger k may suppress outlier noise but decrease classification accuracy near it.

5. Conclusion and Further Research

We are not aware of other works on unsupervised text-to-ontology mappings, so we are not able to discuss them and compare the proposed approach with previous methods.

The reported work in progress revealed that the distribution of the scientific texts substantially differs from the one of ontology annotations. In spite of the high classification accuracy both for the annotations from the considered ontologies and the sentences of the additional 800 scientific papers, this leads to mapping into separate subsets of the embedding space. This is true even for the most sophisticated of the three investigated settings – with the BERT fine-tuned using both the ontology annotations and scientific texts from (un-)related domains.

To avoid such a loss of generality, the future research could include an intermediate step of entity recognition. Using such recognized entities instead of raw text can help to separate the information in scientific papers that is directly related to concepts from ontologies and unrelated words, sentences and other parts of text not eliminated during preprocessing.

Acknowledgments

The research reported in this paper has been supported by the German Research Foundation (DFG) funded projects 467401796 and NFDI2/12020.

References

[1] M. Wolf, J. Logan, K. Mehta, D. Jacobson, M. Cashman, A. M. Walker, G. Eisenhauer, P. Widener, A. Cliff, Reusability first: Toward fair workflows, in: 2021 IEEE International Conference on Cluster Computing (CLUSTER), 2021, pp. 444–455. doi:10.1109/Cluster48925.2021.00053.

[2] J. Grühn, A. S. Behr, T. H. Eroglu, V. Trögel, K. Rosenthal, N. Kockmann, From coiled flow inverter to stirred tank reactor – bioprocess development and ontology design, Chemie Ingenieur Technik 94 (2022) 852–863. doi:https://doi.org/10.1002/cite.202100177.

[3] L. Hirschman, M. Krallinger, A. Valencia, J. Fluck, H.-T. Mevissen, H. Dach, M. Oster, M. Hofmann-Apitius, Prominer: Recognition of human gene and protein names using regularly updated dictionaries, Proceedings of the Second BioCreAtIvE Challenge Evaluation Workshop (2007) 149–151.

[4] A. A. Morgan, Z. Lu, X. Wang, A. M. Cohen, J. Fluck, P. Ruch, A. Divoli, K. Fundel, R. Leaman, J. Hakenberg, C. Sun, H.-H. Liu, R. Torres, M. Krauthammer, W. W. Lau, H. Liu, C.-N. Hsu, M. Schuemie, K. B. Cohen, L. Hirschman, Overview of BioCreative II gene normalization, Genome Biol 9 Suppl 2 (2008) S3.

[5] R. Leaman, R. Islamaj Dogan, Z. Lu, DNorm: disease name normalization with pairwise learning to rank, Bioinformatics 29 (2013) 2909–2917.

[6] İ. Karadeniz, A. Özgür, Linking entities through an ontology using word embeddings and syntactic re-ranking, BMC Bioinformatics 20 (2019) 156. URL: https://doi.org/10.1186/s12859-019-2678-8. doi:10.1186/s12859-019-2678-8.

[7] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606 (2016).

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the NAACL, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[9] Z. Liu, F. Jiang, Y. Hu, C. Shi, P. Fung, NER-BERT: A pre-trained model for low-resource entity tagging, CoRR abs/2112.00405 (2021). URL: https://arxiv.org/abs/2112.00405. arXiv:2112.00405.

[10] K. Lu, A. Grover, P. Abbeel, I. Mordatch, Pretrained transformers as universal computation engines, CoRR abs/2103.05247 (2021). URL: https://arxiv.org/abs/2103.05247. arXiv:2103.05247.

[11] 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 Res 44 (2015) D1214–9.

[12] National cancer institue thesaurus, 2022. URL: https://bioportal.bioontology.org/ontologies/NCIT.

[13] Allotrope foundation ontologies, 2022. URL: https://www.allotrope.org/ontologies.

[14] Systems biology ontology, 2022. URL: https://github.com/EBI-BioModels/SBO.

[15] Chemical methods ontology, 2022. URL: https://obofoundry.org/ontology/chmo.html.

[16] E. Kim, Z. Jensen, A. van Grootel, K. Huang, M. Staib, S. Mysore, H.-S. Chang, E. Strubell, A. McCallum, S. Jegelka, E. Olivetti, Inorganic materials synthesis planning with literature-trained neural networks, Journal of Chemical Information and Modeling 60 (2020) 1194–1201. URL: https://doi.org/10.1021/acs.jcim.9b00995. doi:10.1021/acs.jcim.9b00995.

[17] Bert for chemical industry, 2022. URL: https://huggingface.co/recobo/chemical-bert-uncase.

[18] Bert, 2022. URL: https://huggingface.co/docs/transformers/model_doc/bert.

[19] M. Honnibal, I. Montani, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017.

[20] M. Neumann, D. King, I. Beltagy, W. Ammar, ScispaCy: Fast and robust models for biomedical natural language processing, in: Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Florence, Italy, 2019, pp. 319–327. URL: https://aclanthology.org/W19-5034. doi:10.18653/v1/W19-5034.

[21] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL: https://arxiv.org/abs/1802.03426. doi:10.48550/ARXIV.1802.03426.

[22] L. McInnes, J. Healy, N. Saul, L. Grossberger, Umap: Uniform manifold approximation and projection, The Journal of Open Source Software 3 (2018) 861.

[23] Y. Gal, Uncertainty in Deep Learning, Ph.D. thesis, University of Cambridge, 2016.