Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings

Erik B. Myklebust [a,b,*] Ernesto Jiménez-Ruiz [c,b,**], Jiaoyan Chen [d], Raoul Wolf [a] and Knut Erik Tollefsen [a,e]

[a] *Norwegian Institute for Water Research, Oslo, Norway*
[b] *SIRIUS, University of Oslo, Oslo, Norway*
[c] *City, University of London, London, United Kingdom*
[d] *University of Oxford, Oxford, United Kingdom*
[e] *Norwegian University of Life Sciences, Ås, Norway*

Abstract. Semantic Web technologies enable the interoperability of disparate data sources. We have created a knowledge graph based on major data sources used in ecotoxicological risk assessment. This facilitates the use of the extensive library of Semantic Web tools. We have applied this knowledge graph to an important task in risk assessment, namely chemical effect prediction. We have evaluated nine knowledge graph embedding models from a selection of geometric, decomposition, and convolutional models on this prediction task. We show that using knowledge graph embeddings can increase the accuracy of effect prediction with neural networks. Furthermore, we have implemented a fine-tuning architecture which adapts the knowledge graph embeddings to the effect prediction task and leads to a better performance. Finally, we evaluate certain characteristics of the knowledge graph embedding models to shed light on the individual model performance.

Keywords: Knowledge graph, ecotoxicology, risk assessment, adverse effects, embedding, chemicals, species

1. Introduction

Ecotoxicology is a multidisciplinary field that studies the potentially adverse toxicological effects of chemicals on organisms, starting at molecular level to individuals, sub-populations, communities and ecosystems. One major societal contribution of ecotoxicology is ecological risk assessments, which compare environmental concentrations of chemicals with existing laboratory effect data to evaluate the ecosystem health status. While laboratory experiments are thus crucial, they are both labour intensive and result in a high number of animal testing. Therefore, the development of modelling techniques for extrapolating from existing laboratory effect data is a major effort in the field of ecotoxicology.

A very important challenge in ecotoxicology risk assessment is the interoperability of the relevant data which is typically available in disparate data sources and formats and described using different vocabularies. The use of Semantic Web technologies and (RDF-based) knowledge graphs [7] can address this challenge and facilitate the orchestration of these datasets. Hence, extrapolation or prediction models can benefit from an integrated view of the data and the background knowledge provided by a knowledge graph. The use of knowledge graphs also enables the use of the available infrastructure to perform automated reasoning, explore the data via semantic queries, and compute semantic embeddings for machine learning prediction.

*Corresponding Author: Erik B. Myklebust, Norwegian Institute for Water Research, Gaustadalléen 21, 0349 Oslo, Norway. E-mail: ebm@niva.no
**Corresponding Author: Ernesto Jiménez-Ruiz, City, University of London, College Building, Northampton Square, London EC1V 0HB. E-mail: ernesto.jimenez-ruiz@city.ac.uk

1570-0844/0-1900/$35.00 © 0 – IOS Press and the authors. All rights reserved

In this work we have created the Toxicological Effect and Risk Assessment Knowledge Graph (TERA) and implemented a prediction model over this knowledge graph to extrapolate adverse biological effects of chemicals on organisms. Here, we limit ourselves to binary effect prediction of mortality (shortened to effect prediction), *i.e.*, where there is a chance that a chemical can affect a species in a lethal way. The work and evaluation conducted in this paper is driven by the following research question: *does the use of contextual information in the form of knowledge graph embeddings brings added value in the prediction of adverse biological effects?*

Our contributions can be summarized as follows:

(i) TERA aims at consolidating the relevant information to the ecological risk assessment domain. TERA integrates several disparate datasets and enables a unified (semantic) access. The formats of these data sources vary from tabular, to RDF files and SPARQL queries over public linked data. We have exploited external resources (*e.g.*, Wikidata [75]) and ontology alignment methods (*e.g.*, LogMap [35]) to discover equivalences between the data sources.

(ii) We have designed and implemented a model tailored to binary lethal chemical effect prediction. This model relies on TERA and builds upon existing knowledge graph embedding models. Moreover, it supplies the knowledge graph embedding models with additional information, which is used to tailor the embeddings to this specific task.

(iii) We have evaluated nine knowledge graph embedding (KGE) models, together with a naive baseline on the binary chemical effect prediction task. This evaluation includes four data sampling strategies which highlight the different settings of chemical effect prediction (*i.e.*, the test data contains unseen chemical-organism pairs where: *(a)* the chemical and the organism may be known (but not in previously seen pairs), *(b)* the chemical is unknown, *(c)* the organism is unknown, and *(d)* both the chemical and the organism are unknown).

These contributions are openly shared. A snapshot of the TERA knowledge graph is available on Zenodo [52] (https://doi.org/10.5281/zenodo.3559865) and the source scripts for creating TERA are available on GitHub (https://github.com/NIVA-Knowledge-Graph/TERA). Finally, the scripts to reproduce the conducted evaluation in this paper are also available on GitHub (https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020).

This paper extends our preliminary work presented in the In-Use Track of the 18th International Semantic Web Conference [50]. We have *(i)* extended TERA with new sources (Encyclopedia of Life (EOL), MeSH, and a larger part of ChEMBL) and provided detailed steps about its creation; *(ii)* created a more robust prediction model with nine (up from three) embedding algorithms supported and a task-specific embedding fine-tuning strategy *(iii)* conducted a more comprehensive evaluation with all combinations of KGE models and sampling strategies totalling 648 data points (324 for each prediction model).

The rest of the paper is organized as follows. Section 2 introduces essential concepts to the subsequent sections. Section 3 introduces the use case where the knowledge graph and prediction models are applied. Section 4 introduces related work. The creation of the knowledge graph is described in Section 5. Section 6 introduces the prediction models, while Section 7 presents the evaluation of these models. Section 8 elaborates on the contributions and discusses future directions of research. Finally, Appendix A gives an overview of the knowledge graph embedding models used in this work.

2. Preliminaries

In this section we introduce important background concepts that will be used throughout the paper.

2.1. Ecotoxicological terminology

Taxonomy in this work refers to a species classification hierarchy. Any node in a taxonomy is called a *taxon*. *Species* is a taxon which is also a leaf node in the taxonomy. An *Organism* denotes an individual living organism which is an instance of a species. *Chemicals or compounds* are unique isotopes of substances consisting of two or more atoms. *Effect*, used in this work as short form for chemical effect, refers to the response of an organism (or population) to a chemical at a specific concentration. *Endpoint*[1] denotes a measured effect on the test population at a certain time; *e.g.*, lethal concentration to 50% of test population (LC50) measured at 48 hours. Note that, an experiment can have

[1] Not to be confused with SPARQL endpoint.

several endpoints, *e.g.*, LC50 at 48 hours and LC100 at 96 hours (lethal concentration for all test organisms). See Table 1 for the most common endpoints.

2.2. Ontology-enhanced knowledge graphs

In this work we consider the most broadly accepted notion of knowledge graph within the Semantic Web: an ontology enhanced RDF-based knowledge graph (KG) [32]. This kind of knowledge graph enables the use of the available Semantic Web infrastructure, including SPARQL engines and OWL reasoners.[2] Thus, in our setting, KGs are composed by RDF triples in the form of $\langle s; p; o \rangle$, where s represents a subject (a class or an instance[3]), p represents a predicate (a property[4]) and o represents an object (a class, an instance or a literal). KG entities (*i.e.*, classes, properties and instances) are represented by an URI (Uniform Resource Identifier).

An (ontology-enhanced) KG can be split into a TBox (terminology) and an ABox (assertions). The TBox is composed by triples using RDF Schema (RDFS) constructors like class subsumptions and property domain and range; and OWL constructors like disjointness, equivalence and property inverses.[5] The ABox contains assertions among instances, including OWL equality and inequality, and semantic type definitions. Table 4 shows examples of TBox and ABox triples.

2.3. Ontology alignment

Ontology alignment is the process of finding mappings or correspondences between a source and a target ontology or knowledge graph [25, 65]. These mappings typically represent equivalences or broader/narrower relationships among the entities of the input ontologies. In the ontology matching community [1], mappings are exchanged using the RDF Alignment format [20]; but they can also be interpreted as standard OWL axioms (*e.g.*, [26, 34]). In this work we treat ontology alignments as OWL axioms (*e.g.*, Triple t_{13} in Table 4). An ontology matching system (*e.g.*, LogMap [33]) is a program that, given as input two ontologies or knowledge graphs, generates as output a set of mappings (*i.e.*, an alignment) M.

2.4. Embedding models

Knowledge graph embedding (KGE) [62, 77] plays a key role in link prediction problems where it is applied to knowledge graphs to resolve missing facts in largely connected knowledge graphs, such as DBpedia [44]. Biomedical link prediction is another area where embedding models have been applied successfully (*e.g.*, [2]).

The embeddings of the entities in a KG are commonly learned by *(i)* defining a scoring function over a triple, which is typically proportional to the probability of the existence of that triple in the KG, *i.e.*, $S : E \quad R \quad E \; ! \; \mathsf{R}, S \; \mu \; P(\langle s; p; o \rangle \; 2 \; KG)$; and *(ii)* minimizing a loss function (*i.e.*, deviation of the prediction of the scoring function with respect to the truth available in the KG). More specifically, KGE models *(i)* initialize the entities in a triple $\langle s; p; o \rangle$ into a vector representation $\mathbf{e}_s; \mathbf{e}_p; \mathbf{e}_o \; 2 \; \mathsf{R}^k$ or C^k, where k is the dimension of the vector; *(ii)* apply a scoring function to $(\mathbf{e}_s; \mathbf{e}_p; \mathbf{e}_o)$; and *(iii)* adapt the vector representations to improve the scoring and minimize the loss.

Several knowledge graph embedding models have been proposed. In this work, we used models of three major categories: decomposition models, geometric models, and convolutional models.[6] The decomposition models represent the triples of the KG into a one-hot 3-order tensor and apply matrix decomposition to learn entity vectors. Geometric models, also known as translational, try to learn embeddings by defining a scoring function where the predicate in the triple act as a geometric translation (*e.g.*, rotation) from subject to object. Convolutional models, unlike previous models, learn entity embedding with non-linear scoring functions via convolutional layers.

3. Ecotoxicological risk assessment and adverse biological effect prediction

The task of ecotoxicological risk assessment is to study the potential hazardous effects of chemicals on organisms from individuals to ecosystems. In this context, risk is the result of the intrinsic hazards of a substance on species, populations or ecosystems, com-

[2] RDF, RDFS, OWL and SPARQL are standards defined by the W3C: https://www.w3.org/standards/semanticweb/

[3] E is the set of all classes and instances.

[4] R is the set of all properties.

[5] Note that the Web Ontology Language (OWL) [19] also enables the creation of complex axioms that are translated/serialized into more than one triple: https://www.w3.org/TR/owl2-mapping-to-rdf/

[6] The interested reader please refer to [62] for a comprehensive survey.

bined with an estimate of the environmental exposure, *i.e.*, the product of exposure and effect (hazard).

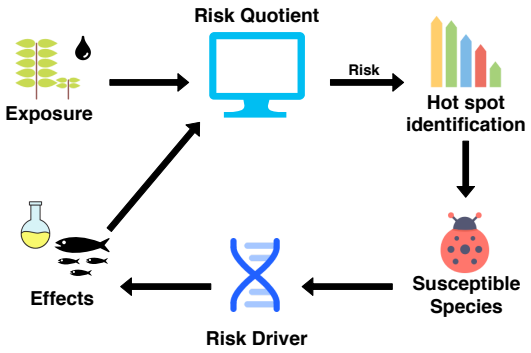

Fig. 1. Simplified ecological risk assessment pipeline.

Figure 1 shows a simplified risk assessment pipeline. *Exposure* data is gathered from analysis of environmental concentrations of one or more chemicals, while *effects* (*hazards*) are characterized for a number of species in the laboratory as a proxy for more ecologically relevant organisms. These two data sources are used to calculate the so-called risk quotient (RQ; ratio between exposure and effects). The RQ for one chemical or the mixture of many chemicals is used to identify chemicals with the highest RQs (risk drivers), identify relevant modes of action[7] (MoA) and characterize detailed toxicity mechanisms for one or more species (or taxa). Results from these predictions can generate a number of new hypotheses that can be investigated in the laboratory or studied in the environment. Note that, this risk assessment pipeline is a simplified version of the one in use at the Norwegian Institute for Water Research,[8] however, similar methodologies are used across regulatory risk assessment pipelines.

The chemical effect data is gathered during laboratory experiments, where a sub-population of a single species is exposed to an increasing concentration of a toxic chemical. The *endpoints* of the experiments are recorded at chemical concentrations and time after exposure. These *endpoints* are categorized into several categories, *e.g.*, lethality rate of test population (see Table 1).

Ecological risk assessment methods require a large amount of these experimental data to give an accurate depiction of the long term risk to an ecosystem. The data must cover the relevant chemicals and species present in the ecosystem, *e.g.*, an ecological risk assessment of agricultural runoff in Norway will mostly concern pesticides and waterflees, copepods, and frogs, among other species [42]. Just with a few relevant chemicals and species the search space becomes immense and performing laboratory experiments becomes unfeasible. Thus, it is essential to develop *in silico* methods to extrapolate new chemical-species effects from known combinations. We differentiate among two types complementary strategies: *(i)* highly specialized (restricted in chemical and species domains) models to predict chemical concentrations that will have an effect on a test species, and *(ii)* models that produce rankings of highly representative chemical-species pair hypothesis which can be used by a laboratory to perform targeted experiments. In this paper we focus on the latter strategy, using a method based on knowledge graph embeddings. Methods that fall into the first strategy are introduced in Section 4.1.

4. Related Work

This section will cover related work from ecotoxicology and knowledge graph based prediction.

4.1. Toxicity extrapolation

There are two main research areas in toxicology to extrapolate chemical effects, *i.e.*, Quantitative Structure-Activity Relationship (QSAR) and read-across. QSAR modelling try to find a relationship between the structure of a chemical and the chemical's biological activity (*cf.* reviews [24, 28]). This relationship is described using derived chemical features. Some features are simple, *e.g.*, octanol-water partition coefficient or logP, others concern the entire chemical, *e.g.*, chemical fingerprints. The basis of the QSAR relationship is usually modeled as polynomial equations. Parthasarathi and Dhawan [59] take this further by using logarithm of chemical concentration to achieve a polynomial relationship: $\log(1/C) = f(p) + g(s)$, $f \approx P_2$ and $g \approx P_1$ (P_n is a polynomial of nth degree), where C is the chemical concentration while p and s denote the derived chemical features hydrophobicity[9] and electronic effects in the molecule, respectively.

[7] The mode of action describes the molecular pathway by which a chemical causes physiological change in an organism.

[8] NIVA: https://www.niva.no/en

[9] Measure of the absence of attraction to water.

Endpoint	Frequency	Description
NR	0:21	Not reported
NOEL	0:17	No-observable-effect-level
LC50	0:16	Lethal concentration for 50% of test population
LOEL	0:14	Lowest-observable-effect-level
NOEC	0:05	No-observable-effect-concentration
EC50	0:05	Effective concentration for 50% of test population
LOEC	0:04	Lowest observable effect concentration
BCF	0:03	Bioconcentration factor
NR-LETH	0:02	Lethal to 100% of test population
LD50	0:02	Lethal dose for 50% of test population
Other	0:11	

Table 1
The most frequent endpoints in ECOTOX [73] chemical effect data.

The drawback of these models is the applicability domains. Usually, a QSAR model considers a small set of chemicals (10ths to 100ths) and one single species. This means that new features and relationships need to be developed for each species and each chemical group.

The read-across methods try to mitigate these drawbacks, mainly by considering extrapolation of the effect at the chemical and species levels. Similar to QSAR models, read-across of chemicals use the chemical features to create similarity measures between chemicals to justify the read-across of chemical effects. The read-across in the species domain is harder. Species do not tend to have easily derived features. Therefore, genetic similarity has emerged as a viable option. Sequence Alignment to Predict Across Species Susceptibility (SeqAPASS) developed by the United States Environmental Protection Agency (U.S. EPA.) is a tool that is used for toxicity extrapolation [22, 41]. The tool uses a large amount of data available for humans, mice, rats, and zebrafish to extrapolate to areas with lower coverage.

4.2. Embedding models

In this work, we use nine KGE models across three categories of models. Here, we will give a brief introduction to the models, while a more extended explanation of the models is found in Appendix A.

The three categories of models are decomposition, geometric, and convolutional [62]. The decomposition models are DistMult, ComplEx, and HolE. DistMult models the score of a triple as the vector multiplication of the representation of each subject, predicate and object [82]. ComplEx uses the same scoring function as DistMult, however, in a complex vector space, such that it can handle inverse relations [72]. HolE is based on holographic embeddings [55], however, it has been shown that HolE is equivalent to ComplEx [30].

The geometric models are TransE, RotatE, pRotatE, and HAKE. TransE is the base of a whole family of models and scores triples based on the translation from subject to object using the representation of the predicate [11]. RotatE is similar to TransE, however, the translation using the predicate is done by rotating it (via Euler's identity) [69]. Furthermore, pRotatE is a baseline for RotatE where the modulus in Euler's identity is ignored [69]. Finally, the hierarchical-aware model, HAKE, where entities at each level in the hierarchy is at equal distance from the origin and relations at a level is modeled as rotation [85].

The convolutional models take a deep learning approach to the task of KGE. We use ConvKB [54] and ConvE [21], which are similar with slightly different architectures. They have shown good performance given the relative small number of parameters.

Although quite a few KGE models have been proposed, the adopted ones are either classic models or can achieve state-of-the-art performance in some benchmarks. They are representative of mainstream techniques, and have been widely adopted in KGE research and applications [62]. Thus, the benefits and shortcomings of the KGE models analysed in this study provide good evidence of the general performance of this type of models in a complex prediction task, *i.e.*, adverse biological effect of chemicals on organisms.

4.3. Using KGE for prediction

Our focus to use KGE models is to predict if a chemical has a lethal effect on an organism. KGE models have been explored in the biomedical domain to solve similar predictions tasks (*e.g.*, finding relationships between diseases, drugs, genes, and treatments). Several works have shown improvements in results by using KGE models for prediction, *e.g.*, [2, 6, 46]. Chen et al. [16] used random walks over networks to perform drug-target predictions. The ChEMBL and DrugBank KGs have also been used to predict chemical mode of action (MoA) of anticancer drugs with high performance on benchmark datasets [81].

Opa2vec [67] and Blagec et al. [9] have developed embedding models to improve similarity-based prediction in the biomedical domain, while OpenBioLink [13] has created a framework for evaluating models in the biomedical domain.

EL Embeddings [40] and opa2vec [67] present new semantic embedding methods for KGs with expressive logic expressions (*i.e.*, OWL ontologies) to predict protein interaction. The former utilizes complex geometric structures to model the logic relationships between entities, while the later learns a language model from a corpus extracted from the ontology. OWL2Vec* [14] also learns a language model from an ontology and applies the computed embeddings into two prediction tasks: class subsumption and class membership. OWL2Vec* has also been used to predict the plausibility of ontology alignments [15].

To the best of our knowledge there is no work using link prediction or KGE models to support ecotoxicological effect prediction. This study will give novel insights and empirical results of these models in this new domain.

5. TERA knowledge graph

One major challenge in ecological risk assessment processes is the interoperability of data. In this section, we introduce the Toxicological Effect and Risk Assessment (TERA) knowledge graph that aims at providing an integrated view of the relevant data sources for risk assessment[10].

The initial inspiration for TERA was the aid of ecotoxicological effect prediction where access to disparate resources was required (see Section 5.3). However, by integrating these sources into a KG, we were also able to directly apply TERA into the prediction process by leveraging knowledge graph embedding models (see Section 5.4).

The data sources integrated into TERA vary from tabular and RDF files to SPARQL queries over public linked data. The sources currently integrated into TERA are: *(i)* biological: NCBI Taxonomy, Encyclopedia of Life, and Wikidata mappings ($500k$ species); *(ii)* chemical: PubChem, ChEMBL, MeSH, and Wikidata mappings ($110M$ compounds); and *(iii)* biological effects: ECOTOXicology Knowledgebase ($1M$ results, $12k$ compounds, $13k$ species), and system-generated mappings . These three distinct parts make up the sub-KGs of TERA, *i.e.*, *(i)* the Taxonomy sub-KG (KG_S), *(ii)* the Chemical sub-KG (KG_C), and *(iii)* the Effects sub-KG (KG_E). The different processes to transform and integrate these sources into TERA are shown in Figure 2.

A snapshot of TERA is available on Zenodo [52], where licenses permit.[11] PubChem and ChEMBL are not included in the snapshot due to size constraints; these can be downloaded from the National Institutes of Health[12] and European Bioinformatics Institute[13], respectively. The subgraph of TERA used for prediction is available alongside the chemical effect prediction models.[14]

TERA is an ontology-enhanced RDF-based knowledge graph. See example RDF triples from TERA in Table 4.[15]

[10] Resources to create and access TERA: https://github.com/NIVA-Knowledge-Graph/TERA

[11] EOL: Various Creative commons (CC), NCBI: Creative Commons CC0 1.0 Universal (CC0 1.0), ECOTOX: No restrictions, PubChem: Open Data Commons Open Database License, ChEMBL: CC Attribution, MeSH: Open, *Courtesy of the U.S. National Library of Medicine*, Wikidata: CC0 1.0.

[12] ftp://ftp.ncbi.nlm.nih.gov/pubchem/RDF/

[13] ftp://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBL-RDF/

[14] https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020

[15] Prefixes associated to the URI namespaces of entities in TERA: `et:` (ECOTOXicology knowledgebase), `ncbi:` (NCBI taxonomy), `eol:` (Encyclopedia of Life), `mesh:` (Medical Subject Heading), `compound:` (PubChem compound), `descr:` (PubChem descriptors), `vocab:` (PubChem vocabulary), `inchikey:` (InChIKey identifiers), `envo:` (Environment Ontology) `cheminf:` (Chemical information ontology), `chembl:` (ChEMBL), `chembl_m:` (ChEMBL molecule subset), `chembl_t:` (ChEMBL target subset), `wd:` (WikiData entities), `wdt:` (Wikidata properties), `qudt:` (Quantities, Units, Dimensions and Types Catalog), `snomedct:` (SNOMED CT ontology), and `bp:` (Biological PAthway eXchange ontology). `owl:`, `rdfs:`, `rdf:` and `xsd:` are prefixes referring to W3C standard vocabularies.

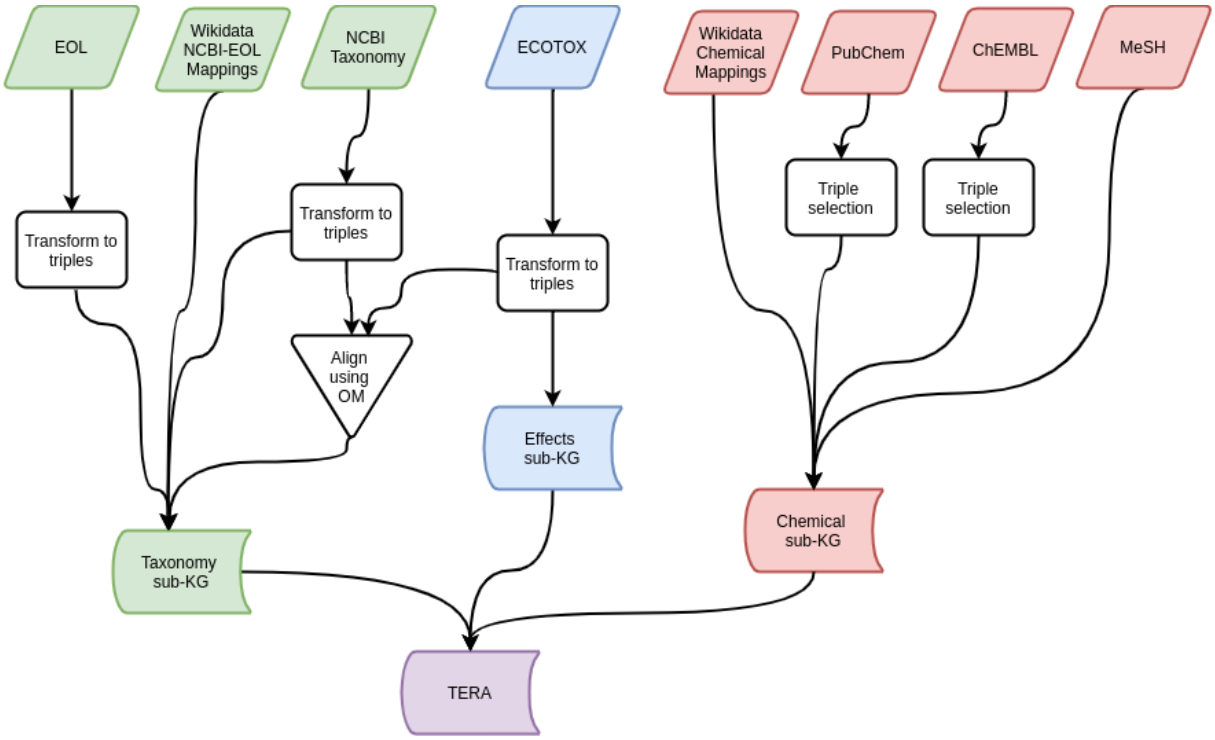

Fig. 2. Data sources and processes to create the TERA knowledge graph.

5.1. Dataset overview

TERA is constructed by gathering a number of sources about chemicals, species and chemical toxicity, with a diverse set of formats including tabular data, RDF dumps and SPARQL endpoints.

Biological effect data of chemicals. The largest publicly available repository of effect data is the ECOTOXicology knowledgebase (ECOTOX) developed by the US Environmental Protection Agency [73]. This data is gathered from published toxicological studies and limited internal experiments. The dataset consists of $1M$ experiments covering $12k$ chemicals and $13k$ species[16], implying a chemical–species pair converge of maximum 0.6%. The resulting endpoint from an experiment is categorised in one of a plethora of predefined endpoints (see Table 1 above).

Tables 2 and 3 contain an excerpt of the ECOTOX database. ECOTOX includes information about the chemicals and species used in the tests. This information, however, is limited and additional (external) resources are required to complement ECOTOX.

Chemicals. The ECOTOX database uses an identifier called CAS Registry Number assigned by the Chemical Abstracts Service to identify chemicals. The CAS numbers are proprietary, however, Wikidata [75] (indirectly) encodes mappings between CAS numbers and open identifiers like *InChIKey*, a 27-character hash of the International Chemical Identifier (InChI) which encodes chemical information uniquely [31].[17] Wikidata also provides mappings to well known databases like PubChem, ChEMBL and MeSH, which include relevant chemical information such as chemical structure, structural classification and functional classification.

Taxonomy. ECOTOX contains a taxonomy[18] (of species), however, this only considers the species represented in the ECOTOX effect data. Hence, to enable extrapolation of effects across a larger taxonomic domain, we include the NCBI Taxonomy [63]. This taxonomy data source consists of a number of database dump files, which contains a hierarchy for all sequenced species,

[16] Version dated Sep. 15, 2020.

[17] While InChI is unique, InChiKey is not, and collisions have greater than zero probability [78].

[18] In the context of the paper "taxonomy" typically refers to a classification of organisms.

test_id	reference_number	test_cas	species_number	organism_habitat
1147366	12448	134623 (diethyltoluamide)	1 (*Pimephales promelas*)	Water

Table 2
ECOTOX database tests example.

result_id	test_id	endpoint	effect	conc1_mean	conc1_unit
102570	1147366	*LC*50	MOR	110000	*mg/L*

Table 3
ECOTOX database results example.

which equates to around 10% of the currently known life on Earth and is one of the most comprehensive taxonomic resources. For each of the taxa (species and classes), the taxonomy defines a handful of labels, the most commonly used of which are the *scientific* and *common* names. However, labels such as *authority* can be used to see the citation where the species was first mentioned, while *synonym* is a alternate *scientific* name, that may be used in the literature.

Species traits. As an analog to chemical features, we use species traits to expand the coverage of the knowledge graph. Apart from taxonomic classifications, traits are the most important information to identify species and will be of great importance when predicting the effect on the species.

The traits we have included in the knowledge graph are the habitat, endemic regions, and presence (and classifications of these). This data is gathered from the Encyclopedia of Life (EOL) [57], which is available as a property graph. Moreover, EOL uses external definitions of certain concepts, and mappings to these sources are available as glossary files. In addition to traits, researchers may be interested in species that have different conservation statuses, *e.g.*, if the population is stable or declining, etc. This data can also be extracted from EOL.

5.2. Dataset preprocessing

In this section we present the different steps to extract, transform and integrate the source datasets into the main TERA components and sub-KGs. All data is transformed using custom mappings (scripts) from the sources to RDF triples. Table 4 shows an excerpt of the triples in TERA.

5.2.1. Effects sub-KG construction

The effect data in ECOTOX consist of two parts, *i.e.*, test definitions and results associated with the test definitions (see Tables 2 and 3, respectively). The important columns of a test are the chemical and the species used. Other columns include metadata, but these are optional and often empty. Each result is composed by an endpoint, an effect, and a concentration (with a unit) at which the endpoint and effect are recorded.

This tabular data in ECOTOX is transformed into triples that form the *effects sub-KG* in TERA (KG_E). Note that a test can have multiple results. A subset of the effect triples are listed in Table 4 (see Triples t_1–t_{12}). A graphical representation for an effect test and its result is also shown in Figure 3.

ECOTOX contains metadata about the species and chemicals used in the experiments. This metadata is also included in TERA to facilitate the alignment with other resources (see Section 5.2.2).

(i) The ECOTOX metadata file *species.txt* includes common and Latin names, along with a (species) ECOTOX group (see triples t_8-t_{10} in Table 4). This group is a categorization of the species based on ECOTOX use cases. Prefixes and abbreviations like *sp.*, *var.* are removed from the label names.

(ii) The full hierarchical lineage[19] is also available in the metadata file *species.txt*. Each column represents a taxonomic level, *e.g.*, *genus* or *family*. If a column is empty, we construct an intermediate classification; for example, *Daphnia magna* has no genus classification in the data, then its classification is set to Daphniidae genus (family name + genus, actually called *Daphnia*). We construct these classifications to ensure the number of lev-

[19] As defined by U.S. EPA. Note that species hierarchies are contested among researchers.

#	subject	predicate	object
		Effects sub-KG	
t_1	et:test/1147366	et:compound	et:chemical/134623
t_2	et:test/1147366	et:species	et:taxon/1
t_3	et:test/1147366	et:hasResult	et:result/102570
t_4	et:result/102570	et:endpoint	et:LC50
t_5	et:result/102570	et:effect	et:Mortality
t_6	et:taxon/1	rdf:type	et:taxon/Pimephales
t_7	et:taxon/Pimephales	rdfs:subClassOf	et:taxon/Cyprinidae
t_8	et:taxon/1	et:latinName	"Pimephales promelas"
t_9	et:taxon/1	et:commonName	"Fathead Minnow"
t_{10}	et:taxon/1	et:speciesGroup	et:group/Fish
t_{11}	et:taxon/1	et:rank	et:rank/species
t_{12}	et:chemical/134623	rdfs:label	"diethyltoluamide"
		Entity Mappings	
t_{13}	et:taxon/1	owl:sameAs	ncbi:taxon/90988
t_{14}	ncbi:taxon/90988	owl:sameAs	wd:Q2700010
t_{15}	wd:Q2700010	owl:sameAs	eol:211492
t_{16}	et:chemical/134623	owl:sameAs	wd:Q408389
t_{17}	wd:Q408389	owl:sameAs	chembl_m:CHEMBL1453317
t_{18}	wd:Q408389	owl:sameAs	compound:CID4284
t_{19}	wd:Q408389	owl:sameAs	mesh:D003671
t_{20}	wd:Q408389	owl:sameAs	inchikey:MMOXZBCLC...[1]
		Taxonomy sub-KG	
t_{21}	ncbi:taxon/90988	rdf:type	ncbi:taxon/51137[2]
t_{22}	ncbi:taxon/90988	rdf:type	ncbi:division/10
t_{23}	ncbi:taxon/90988	ncbi:scientific_name	"Pimephales promelas"
t_{24}	ncbi:taxon/90988	ncbi:rank	ncbi:species
t_{25}	ncbi:taxon/51137	rdfs:subClassOf	ncbi:taxon/7953[3]
t_{26}	ncbi:division/10	rdfs:label	"Vertebrates"
t_{27}	ncbi:division/10	owl:disjointWith	ncbi:division/1
t_{28}	ncbi:division/1	rdfs:label	"Invertebrates"
t_{29}	eol:211492	eol:habitat	envo:00000153[4]
		Chemical sub-KG	
t_{30}	mesh:D003671	mesh:broaderDescriptor	mesh:D001549[5]
t_{31}	mesh:D003671	mesh:pharmacologicalAction	mesh:D007302[6]
t_{32}	chembl_m:CHEMBL1453317	chembl:hasTarget	chembl_t:CHEMBL1907594[7]
t_{33}	chembl_t:CHEMBL1907594	chembl:relSubsetOf	chembl_t:CHEMBL3137273[8]
t_{34}	compound:CID89845769[9]	vocab:hasParentCompound	compound:CID4284
t_{35}	compound:CID131721069[10]	cheminf:CHEMINF_000478[11]	compound:CID4284
t_{36}	compound:CID131721069	rdf:type	bp:SmallMolecule
t_{37}	compound:CID7547[12]	vocab:is_active_ingredient_of	snomedct:411346009[13]
t_{38}	compound:CID131721069	cheminf:CHEMINF_000480[14]	compound:CID10751691[15]

Table 4

Example triples from the TERA knowledge graph. For space reasons, we have added the full id or label for some of the entities using footnote marks where [1]inchikey:MMOXZBCLCQITDF-UHFFFAOYSA-N, [2]Pimephales, [3]Cyprinidae, [4]Headwater, [5]Benzamides, [6]Insect Repellents, [7]CHRNA3, [8]CHRNB4, [9]DETA-20, [10]DETA Epichlorohydrin, [11]Has component, [12]Triclocarban, [13]Trichlorocarbanilide-containing product, [14]Similar to, [15]3-Chloromethyl-N,N-diethylbenzamide.

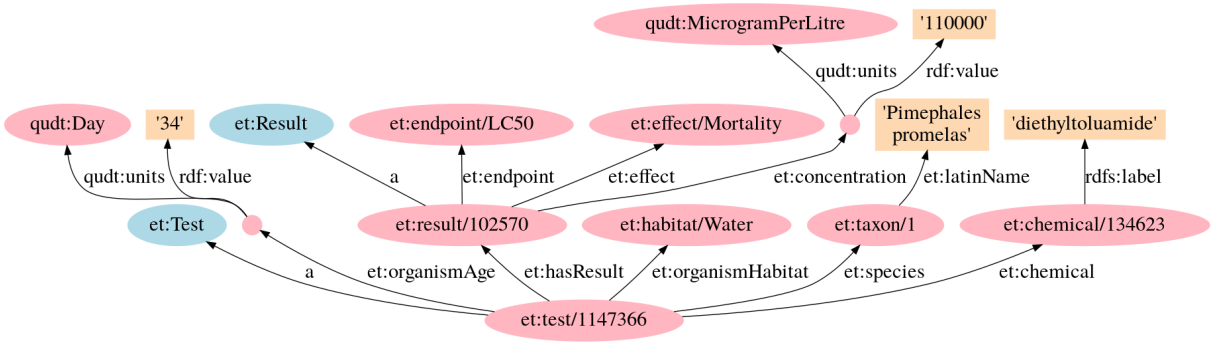

Fig. 3. Example of a ECOTOX related triples.

els in the taxonomy is consistent (see triples t_6 and t_7 in Table 4). Note that when adding triples such as t_{11} in Table 4, we also add a taxonomic rank to facilitate the querying for a specific taxonomic level.

(iii) The ECOTOX source file *chemicals.txt* includes chemical metadata and it is handled similarly to *species.txt*. The file includes chemical name (see t_{12} in Table 4) and a (chemical) ECOTOX group.

For the units in the effect data, *e.g.*, chemical concentrations (mg/L, mol/L, mg/kg, etc.), we reuse the QUDT 1.1[20] ontologies. When an unit such as mg/L is not defined, we define it according to Listing 1.

5.2.2. *Alignment with state-of-the-art tools*

ECOTOX database provides proprietary chemical identifiers (*i.e.*, CAS numbers) and internal ECOTOX ids for species. In order to extrapolate effects across a larger set of chemicals and species than those available in ECOTOX, TERA integrates taxonomy and trait data from NCBI and EOL, and chemical data from PubChem, ChEMBL and MeSH

Alignment between ECOTOX and the NCBI Taxonomy. There does not exist a complete and public alignment between the 23,439 ECOTOX species and the 1,830,312 the NCBI Taxonomy species.[21] We have used three methods, two state-of-art ontology alignments systems and a baseline, to align ECOTOX and the NCBI Taxonomy: *(i)* LogMap [35], *(ii)* AgreementMakerLight (AML) [27], and *(iii)* a string matching algorithm based on Levenshtein distance [45]. LogMap and AML were chosen since they have performed well across many datasets in the Ontology Alignment Evaluation Initiative (*e.g.*, [1, 3, 4]). Most mappings in our setting are expected to be lexical, therefore, we also selected a purely lexical matcher to evaluate if more sophisticated systems like LogMap and AML bring an additional value.

Due to the large size of the NCBI Taxonomy, we needed to split NCBI into manageable chunks to enable the use of ontology alignment systems. Fortunately, this can be easily done by considering the species division, *e.g.*, mammal or invertebrate. This divides the NCBI Taxonomy into 11 distinct parts, which can be aligned to the taxonomy in ECOTOX.

Note that it is expected an entity from ECOTOX to match to a single entity in the NCBI Taxonomy, and vice-versa. Hence, 1-to-N and N-to-1 alignments were filtered according to the system computed confidence. A partial mapping curated by experts can be obtained through the ECOTOX Web.[22] We have gathered a total of 2,321 mappings for validation purposes. Table 5 shows the alignment results over the ground truth sam-

Method	1-to-1 mappings		
	# M	R	P
LogMap	20,585	0.81	0.87
AML	14,148	0.77	0.94
String similarity (> 0.8)	20,423	0.76	0.87
Consensus (LogMap \ AML)	12,740	0.76	0.98
LogMap ⌈ AML	**21,145**	**0.83**	**0.86**

Table 5

Alignment results for ECOTOX-NCBI. #M: number of mappings (at instance level), R: Recall, P : estimated precision.

[20] QUDT 1.1: http://linkedmodel.org/catalog/qudt/1.1/

[21] There are a total of 27,133 and 2,246,074 taxa in ECOTOX and NCBI, respectively. However, we focus on species, *i.e.*, instances.

[22] ECOTOX search interface: https://cfpub.epa.gov/ecotox/search.cfm

```
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix qudt: <http://qudt.org/schema/qudt#> .
@prefix et:   <https://cfpub.epa.gov/ecotox> .
et:MilligramPerLiter
    rdf:type qudt:MassPerVolumeUnit, qudt:SIDerivedUnit ;
    rdfs:label "Milligram per Liter"^^xsd:string ;
    qudt:abbreviation "mg/L"^^xsd:string ;
    qudt:conversionMultiplier 0.000001 ;
    qudt:conversionOffset 0.0 ;
    qudt:symbol "mg/dm^3"^^xsd:string .
```

Listing 1: Unit definition of mg/L using QUDT.

ples for the 1-to-1 (filtered) system mappings. We report number of mappings (#M), Recall (R) and estimated precision (P) with respect to the known entities in the incomplete ground truth, assuming only 1-to-1 mappings are valid. P is calculated as

$$P = jM \setminus M_{ref}j=jM j; \quad (1)$$

$$M = f(e_e;e_n) \; 2 \; M \; j \; e_e \; 2 \; E_e^{ref} \_ \; e_n \; 2 \; E_n^{ref} g; \quad (2)$$

where M_{ref} is the (incomplete) reference mapping set and M is the set of generated mappings, E_e^{ref} and E_n^{ref} are the sets of entities in the reference mappings for ECOTOX and the NCBI Taxonomy, respectively. Thus, M is defined as a subset of mappings from M involving entities in the reference mapping set M_{ref}. Recall is defined in the standard way as

$$R = jM \setminus M_{ref}j=jM_{ref}j: \quad (3)$$

Note that, the recall will be the same for M and M .

We have selected the union of the 1-to-1 equivalence[23] mappings computed by AML and LogMap to be integrated within TERA, as they represent the mapping set with the best recall with a reasonable estimated precision. This choice was made by considering the large uncertainty of downstream applications (effect prediction and risk assessment), where we prefer large coverage of the domain. See Triple t_{13} in Table 4 for an example of a system computed mapping between ECOTOX and the NCBI Taxonomy.

We use Wikidata as source of alignments between the NCBI Taxonomy and EOL, and among the used chemical datasets. Alignments are extracted via Wikidata's query interface (*i.e.*, SPARQL endpoint).[24] The data in Wikidata concerning species and chemicals are in large parts manually curated [76] and will have a low error rate, comparatively to using the automated ontology alignment systems.

Alignment between the NCBI Taxonomy and EOL. In order to include in TERA trait data from EOL, we need to establish an alignment between EOL and the NCBI Taxonomy. We have constructed equivalence triples between the NCBI Taxonomy and EOL identifiers using Wikidata. The species identifiers are available as literals in Wikidata. Therefore, we concatenate them with the appropriate namespace. Listing 2 represents the SPARQL CONSTRUCT query used against the Wikidata endpoint. Here, we query Wikidata for instances of taxa, thereafter adding optional triple patterns for NCBI Taxonomy and EOL identifiers which are added as `owl:sameAs` triples to TERA.

Examples of resulting mapping triples are shown in t_{14}-t_{15} in Table 4. The proportion of species in Wikidata where this mapping exists is 49%.

Alignment between chemical entities. The mapping between ECOTOX chemical identifiers (CAS Registry Numbers) to Wikidata entities enables the alignment to a vast set of chemical datasets, *e.g.*, PubChem, ChEBI, KEGG, ChemSpider, MeSH, UMLS, to name a few. The construction of equivalence triples between CAS, ChEMBL, MeSH, PubChem and Wikidata identifiers is shown in Listing 3. As for the case of species identifiers, the literal representing a chemical identifier is concatenated with the corresponding namespace. For the CAS Registry Numbers we also remove the hyphens to match ECOTOX notation. Examples of resulting mapping triples are shown in t_{16}-t_{20} in Table 4.

These mappings are not complete, but for some the coverage is large. Out of the chemicals used in ECOTOX, 73% have an equivalence in Wikidata (through the CAS registry numbers). Moreover, Wikidata chem-

[23] There is no need for more complex mappings in this use case.
[24] Wikidata endpoint: https://query.wikidata.org/sparql

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
CONSTRUCT {?taxon owl:sameAs ?ncbi , ?eol .}
WHERE {
    ?taxon wdt:P31 wd:Q16521 .
    OPTIONAL {
        ?taxon wdt:P685 ?ncbi_id .
        BIND(
            IRI(CONCAT(
               "https://www.ncbi.nlm.nih.gov/taxonomy/taxon/",
                ?ncbi_id))
            AS ?ncbi)
    }
    OPTIONAL {
        ?taxon wdt:P830 ?eol_id .
        BIND(IRI(CONCAT("https://eol.org/pages/",?eol_id)) AS ?eol)
    }
}
```

Listing 2: Construct taxon mapping between Wikidata and, NCBI and EOL. `wd:Q16521` is the class of all taxa, while `wdt:P31`, `wdt:P685` and `wdt:P830` are the relations *instance of*, *NCBI Taxonomy ID* and *Encyclopedia of Life ID*, respectively.

icals has 4% ChEMBL identifiers, 0:5% MeSH identifiers, 55% PubChem identifiers, and 95% InChiKey identifiers.

5.2.3. Taxonomy sub-KG construction

The Taxonomy sub-KG (KG_S) integrates data from the NCBI Taxonomy and the EOL trait data. The integration of the NCBI Taxonomy into the TERA knowledge graph is split into several sub-tasks.

(i) We load the hierarchical structure included in the NCBI Taxonomy file *nodes.dmp*. The columns of interest are the taxon identifiers of the child and parent taxon, along with the rank of the child taxon and the division where the taxon belongs. We use this to create triples like t_{21}-t_{22} and t_{24}-t_{25} in Table 4.

(ii) To aid alignment between the NCBI Taxonomy and the ECOTOX identifiers, we add the synonyms found in *names.dmp*. Here, the taxon identifier, its name and name type are used to create triples like t_{23} in Table 4. Note that a taxon in the NCBI Taxonomy can have several synonyms while a taxon in ECOTOX usually has two, *i.e.*, common name and scientific name.

(iii) Finally, we add the labels of the divisions found in *divisions.dmp* (see triples t_{26} and t_{28}). In addition, we add disjointness axioms among unrelated divisions, *e.g.*, triple t_{27} in Table 4.

We use the TraitBank from EOL [58] to add species traits to TERA. The TraitBank is modeled as a property graph and can be accessed as a *neo4j* database or via a set of tabular files. To integrate the TraitBank into TERA we validate the identifiers used in EOL and convert to URIs. If an identifier is not a valid URI, we replace invalid symbols. A trait example is shown as triple t_{29} in Table 4. The EOL TraitBank also includes assertions of `rdfs:subClassOf` for a large portion of traits. These assertions can be downloaded separately and are added to TERA in a similar way as mentioned above.

5.2.4. Chemical sub-KG construction

The Chemical sub-KG (KG_C) is created from PubChem [38], ChEMBL [29], and MeSH [56]. These datasets are available for download as RDF triples. In addition, ChEMBL and MeSH can be accessed through the EBI and MeSH SPARQL endpoints, respectively.

The chemical subset of PubChem is used since information about chemicals is standardized in PubChem, while information about substances is not. In this subset we use *(i)* component information, *i.e.*, what are the building blocks of the chemical or parts of a mixture, *(ii)* type assertions, which either link to ChEBI or describe the type of molecule, *e.g.*, small or large, *(iii)* role assertions, which describe if

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
CONSTRUCT {?chemical owl:sameAs
            ?cas, ?chembl, ?mesh, ?pubchem , ?inchikey .}
WHERE {
?chemical wdt:P31 wd:Q11173 .
    OPTIONAL {
    ?chemical wdt:P231 ?cas_id .
    BIND(IRI(
        CONCAT("https://cfpub.epa.gov/ecotox/chemical/",
                REPLACE(?cas_id,'-',''))) AS ?cas)
    }
    OPTIONAL {
        ?chemical wdt:P592 ?chembl_id .
        BIND(IRI(
        CONCAT("http://rdf.ebi.ac.uk/resource/chembl/molecule/",
                ?chembl_id)) AS ?chembl)
    }
    OPTIONAL {
        ?chemical wdt:P486 ?mesh_id .
        BIND(IRI(
        CONCAT("http://id.nlm.nih.gov/mesh/",?mesh_id)) AS ?mesh)
    }
    OPTIONAL {
        ?chemical wdt:P662 ?pubchem_id .
        BIND(IRI(
        CONCAT("http://rdf.ncbi.nlm.nih.gov/pubchem/compound/CID",
                ?pubchem_id)) AS ?pubchem)
    }
    OPTIONAL {
        ?chemical wdt:P235 ?inchikey_id .
        BIND(IRI(
        CONCAT("https://rdf.ncbi.nlm.nih.gov/pubchem/inchikey/",
                ?inchikey_id)) AS ?inchikey)
    }
}
```

Listing 3: Construct chemical mapping between Wikidata and ECOTOX, ChEMBL, MeSH and PubChem. `wdt:P31` is the predicate for *instance of* and `wd:Q11173` is the class of all chemical chemicals. `wdt:P231`, `wdt:P592`, `wdt:P486`, `wdt:P662` and `wdt:P235` are the relations for *CAS Registry Number*, *ChEMBL ID*, *MeSH ID*, *PubChem CID* and `InChIKey`, respectively.

a chemical has any role, *e.g.*, `FDAApprovedDrug`, *(iv)* drug products linking to the clinical data in SNOMEDCT [8]. Examples of these can be seen in triples t_{35}, t_{36} and t_{37} in Table 4.

Parent chemical data in PubChem is limited to permutations *e.g.*, bonds, polarity, and part of mixtures axioms (triple t_{34} in Table 4). Therefore, we use the hierarchical data about chemicals from MeSH. In addition to this data, we create similarity triples between chemicals. This is impractical to download, but can be calculated on demand. We add similarity triples to TERA where the Tanimoto (Jaccard) distance between the chemical fingerprints (gathered using PubChemPy [70]) is > 0:9,[25] see triple t_{38} in Table 4.

[25] Default value used in PubChem [37].

ChEMBL contains facts about bioactivity of chemicals. This contributes in assessing the danger of a chemical. In TERA, we use the Mode of Action (MoA) and target (receptor targeted by MoA; triple t_{32} in Table 4). These targets are organized in a hierarchy using `chembl:relSubsetOf` relations (see triple t_{33}). The receptors will link to which organism it belongs to, however, we leave the inclusion of this information for future work.

We use the entire MeSH dataset in TERA. MeSH is organised as several hierarchies. The most prominent classifications are based on chemical groups and the intended use of the chemicals. Triples t_{30} and t_{31} in Table 4 show examples of chemical group and functional classifications.

5.3. TERA for data access

TERA covers knowledge and data relevant to the ecotoxicological domain and enables an integrated semantic access across data sets. In addition, the adoption of an RDF-based knowledge graph enables the use of an extensive range of Semantic Web infrastructure (*e.g.*, reasoning engines, ontology alignment systems, SPARQL query engines).

The data integration efforts and the construction of TERA go in line with the vision in the computational risk assessment communities (*e.g.*, Norwegian Institute for Water Research's Computational Toxicology Program (NCTP)), where increasing the availability and accessibility of knowledge enables optimal decision making.

The knowledge in TERA can be accessed via predefined queries[26] (*e.g.*, classification, sibling, and name queries, and fuzzy queries over the species names) and arbitrary SPARQL queries. The (final) output is flexible to the task, and can be given either as a graph or in tabular format. Listing 4 shows an example query to extract the chemicals and concentrations, at which, the species in the *Oslofjord* experience lethal effects.

5.4. TERA for effect prediction

TERA is used as background knowledge in combination with machine learning models for chemical effect prediction. TERA's sub-KGs play different roles in effect prediction. The rich semantics of the species and chemical entities in the Taxonomy sub-KG (KG_S) and the Chemical sub-KG (KG_C), respectively, are embedded into low-dimensional vectors; while the Effects sub-KG (KG_E) provides the training samples for the prediction model. Each sample is composed of a chemical, a species, a chemical concentration, and the outcome or endpoint of the experiment. More details are given in Section 6, where the effect prediction model is built upon existing knowledge graph embedding models.

[26] Predefined queries are typically abstractions of SPARQL queries.

Dataset	RD	ED	RE	EE	AD
TERA KG_C	2:3 10^5	5:5	3:0	24	4:6 10^7
TERA KG_S	6:6 10^4	5:1	2:7	23	3:7 10^7
TERA $KG_C^{\emptyset}$	6:9 10^3	8:6	2:3	17	7:7 10^5
TERA $KG_S^{\emptyset}$	3:8 10^2	15	2:3	14	8:9 10^4
YAGO3-10	2:9 10^4	18	2:0	20	7:1 10^5
FB15k-237	1:3 10^3	43	4:5	16	1:3 10^3
WN18	8:4 10^3	7:4	2:1	16	9:0 10^5
WN18RR	8:5 10^3	4:5	1:5	19	5:5 10^5

Table 6

Densities and entropies of benchmark datasets. TERA KG_C and KG_S are the chemical and species parts of TERA, while $KG_C^{\emptyset}$ and $KG_S^{\emptyset}$ denote the parts of TERA used in prediction in Section 7.

Table 6 shows the sparsity-related measures of common benchmark datasets[27] and TERA's KG_C and KG_S (triples involving literals are removed). We follow Pujara et al. [61] and calculate the relational density, $RD = jTj{=}jRj$, and entity density, $ED = 2jTj{=}jEj$, where T, R, and E are the sets of triples, relations, and entities in the knowledge graph, respectively. The entity entropy (EE) and the relation entropy (RE) indicate whether there are biases (the lower EE or RE, the larger bias) in the triples in the KG [61], and are calculated as

$$P(r) = \frac{jt{:}p = rj}{jTj};$$

$$P(e) = \frac{jt{:}s = ej + jt{:}o = ej}{jTj};$$

$$RE = \sum_{r2R} P(r)log(P(r));$$

$$EE = \sum_{e2E} P(e)log(P(e));$$

[27] YAGO3-10 [68], FB15k-237 [10], WN18 [47] and WN18RR [21].

```
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX eol: <http://eol.org/schema/terms/> .
PREFIX et:  <https://cfpub.epa.gov/ecotox/> .
PREFIX et_endpoint:  <https://cfpub.epa.gov/ecotox/endpoint/> .
PREFIX et_effect:  <https://cfpub.epa.gov/ecotox/effect/> .
PREFIX qudt:  <http://qudt.org/schema/qudt#> .
SELECT ?s ?c ?conc ?concunit
WHERE {
    ?s  eol:endemicTo [ rdfs:label "Oslofjorden"@no ] .
    _:b a et:Test ;
        et:species ?s .
        et:chemical ?c .
        et:hasResult [
            et:endpoint et_endpoint:LC50 ;
            et:effect et_effect:Mortality ;
            et:concentration [
                                rdf:value ?conc ;
                                qudt:units ?concunit
                              ] .
        ]
}
```

Listing 4: Query to select all species, chemicals, and concentrations and unit, where the species is endemic to the *Oslofjord*.

where $jt{:}p = rj$ is the number of triples with r as predicate, and $jt{:}s = ej + jt{:}o = ej$ is the number triples with e as subject or object.

In addition, we calculate the absolute density of the graph, which is $AD = jTj{=}(jEj(jEj \quad 1))$. This is the ratio of edges to the maximum number of edges possible in a simple directed graph [18].

High RD and low RE typically lead to a worse performance, while high ED and low EE often lead to better link prediction performance (*e.g.*, [21]). In Table 6 we can see that the density and entropy values are in between those for YAGO3-10 and FB15k-237, which typically lead to worse and better predictive performance, respectively [21]. This shows that TERA is a suitable background knowledge to extrapolate effect data and, at the same time, an interesting dataset to benchmark state-of-the-art knowledge graph embedding models. Note that using the full TERA (*i.e.*, KG_C and KG_S), according to RD, will be more challenging than using the reduced TERA fragments (*i.e.*, KG_C^0 and KG_S^0) for prediction.[28]

[28] Full details of the construction of KG_C^0 and KG_S^0 are given in Section 7.1.1.

6. Adverse biological effect prediction

The aim of chemical effect prediction is to extrapolate exiting data to new and unknown combinations of chemicals and species. In this section we present three classification models used to predict the adverse biological effect of chemicals on species: *(i)* a multilayer perceptron (MLP) model (our baseline), *(ii)* the baseline model fed with pre-trained KG embeddings, *(iii)* a model that simultaneously trains the baseline model and the KGE models (*i.e.*, it fine-tunes the KG embeddings). A MLP was chosen as baseline as it is a basic model where additional components and penalties can be easily added and assessed as we do in our third model (see Section 6.3).

The models have three inputs, namely a chemical c, a species s, and a chemical concentration k (denoted $x_{c;s;k}$). The output is a binary value that represents whether the chemical at the given concentration has a lethal effect on the species:

$$y_{c;s;k} = \begin{cases} 1 & c \text{ is lethal to } s \text{ at } k{:} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Note that the effect can have a more fine-grained categorization (endpoints LCx, LDx, ECx[29], and NR-LETH in Table 1). Without losing the generality in introducing and evaluating our effect prediction methods, we simplify the effect into two cases: "lethal" and "non-lethal".

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as $\mathbf{e}_e$ and $\mathbf{e}_p$, respectively. These vectors are either in $\mathbb{R}^k$ or $\mathbb{C}^k$, where k is the embedding dimension.

6.1. Baseline model

Our baseline prediction model is a multilayer perceptron (MLP) with multiple hidden layers. n_c hidden layers are appended to the embedding $\mathbf{e}_c$ of the chemical c, n_s hidden layers are appended to the embedding $\mathbf{e}_s$ of species s, and n_k hidden layers appended to the real valued chemical concentration k. Thereafter, n hidden layers are further appended to the output of the previous hidden layers concatenated. Specifically, the model can be expressed by the following equations (with $x_{c;s;k}$ as input):

$$\mathbf{y}_c^0 = \mathbf{e}_c;\ \mathbf{y}_s^0 = \mathbf{e}_s;\ y_k^0 = k \tag{5}$$

$$\mathbf{y}_c^t = ReLu(\mathbf{y}_c^{t-1}\mathbf{W}_c^t + \mathbf{b}_c^t);\ t \in \{0;\dots;n_c\} \tag{6}$$

$$\mathbf{y}_s^t = ReLu(\mathbf{y}_s^{t-1}\mathbf{W}_s^t + \mathbf{b}_s^t);\ t \in \{0;\dots;n_s\} \tag{7}$$

$$\mathbf{y}_k^t = ReLu(\mathbf{y}_k^{t-1}\mathbf{W}_k^t + \mathbf{b}_k^t);\ t \in \{0;\dots;n_k\} \tag{8}$$

$$\mathbf{y}^0 = [\mathbf{y}_c^{n_c};\mathbf{y}_s^{n_s};\mathbf{y}_k^{nk}] \tag{9}$$

$$\mathbf{y}^t = ReLu(\mathbf{y}^{t-1}\mathbf{W}^t + \mathbf{b}^t);\ t \in \{1;\dots;n\} \tag{10}$$

$$\hat{y} = \sigma(\mathbf{y}^n\mathbf{W}^n + \mathbf{b}^n). \tag{11}$$

$\mathbf{e}_c;\mathbf{e}_s \in \mathbb{R}^k$ in (5) denote the embeddings of c and s respectively, and are calculated as

$$\mathbf{e}_c = d_c\mathbf{W}_c;\ \mathbf{e}_s = d_s\mathbf{W}_s; \tag{12}$$

where d_c and d_c denote the one-hot encoding vectors of the chemical entity c (w.r.t. all the chemical entities) and the species entity s (w.r.t. all the species entities), respectively[30], $\mathbf{W}_c \in \mathbb{R}^{jCj \times k}$ and $\mathbf{W}_s \in \mathbb{R}^{jSj \times k}$ are embedding transformation matrices to learn. (6), (7) and (10) represent the hidden layers, where *ReLu* denotes the rectifier function (*i.e.*, $ReLu(x) = \max(0;x)$), $\mathbf{W}_c^t$, $\mathbf{W}_s^t$ and $\mathbf{W}^t$ denote the weights, $\mathbf{b}_c^t$, $\mathbf{b}_s^t$ and $\mathbf{b}^t$ denote the biases. $[\,;\,]$ in (9) denotes vector concatenation. σ in (11) denotes the sigmoid function (*i.e.*, $\sigma(x) = 1/(1+\exp(-x))$). Note that a dropout and a normalization layer is stacked after each hidden layer for regularization.

We differentiate between two settings of the baseline model (see Figure 4):

(i) Simple setting. Figure 4a shows the model without embedding transformation layers, *i.e.*, $n_s = n_c = n_k = 0$, and $n = 1$.

(ii) Complex setting. The complex model shown in Figure 4b introduces transformation layers on the embeddings and chemical concentration input. These transformations will aim at extracting the important information in the inputs and disregard the redundant information based on the output.

In the experiments we refer to the baseline models as *Simple one-hot* and *Complex one-hot*, depending on the selected MLP setting.

6.2. Baseline model with pre-trained KG embeddings

This models relies on pre-trained embeddings of chemicals and species computed using state-of-the-art KGE models (see Section 4.2 and Appendix A for an overview). A (different) KGE model is applied to the chemicals KG_C and the species KG_S.

These pre-trained KG embeddings are then given as input instead of the one-hot encoding vectors in the baseline model. We replace the trainable matrices $\mathbf{W}_c$ and $\mathbf{W}_s$ in Equation (12) by the matrices composed of embeddings by the respective KGE models. Namely $\mathbf{W}_c$ is set to $[\mathbf{e}_{c;1};\mathbf{e}_{c;2};\dots;\mathbf{e}_{c;jCj}]$, $\mathbf{W}_s$ is set to $[\mathbf{e}_{s;1};\mathbf{e}_{s;2};\dots;\mathbf{e}_{s;jSj}]$, where $[\,;\,]$ denotes stacking vectors, $\mathbf{e}_{c;i}$ denotes the embedding of i^{th} chemical in the chemicals KG_C, $\mathbf{e}_{s;i}$ denotes the embedding of i^{th} species in the species KG_S.

In the experiments we refer to these models as *Simple PT KGE$_C$-KGE$_S$* and *Complex PT KGE$_C$-KGE$_S$*, depending on the selected MLP setting, where PT stands for pre-trained, and KGE$_C$ and KGE$_S$ are the KGE models used for the chemicals KG and the

[29] If effect is mortality. See Table 3.

[30] $d_c \in \mathbb{R}^{jEj}$ (E is the set of entities), where $d_c^i = 1$ if $c = E_i$, else 0. d_s is defined similarly.

(a) Simple setting. Without transformation layers: $n_c = 0; n_s = 0; n_k = 0$ and $n = 1$.

(b) Complex setting. Model with branches/transformation layers. In contrast to the simple setting, here $n_c > 1; n_s > 1; n_k > 1$ and $n > 1$.

Fig. 4. Baseline model. Inputs c, s, k as in Equation (5); Output $\hat{y}$ as in Equation (11).

species KG, respectively (e.g., Complex PT DistMultHAKE). For simplicity, we also refer to these models as PT-based models.

6.3. Fine-tuning optimization model

This model improves upon the pre-trained KG embeddings with fine-tuning based on the effect prediction data. This is done by simultaneously training the (selected) KGE models and the MLP-based baseline model. Such that the W_C and W_S, and the MLP weights (W_x and b_x in Equations (6), (7), (10) and (11)) are optimized simultaneously. Note that we initialize the KGE models with the previously pre-trained embeddings.

The model architecture is shown in Figure 5 and the overall loss to minimize is

$$L = a_C L_{KGE_C} + a_S L_{KGE_S} + a_{MLP} L_{MLP}; \qquad (13)$$

where L_{KGE_C} and L_{KGE_S} respectively denote the loss of the chemical KG, c and the species KG, s when a specific KGE model is used,[31] a_C and a_S denote their weights respectively. L_{MLP} and a_{MLP} denote the loss of the MLP and its weight. Specially, we use binary cross-entropy (BCE) as the loss for the classification. L_{MLP} is calculated as

$$L_{MLP} = \frac{1}{N} \sum_i^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \qquad (14)$$

where N denotes the size of training samples, y_i and $\hat{y}_i$ denote the sample label and the MLP output, respectively (as in Equation (4)). With the overall loss, gradient-based learning algorithms such as Adam optimizer [39] can be adopted to jointly training the embeddings of both KGEs and the MLP.

Figure 5 shows the full simultaneous fine-tuning model and the optimization process. The initial state of the entity lookups is the pre-trained embeddings. The full training procedure is described in the following steps:

1. Select N triples from KG_C and KG_S, where N is the length of the effects training set.[32]
2. Generate negative knowledge graph triples (see Appendix A.5 for details) from the extracted subsets of triples from KG_C and KG_S, these negative KGs triples are referred to as $\overline{KG}_C$ and $\overline{KG}_S$.
3. Feed-forward the input through the model and calculate loss for each model component and combine according the loss weights.
4. Optimize the KG entity and relation embeddings, and the MLP layers.

These steps are repeated until the loss (L_{MLP}) over the validation set stops improving.

[31] Appendix A.5 introduces the used loss-functions in this work. The selection of the loss function for a KGE model will be via a hyper-parameter.

[32] Section 7.1 describes how the known effect data extracted from ECOTOX is split into training, validation and test sets.

Fig. 5. Fine-tuning optimization model. In addition to variables described in Figures 4a and 4b, $t_C = (s_C; p_C; o_C) \in KG_C \cup \overline{KG}_C$, $t_S = (s_S; p_S; o_S) \in KG_S \cup \overline{KG}_S$. Entity lookups transform an entity into a vector (see Equation (12)). S_{KGE_C} and S_{KGE_S} are the triple scoring functions implemented by the used KGE models (see Appendix A), $x_{t;S;k}$ is the prediction input and $y_{t;S;k}$ is described in Equation (4). Triple labels l_{t_C} and l_{t_S} will depend on which loss functions L_{KGE_C} and L_{KGE_S} (from Appendix A.5) are used. BCE is the binary cross-entropy loss function (from Equation (14)). The summation of the losses is described in Equation (13), that is the loss used by the optimizer to apply changes to model weights.

In the experiments we refer to these models as Simple FT KGE$_C$-KGE$_S$ and Complex FT KGE$_C$-KGE$_S$, depending on the selected MLP setting, where FT stands for fine-tuning, and KGE$_C$ and KGE$_S$ are the KGE models used for the chemicals KG and the species KG, respectively (e.g., Simple FT HAKE-HAKE). For simplicity, we also refer to these models as FT-based models.

7. Results

7.1. Experimental setup

All models are implemented using Keras [17] and the model codes are available in our GitHub repository, alongside all data preparation and analysis scripts.[33]

7.1.1. Preparation of TERA for prediction

As shown earlier, TERA[34] consists of three sub-KGs. These are the basis for the chemical effect prediction. We process the sub-KGs further to limit their size by removing irrelevant triples for prediction. This is necessary to scale up the training of the KGE models. The reduction of TERA's sub-KGs is performed according to the following steps:

(i) Effect data. For prediction purposes, the effect data in KG_E is limited to four features, namely, chemical, species, chemical concentration, and effect. The chemical concentrations ($c_{c,s}$ converted to mg/L) are log-normalized to remove the large discrepancy in scales. As mentioned, we separate the effects into two categories for simplicity, lethal and non-lethal effects. This reduces the possibility of ambiguity among the effects that does not cause death in the test species. We label lethal effects as 1 and non-lethal effects as 0

[33] https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020

[34] All data used to create TERA was downloaded on the 14th of May 2020.

(ii) KG_C. For each chemical in the effect data, we extract all triples connected to them using a directed crawl. This reduces the size of KG_C to a manageable size for the KGE models. Moreover, we do not deem triples not directly connected to the effect data relevant for the prediction task, and may introduce unnecessary noise. As mentioned before, PubChem contains similarities between chemicals based on chemical fingerprints, however, for our use-case it is unpractical to query them from the PubChem RDF data, therefore, we calculate similarity triples based on queried PubChem fingerprints. We use the same similarity threshold as PubChem, i.e., 0.9.

(iii) KG_S. The same steps as for KG_C are conducted for all species in the effect data.

A simple directed crawl over all predicates is sufficient to gather the interesting data in this setting as both KG_C and KG_S are primarily hierarchical and we start the crawls at the leaf nodes.

These steps reduce KG_C to 241,442 triples and KG_S to 59,673 triples. Some statistics of KG_C and KG_S, and the reduced fragments KG_C^0 and KG_S^0, are given in Table 6 (Section 5.4). In the rest of the paper were refer to TERA's reduced sub-KGs simply as KG_C and KG_S.

The transformation from TERA's KG_C and KG_S to model input is done by first dropping literals, thereafter assigning each entity an unique integer identifier $i \in [0, |E| - 1]$ which corresponds to the index of a column vector in matrices W_c or W_s (12), depending on which sub-KG is transformed. Relations are treated similarly.

7.1.2. Sampling

We use four sampling strategies of the effect data to see how the proposed classification models behave by varying the data parts that are used for training and testing. Note that, we only consider effect data where the chemical and species have mappings to external sources (e.g., NCBI Taxonomy and Wikidata, cf. Section 5.2.2) so that there is additional contextual information that can be used by the KGE models.

For each of the strategies, the validation and test sets contain unseen chemical-organism pairs with respect to the training set. The strategies, however, differ with respect to the individual organism and chemical as follows:

(i) Random 70%–15%–15% training/validation/test split on the entire dataset (i.e., the chemicals and the organisms in the validation and test will most probably be known).

(ii) Training/validation/test split where there is no overlap between chemicals in the three sets, (i.e., the chemicals in the validation and test sets are unknown). This resulted on a 72%–14%–14% split.

(iii) Training/validation/test split where there is no overlap between species in the three sets, (i.e., the species in the validation and test sets are unknown). This resulted on a 72%–14%–14% split.

(iv) Training/validation/test split with no chemicals or species overlap in the three sets. (i.e., both the chemicals and the organisms in the validation and test sets are unknown). This resulted on a 72%–14%–14% split.

Note that since we use the species and chemicals as groups to divide the data rather than the samples, the splits can vary. For strategies (i)–(iii) there is a total of 14,377 effect data samples while for strategy (iv) the total number samples is 5,621. As above, this discrepancy is down to the way we split the data. We do not split across samples, but across chemicals and species, and some chemicals are used on (close to) all species, therefore, these chemicals are removed from the data in method (iv).

There were originally 57,560 samples, however, this includes experiment duplicates, i.e., same chemical, species, and endpoint, with different chemical concentrations. This is down to large discrepancies in laboratory testing variance, therefore, we use the median concentration across the duplicates. The prior probability is approximately 0.16–0.84 (i.e., 16% of samples are labelled as non-lethal and 84% of samples are labelled as lethal) across all sampling methods. We solve this when training by randomly oversampling the minority class until the prior probabilities are 0.5–0.5 in the training set. In this case, the oversampling is performed by adding duplicates samples labelled as non-lethal. Oversampling is a well established technique used in many classification problems to remove bias during learning [12].

7.1.3. Hyper-parameters

To optimize the hyper-parameters for the KGE and classification models we use random search over the parameter ranges. We conduct 20 trials per model. Tables 7 and 8 contain the best hyper-parameters and can be used to reproduce the top performing models.

To find the best hyper-parameters for the KGE models, we use the loss as a proxy for performance, normalized by the initial loss, $RL_k = L_k / L_0$, where L_k is the training loss at epoch k, L_0 is the loss with initial weights.

We use validation loss to select the best hyper-parameter setting for the classification models presented in Section 6. The best prediction models are refitted and evaluated 10 times to reduce the influence of initial conditions on the metrics. The average and standard deviation of the metrics are presented in Section 7.2.

The hyper-parameter ranges for the KGE models are shown in Table 7 based on common values used in the literature. We conduct 20 trials of random hyper-parameters choices and validate over the validation data. In Table 8 we show the best hyper-parameters.

We can see in Table 8 that the decomposition models have similar hyper-parameters for KG_C and KG_S. As shown in Section 5.4, the major difference between KG_C and KG_S is the relational density. Therefore, it is reasonable to believe that a lower relational density KG requires more parameters to have an equivalent representation in the embedding space. We can get the same observation for the geometric models except for TransE, where the embedding dimensions are similar. ConvE is more efficient in embedding dimension than ConvKB, however, since ConvE is slightly more complex than ConvKB this is expected. The difference in negative samples could be down to our implementation of ConvE, which varies from the original. Our implementation of all models relies on 1-to-1 scoring of triples, while the implementation of ConvE originally used 1-to-M scoring, where M is the number of entities in the KG [21].

To save on intensive computation we reuse the hyper-parameters found for the KGE models in the fine-tuning optimization models presented in Section 6.3. Depending on the optimizer choice, the choice of loss weights, a_C; a_S; and a_{MLP}, is important. However, our optimizer choice has dynamic learning rates per variable, and therefore, will adapt regardless of the loss weights and we can set $a_C = a_S = a_{MLP} = 1$. Had we used, e.g., stochastic gradient descent, these variables would needed to be tuned.

KGE hyper-parameter	Search space
Loss function	$\{L_{H_2}; L_{H_2}; L_{L_1}; L_{L_2}\}$
Margin (only hinge loss)	$\{1; 2; \ldots; 10\}$
Bias (only geometric models)	$\{0; 1; \ldots; 20\}$
Embedding dimension	$\{100; 101; \ldots; 400\}$
Negative samples	$\{10; 11; \ldots; 100\}$
Prediction hyper-parameters	**Search space**
n_C (6), n_S (7), n_k (8), n (10)	$\{0; 1; 2; 3\}$
# units (6), (7), (10)	$2^k;\ k \in \{4; 5; \ldots; 10\}$
# units (8)	$2^k;\ k \in \{2; 3; 4; 5\}$

Table 7
Hyper-parameter choices for models.

7.1.4. Initialization of the fine-tuning optimization models

As presented in Section 6.3, we simultaneously train the KGE models and the MLP-based baseline model. This is done by initializing the model with (i) the weights learned in the correspondent baseline model with pre-trained embeddings, and (ii) the KG embeddings learned with the respective KGE models. For example, the Complex FT DistMult-HAKE model is initialized with the learned weights with the Complex PT DistMult-HAKE model and the pre-trained KG embeddings using DistMult and HAKE models. Then the model is further trained with a small learning rate. We found that reducing the learning rate by a factor of 100 worked well. Using this learning rate we optimize the model until convergence.

7.1.5. Simple and complex settings

As presented in Section 6.1, we use two settings in our classification models: simple and complex. This will help us isolate the effects of the KG embeddings versus the power of the MLPs model. The simple setting uses no branching layers, i.e., $n_C = n_S = n_k = 0$ and $n = 1$ as in Equations (6), (7), (8) and (10) with 128 units in the hidden dense layer. For the complex models we use random search (20 trials) to find the optimal number of layers and units out of the ranges shown in Table 7. The optimal choices for the top performing models (using one-hot and pre-trained embeddings) are shown in Table 9.

Looking at the increasing complexity of the layer configuration of the one-hot models in Table 9 we can see a correlation from the simplest sampling strategy (i) through the most challenging one (iv). The same can be seen for PT HAKE-ComplEx from setting (ii) to (iv), where the number of layers increase. Overall we can see that the layer configurations of the chemical branch is more complex than for the species branch. This indicates that the KGE models are better at representing the KG_S than KG_C.

7.2. Prediction results

In this section we present a summary of the conducted chemical effect prediction evaluation. Com-

Model	Loss function	Margin	Bias	Embedding dimension	Negative Samples
DistMult	L_{L_2}/L_{H_2}	-/2		143/383	28/43
ComplEx	L_{L_2}/L_{H_2}	-/4		163/372	27/42
HolE	L_{H_2}/L_{L_2}	6/-		188/376	30/100
TransE	L_{H_2}/L_{H_1}	4/7	14/20	226/196	23/57
RotatE	L_{H_2}/L_{H_2}	5/2	16/6	271/398	75/22
pRotatE	L_{L_2}/L_{L_2}	-/-	14/16	164/210	34/82
HAKE	L_{L_2}/L_{L_2}	-/-	12/10	108/359	56/13
ConvKB	L_{L_2}/L_{H_2}	-/5	-	248/276	18/90
ConvE	L_{H_1}/L_{H_1}	7/3	-	228/196	68/40

Table 8

Best hyper-parameters for KGE models. The two values before and after / are for the embeddings of KG_C and KG_S, respectively.

Model	Sampling	# units
Complex one-hot	(i)	(128)=(128)= =
	(ii)	(128)=(256)=(8;8)=
	(iii)	(256;128)=(128)=(4;4;4)=
	(iv)	(256;256)=(128)=(8;8)=(128)
Complex PT DistMult-HAKE (top-1 in (i))	(i)	(256;256)=(256)=(16;4)=(512;64)
Complex PT HolE-ConvKB (top-1 in (ii))	(ii)	(512;128;128)=(512)= =(64)
Complex PT HAKE-DistMult (top-1 in (iii,iv))	(iii)	(64)=(512)=(16;32)=(16)
	(iv)	(128)= =(4;8;8)=(256;128)

Table 9

Number of units in the hidden layers in the (complex) one-hot model and the top-1 prediction models with pre-trained KG embeddings. The same parameters are used for the ne-tuning models. Organized as follows: $(jb_c^1j;:::;jb_c^{n_c}j)=(jb_s^1j;:::;jb_s^{n_s}j)=(jb_k^1j;:::;jb_k^{n_k}j)=(jb^1j;:::;jb^nj)$ as in Equations (6), (7), (8), and (10)). denotes no hidden layers, e.g., $(128)=(256)=(8;8)=$ denotes $n_c = 1; n_s = 1; n_k = 2; n = 0$ and $jb_c^1j = 128$, $jb_s^1j = 256$, $jb_k^1j = 8$ and $jb_k^2j = 8$.

plete results are available at the project repository.[35] The default decision threshold is set to 0.5, that is, if a model predicts $y > 0.5$ for an input $x_{c;s;k}$ then the chemical c is considered lethal to s at a concentration k.[36]

We use several metrics to compare the different prediction models. These are Sensitivity (i.e., recall), Speci city, and Youden's index YI [84]. Precision and F-score were also considered as metrics, however, they were not representative for the performance with respect to non-harmful chemicals as the (test) data has a much larger number of positive samples (i.e., harmful chemicals) than negative samples (i.e., non-harmful chemicals).

Sensitivity and Speci city are de ned as

$$\text{Sensitivity} = \frac{TP}{TP + FN}; \tag{15}$$

$$\text{Speci city} = \frac{TN}{FP + TN}; \tag{16}$$

where TP, FN, TN, and FP are true positives, false negatives, true negatives and false positives, respectively. YI is de ned as

$$YI = \text{Sensitivity} + \text{Speci city} \quad 1: \tag{17}$$

We also present the maximized Youden's index YI_{max}, this is de ned as

$$YI_{max} = \max_t \text{Sensitivity} + \text{Speci city} \quad 1; \tag{18}$$

i.e., we maximize Youden's index based on the decision threshold t, we call this optimal threshold t_{max}. This metric is equivalent to the maximum of the Receiver operating characteristic (ROC) curve over a ran-

[35] https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020

[36] We set the decision threshold to 0.5 since the model output bias (cf. Equation (11)) will be (close to) 0.5 after training. Recall that we have oversampled the classes to reach a 0.5 prior probability during training (cf. Section 7.1.2).

dom model and can be used to select the optimal decision threshold in a production environment (based on validation data). We do not present ROC (or area under ROC, AUC) as a metric as it correlates (> 0.99) with YI_{max} in our case.

In our setting, sensitivity is a measure on how well the models identify harmful chemicals while specificity measures models' ability to identify non-harmful chemicals. Youden's index is used to capture the usefulness of a diagnostic test (or in our case, a toxicity test). A useless test will have $YI = 0$ while with $YI > 0$ a test is useful. YI is also thought of as how well informed a decision might be. Note that YI can be less that 0, but this is solved by swapping labeled classes. Similarly to how negative correlation is still useful.

Tables 10-13 show the results for each of the data sampling strategies (i)-(iv), respectively. The tables include the three best models (based on YI) for the baseline model using one-hot and pre-trained (PT) KG embeddings, and the fine-tuning (FT) models using the same combination of KGE models as the selected PT-based models. We have also included a model with middling performance (i.e., 40 out of 81 models) and the worst performing model. Note that for the PT- and FT-based models we have evaluated 81 combinations KGE_C-KGE_S of KGE models. All models were evaluated using the simple and complex MLP settings. For example, the model *Complex FT DistMult-HolE* denotes that fine-tuning was used together with the complex MLP setting, and DistMult was selected to embed the chemicals (KG_C) while HolE was used to embed the species (KG_S). We present the mean and standard deviation over 10 evaluation runs (i.e., we re-initialize and re-train the models 10 times. Results highlighted in bold are the best mean results of the corresponding metrics. Underlined results are where there is $> a$ 32% chance that a single run outperforms the best mean (i.e., one standard deviation contains 68% of results, assuming normally distributed results[37]).

Overall, models with the complex setting and fine-tuning are needed as the data sampling strategies become more challenging. Moreover, all models favour sensitivity over specificity at default decision threshold (0.5). This is down to the imbalance in the data. We can see the imbalance by n_{max}, it is > 0.5 for most models. As we use a log-loss instead of a discrete loss, this is to be expected for imbalanced data.

[37] Note that we only consider the best mean result and not the standard deviation in both directions.

For settings (iii) and (iv) the performance drops and the standard deviation increases compared to the other strategies. This large standard deviation leads to large overlaps in quantiles among top-3 models in all categories, such that, by chance, one of these models could perform best in one individual evaluation.

7.2.1. One-hot baseline models

For the sampling strategy (i) the one-hot baseline models perform well, especially, with the complex one-hot model. This complex model is equivalent in terms of YI as the best simple pre-trained model. The story is largely the same in setting (ii), where the complex one-hot model performs within 5% of the best simple pre-trained models. With strategies (iii) and (iv) the one-hot models degrade, especially in strategy (iv) where the Youden's index is near zero (< 0.1). This is expected as the one-hot baseline models lack important background information, specially for unseen chemicals and species, that the KG embeddings aim at encoding.

7.2.2. Baseline with pre-trained KG embeddings

We can see that the PT-based models do not improve YI_{max} with sampling strategy (i). The top-1 complex PT model, however, yields a better balance between sensitivity and specificity leading to an improved YI over the complex one-hot models. The two middling performing models, *Simple PT pRotatE-ConvE* and *Complex PT ComplEx-DistMult*, still retain a decent level of performance.

The results with the strategy (ii) are similar to strategy (i), the delta in YI between the simple and the complex PT-based models are about 5%. This slight improvement is due to the increased balance between sensitivity and specificity which in turn leads to a higher YI.

In the sampling strategy (iii) we can observe that the improvement of the PT-based models over the one-hot models increases. The increase is up to 25% in YI of the the best PT-based model over the best one-hot model. In addition, we observe in this strategy that the standard deviation increases, especially in specificity, leading to a large portion of the models that are within one standard deviation of the best model in terms of YI.

Finally, the impact of using a PT-based models is strengthen in strategy (iv). The delta between the one-hot and PT-based models is up to 40% in YI, and larger for YI_{max}. We see that all models struggle with specificity in this setting, this is down to the difficulty of predicting true negatives. This also leads to a larger variation, with certain models yielding standard devia-

Model	Sensitivity	Speci city	YI	YI $_{max}$	t $_{max}$
Simple one-hot	0:939 0:009	0:657 0:018	0:595 0:015	0:666 0:011	0:809 0:049
Simple PT HAKE-HAKE	0:912 0:006	0:773 0:018	0:685 0:016	0:719 0:012	0:707 0:044
Simple PT pRotatE-HAKE	0:934 0:005	0:749 0:044	0:683 0:04	0:718 0:02	0:665 0:082
Simple PT ConvE-HAKE	0:937 0:006	0:738 0:006	0:674 0:004	0:724 0:007	0:721 0:054
Simple PT pRotatE-ConvE	0:924 0:029	0:436 0:155	0:36 0:182	0:469 0:196	0:784 0:052
Simple PT RotatE-ConvE	0:997 0:003	0:024 0:035	0:021 0:035	0:195 0:111	0:812 0:086
Simple FT HAKE-HAKE	0:921 0:005	0:814 0:009	0:734 0:006	0:743 0:007	0:547 0:074
Simple FT pRotatE-HAKE	0:92 0:005	0:808 0:013	0:728 0:011	0:738 0:007	0:56 0:107
Simple FT ConvE-HAKE	0:942 0:003	0:733 0:019	0:675 0:019	0:729 0:007	0:864 0:053
Simple FT pRotatE-ConvE	0:949 0:003	0:766 0:017	0:715 0:016	0:765 0:006	0:842 0:064
Simple FT RotatE-ConvE	0:928 0:015	0:797 0:036	0:726 0:022	0:761 0:01	0:722 0:069
Complex one-hot	0:937 0:004	0:748 0:016	0:685 0:015	0:728 0:009	0:769 0:094
Complex PT DistMult-HAKE	0:895 0:008	0:817 0:008	0:713 0:007	0:723 0:008	0:456 0:088
Complex PT HAKE-ConvKB	0:927 0:006	0:784 0:017	0:711 0:013	0:739 0:009	0:686 0:109
Complex PT HolE-ConvKB	0:932 0:013	0:779 0:024	0:711 0:013	0:729 0:009	0:676 0:104
Complex PT ComplEx-DistMult	0:96 0:006	0:584 0:04	0:543 0:039	0:664 0:024	0:838 0:048
Complex PT HolE-pRotatE	0:996 0:006	0:011 0:02	0:006 0:014	0:182 0:041	0:804 0:071
Complex FT DistMult-HAKE	0:903 0:009	0:816 0:015	0:719 0:008	0:729 0:005	0:597 0:098
Complex FT HAKE-ConvKB	0:935 0:006	0:791 0:021	0:726 0:018	0:754 0:008	0:776 0:109
Complex FT HolE-ConvKB	0:895 0:01	0:835 0:016	0:73 0:01	0:739 0:011	0:61 0:123
Complex FT ComplEx-DistMult	0:927 0:005	0:78 0:018	0:707 0:016	0:742 0:011	0:797 0:093
Complex FT HolE-pRotatE	0:913 0:008	0:795 0:017	0:708 0:012	0:734 0:008	0:777 0:049

Table 10

Prediction results (mean and standard deviation over 10 runs) for sampling method (i). Bold denotes best mean result and underline denotes within one standard deviation of best result. PT pre x denotes pre-trained and FT denotes ne-tuning. Simple denotes $n_C = n_S = n_k = 0$ and $n = 1$ while in complex $n_C; n_S; n_k$ and n are hyper-parameters in Equations (6), (7), (8) and (10).

tion in the same order of magnitude as the metric (e.g. Simple FT HAKE-ComplEx).

7.2.3. Fine-tuning optimization model

The FT-based models, with some exceptions, improve the results over the PT-based models, most notably in sampling strategies (i) and (iv). For example, the FT-based model Complex FT HolE-DistMult and Simple FT HolE-ComplEx are the best models in terms of YI and YI $_{max}$ in strategy (iv), respectively. We can also see in strategies (i) and (ii) that the FT-based models improve middling and worst performing PT-based models, e.g., Simple FT RotatE-ConvE in strategy (i) improves from YI = 0:021 to YI = 0:726 using ne-tuning of the KG embeddings. The results are expected as the ne-tuned KG embeddings are tailored to the effect prediction task.

7.3. KG embedding analysis

In this section we look at correlations between KGE model choices and prediction performance. KGE models are designed to capture certain structures in the data, and this can give some explanation of which parts of the KGs are important for prediction.

First, in Table 14 we show how many times a KGE model is used when regarding the top 10 performing combinations (out of the total 81 possible combinations). We focus on the choices when using the simple MLP setting to reduce the in uence of the non-linear transforms on the embeddings.

Looking at Table 14 we can see that the KGE models used to embed the chemicals in KG $_C$ in the best performing models is distributed evenly across most models and settings. This indicates that the performance of the prediction models is not highly correlated with the use of a KGE model on KG $_C$. Referencing Table 6, the high relational density in KG $_C$ can contribute to worse performance [61] and therefore equal distribution of models in Table 14. This is different for KG $_S$. For sampling strategies (i) and (ii), HAKE is extensively used in the top models to embed KG $_S$. Since HAKE is designed to embed hierarchies, this could indicate that in

Model	Sensitivity	Specificity	YI	YI_{max}	t_{max}
Simple one-hot	0.88 0.022	0.628 0.048	0.508 0.057	0.556 0.051	0.713 0.13
Simple PT HAKE-ConvKB	0.926 0.007	0.823 0.016	0.748 0.017	0.775 0.013	0.623 0.064
Simple PT HAKE-HAKE	0.908 0.007	0.829 0.014	0.738 0.012	0.759 0.01	0.613 0.132
Simple PT pRotatE-HAKE	0.924 0.003	0.802 0.009	0.726 0.008	0.76 0.006	0.79 0.084
Simple PT RotatE-ConvKB	0.972 0.021	0.42 0.255	0.392 0.236	0.62 0.111	0.814 0.06
Simple PT RotatE-ConvE	0.997 0.004	0.021 0.057	0.018 0.054	0.22 0.088	0.824 0.095
Simple FT HAKE-ConvKB	0.909 0.003	0.883 0.006	0.792 0.006	0.803 0.004	0.556 0.138
Simple FT HAKE-HAKE	0.897 0.007	0.86 0.01	0.757 0.012	0.769 0.006	0.61 0.134
Simple FT pRotatE-HAKE	0.905 0.004	0.859 0.012	0.764 0.012	0.775 0.011	0.544 0.099
Simple FT RotatE-ConvKB	0.93 0.007	0.853 0.013	0.784 0.008	0.81 0.008	0.732 0.119
Simple FT RotatE-ConvE	0.912 0.02	0.821 0.028	0.733 0.01	0.753 0.005	0.735 0.17
Complex one-hot	0.875 0.014	0.859 0.015	0.734 0.012	0.749 0.009	0.448 0.2
Complex PT HolE-ConvKB	0.894 0.006	0.889 0.014	0.783 0.014	0.793 0.01	0.489 0.035
Complex PT pRotatE-ConvKB	0.901 0.012	0.875 0.027	0.776 0.024	0.79 0.018	0.592 0.081
Complex PT TransE-ConvKB	0.906 0.008	0.868 0.021	0.774 0.019	0.787 0.012	0.588 0.112
Complex PT ComplEx-ConvE	0.928 0.006	0.768 0.015	0.696 0.015	0.731 0.008	0.689 0.095
Complex PT ConvKB-pRotatE	0.995 0.005	0.011 0.012	0.007 0.008	0.265 0.054	0.77 0.089
Complex FT HolE-ConvKB	0.871 0.007	0.906 0.007	0.778 0.007	0.791 0.005	0.441 0.07
Complex FT pRotatE-ConvKB	0.869 0.008	0.914 0.011	0.783 0.007	0.794 0.006	0.483 0.083
Complex FT TransE-ConvKB	0.878 0.008	0.895 0.011	0.772 0.008	0.792 0.006	0.511 0.133
Complex FT ComplEx-ConvE	0.916 0.009	0.83 0.021	0.746 0.016	0.76 0.011	0.596 0.151
Complex FT ConvKB-pRotatE	0.9 0.013	0.794 0.026	0.694 0.018	0.723 0.014	0.785 0.111

Table 11
Prediction results for sampling method (ii). Same notation as Table 10.

strategies (i) and (ii) the hierarchical structure of KG_S is more important than other KG parts (e.g., habitat). KG_S has a higher entity density and lower entity entropy (Table 6) than KG_C which should lead to higher performance generally, but might also lead to larger discrepancies between models as seen in Table 14.

The use of the decomposition models increase in strategies (iii) and (iv) for the embedding of KG_S, which indicates that KG structures, other than the hierarchy, are important. Overall, DistMult and ComplEx can be used to great effect in strategies (iii) and (iv) while the geometric model, HAKE, is more successful in the less challenging strategies (i) and (ii).

7.3.1. Explained variance

Explained variance is a measure of how many principal components are required to describe all components.[38] In Figure 6, we present how the YI metric depends on the explained variance of the top-10 principal components (i.e., $\sum_{i=1}^{10} pca_i$). We show all (81 per sampling strategy) PT-based prediction model results, simple MLP setting in Figure 6a and complex setting in Figure 6b. For example, in Figure 6a, the best model in the strategy (iv), Simple PT pRotatE-ComplEx have a explained variance of 0.49 compared to the worst model, Simple PT HAKE-HAKE, with explained variance of 0.34. Coincidentally, these two points does not follow the trend lines in these figures which indicate negative correlation between YI and explained variance. The trend lines can be interpreted in two ways. First, it is counter-intuitive as we would expect more descriptive embeddings (i.e., larger explained variance, to perform better. On the other hand, the top-10 principal components may not be representative enough to capture the semantics of the KG embeddings, and thus, a large explained variance does not necessarily correlate with a high performance.

Figure 7 represents the explained variance against sensitivity. We can see that the trend is flat for strategy (iv), but positive for strategies (i)-(iii). This means that the trends in Figure 6 are explained by specificity rather than sensitivity. By balancing sensitivity

[38] We use the scikit-learn implementation [60] based on [71].

Model	Sensitivity	Speci city	YI	YI_{max}	t_{max}
Simple one-hot	0:822 0:058	0:439 0:054	0:261 0:058	0:31 0:047	0:597 0:182
Simple PT ConvKB-DistMult	0:966 0:007	0:626 0:047	0:591 0:045	0:623 0:049	0:67 0:058
Simple PT HAKE-DistMult	0:958 0:023	0:628 0:026	0:586 0:033	0:626 0:045	0:613 0:092
Simple PT ConvKB-TransE	0:969 0:009	0:614 0:048	0:583 0:04	0:642 0:01	0:643 0:059
Simple PT ConvE-RotatE	0:934 0:055	0:276 0:026	0:209 0:043	0:273 0:071	0:596 0:13
Simple PT HolE-HAKE	0:88 0:089	0:115 0:083	0:005 0:075	0:077 0:057	0:783 0:18
Simple FT ConvKB-DistMult	0:947 0:014	0:667 0:02	0:614 0:013	0:645 0:011	0:736 0:087
Simple FT HAKE-DistMult	0:947 0:012	0:662 0:035	0:609 0:031	0:634 0:026	0:701 0:132
Simple FT ConvKB-TransE	0:934 0:009	0:68 0:018	0:615 0:014	0:642 0:015	0:687 0:065
Simple FT ConvE-RotatE	0:915 0:013	0:454 0:028	0:369 0:027	0:402 0:028	0:658 0:083
Simple FT HolE-HAKE	0:931 0:009	0:118 0:036	0:049 0:038	0:171 0:038	0:882 0:127
Complex one-hot	0:796 0:028	0:571 0:041	0:367 0:054	0:398 0:043	0:526 0:076
Complex PT HAKE-DistMult	0:969 0:016	0:642 0:044	0:61 0:034	0:643 0:026	0:675 0:105
Complex PT pRotatE-ComplEx	0:929 0:024	0:668 0:048	0:597 0:048	0:62 0:046	0:526 0:145
Complex PT ConvKB-DistMult	0:965 0:013	0:631 0:078	0:597 0:07	0:627 0:039	0:597 0:149
Complex PT ComplEx-HolE	0:991 0:01	0:237 0:106	0:228 0:098	0:45 0:028	0:721 0:047
Complex PT ComplEx-HAKE	0:9 0:055	0:097 0:047	0:003 0:064	0:133 0:081	0:696 0:22
Complex FT HAKE-DistMult	0:932 0:011	0:69 0:024	0:622 0:023	0:652 0:022	0:706 0:134
Complex FT pRotatE-ComplEx	0:931 0:025	0:672 0:042	0:602 0:045	0:631 0:037	0:627 0:157
Complex FT ConvKB-DistMult	0:953 0:008	0:642 0:027	0:596 0:027	0:625 0:028	0:753 0:138
Complex FT ComplEx-HolE	0:898 0:035	0:591 0:064	0:489 0:042	0:521 0:027	0:612 0:156
Complex FT ComplEx-HAKE	0:88 0:032	0:255 0:026	0:135 0:034	0:204 0:06	0:775 0:268

Table 12
Prediction results for sampling method (d) . Same notation as Table 10.

(a) Simple PT models.

(b) Simple PT models.

Fig. 6. Relation between explained variance using 10 principal components and model performance represented as

Model	Sensitivity	Speci city	YI	YI $_{max}$	t$_{max}$
Simple one-hot	0:612 0:096	0:421 0:107	0:033 0:14	0:113 0:076	0:555 0:306
Simple PT HAKE-ComplEx	0:971 0:011	0:361 0:065	0:332 0:056	0:546 0:031	0:89 0:042
Simple PT pRotatE-ComplEx	0:972 0:008	0:36 0:079	0:332 0:074	0:527 0:045	0:852 0:04
Simple PT HolE-ComplEx	0:965 0:032	0:363 0:068	0:328 0:063	0:549 0:075	0:856 0:077
Simple PT pRotatE-RotatE	0:917 0:01	0:168 0:016	0:084 0:013	0:151 0:021	0:779 0:182
Simple PT HAKE-HAKE	0:8 0:095	0:128 0:066	0:072 0:07	0:033 0:027	0:736 0:321
Simple FT HAKE-ComplEx	0:963 0:01	0:423 0:102	0:386 0:096	0:57 0:03	0:875 0:079
Simple FT pRotatE-ComplEx	0:954 0:009	0:5 0:058	0:454 0:052	0:569 0:024	0:854 0:073
Simple FT HolE-ComplEx	0:965 0:007	0:418 0:058	0:383 0:053	0:571 0:042	0:9 0:046
Simple FT pRotatE-RotatE	0:806 0:039	0:229 0:027	0:035 0:016	0:131 0:032	0:782 0:157
Simple FT HAKE-HAKE	0:893 0:046	0:104 0:051	0:003 0:031	0:037 0:033	0:588 0:332
Complex one-hot	0:656 0:069	0:422 0:075	0:078 0:053	0:124 0:036	0:645 0:178
Complex PT HAKE-DistMult	0:923 0:013	0:434 0:059	0:357 0:052	0:488 0:074	0:808 0:07
Complex PT HolE-DistMult	0:949 0:016	0:38 0:084	0:33 0:076	0:443 0:089	0:805 0:07
Complex PT ConvKB-DistMult	0:942 0:01	0:387 0:038	0:329 0:039	0:484 0:066	0:817 0:052
Complex PT HolE-RotatE	0:932 0:014	0:15 0:018	0:082 0:023	0:168 0:015	0:861 0:064
Complex PT TransE-HAKE	0:756 0:047	0:19 0:077	0:054 0:089	0:057 0:046	0:742 0:253
Complex FT HAKE-DistMult	0:925 0:021	0:513 0:064	0:437 0:058	0:522 0:034	0:83 0:09
Complex FT HolE-DistMult	0:926 0:015	0:536 0:03	0:462 0:03	0:543 0:039	0:81 0:084
Complex FT ConvKB-DistMult	0:933 0:01	0:525 0:065	0:459 0:063	0:55 0:04	0:746 0:122
Complex FT HolE-RotatE	0:863 0:057	0:194 0:053	0:057 0:015	0:11 0:021	0:81 0:278
Complex FT TransE-HAKE	0:892 0:027	0:075 0:043	0:033 0:049	0:072 0:048	0:958 0:077

Table 13
Prediction results sampling method (iv). Same notation as Table 10.

KGE model	# uses (i)	# uses (ii)	# uses (iii)	# uses (iv)
DistMult	1=0	0=1	1=7	0=4
ComplEx	1=1	1=3	2=1	1=5
HolE	2=0	1=0	1=0	1=0
Total decomposition	4=1	2=4	4=8	2=9
TransE	1=0	2=0	1=2	0=0
RotatE	0=0	0=0	0=0	1=0
pRotatE	1=0	1=0	1=0	3=0
HAKE	2=8	3=5	1=0	2=0
Total geometric	4=8	6=5	3=2	5=0
ConvKB	1=1	0=1	2=0	0=1
ConvE	1=0	2=0	1=0	2=0
Total convolutional	2=1	2=1	3=0	2=1

Table 14
Usage of KGE models for each sampling strategy in simple MLP setting in top-10 performing combinations. Note that, there is one model for the KG$_C$ and one for KG$_S$, such that there is a total of 20 models per sampling strategy. Notation: `used in KG$_C$ / used in KG$_S$', e.g., HAKE, 2=8 in sampling strategy (i), indicates that HAKE is used to embed KG$_C$ 2 out of top-10 combinations and it is used to embed KG$_S$ 8 out of top-10 combinations.

(a) Simple PT models.

(b) Complex PT models.

Fig. 7. Relation between explained variance using 10 principal components and model performance represented as sensitivity.

and speci city, i.e., Y_{max} as seen in Figure 8, the rate of change is reduced compared to Y_{min} in Figure 6.

7.4. Example predictions

Table 15 shows a few examples of correct (TP and TN) and incorrect predictions (FN and FP).

Benthiocarb and permethrin are both biocides with different targets: benthiocarb is a herbicide and permethrin is an insecticide. It is therefore not surprising that benthiocarb has a low predicted effect on sea urchins, while permethrin has a severe effect on bivalves.

There are several possible explanations for the failed predictions. A wrong prediction of potassium chloride toxicity to a marine copepod (Megacyclops viridis) could be due to the prediction model not being accurate enough for metal salts, or the copepod species being particularly sensitive to changes in osmolarity due to salt content. The wrong prediction of lack of herbicide toxicity (i.e., carfentrazone-ethyl) to a ower (i.e., eudicots) could be due to the fact that owers, and plants in general, are severely underrepresented in the available effect prediction data.

8. Discussion

We have introduced the Toxicological Effect and Risk Assessment (TERA) knowledge graph and shown how we can directly use it in chemical effect prediction. The use of TERA improves the PT-based prediction models over the one-hot baselines. In the most challenging data sampling strategies, we have also seen the bene ts of creating tailored (i.e., ne-tuned) KG embeddings in the FT-based prediction models.

8.1. TERA knowledge graph

The constructed knowledge graph consists of several sources from the ecotoxicological domain. There are three major parts in TERA: the effects data, the chemical data, and the species taxonomic data. Integrating each part has different challenges. The chemical and pharmacological communities have come a long way in annotating their data as knowledge graphs and ontologies. Here, selecting the correct subsets to work with the chemical effect prediction data was a major challenge. This had to be done based on mappings between effect data and chemical data that were extracted from Wikidata. We selected a relatively small subset of the chemical sub-KG to facilitate faster model training, however, still larger than the extracted fragment from the species sub-KG. The species sub-KG was created from tabular data and cleaned by removing several annotation labels with redundant information. This sub-KG was aligned using ontology alignment systems to the species taxonomy in the ef-

(a) Simple PT models. (b) Complex PT models.

Fig. 8. Relation between explained variance using 10 principal components and model performance represented as K_{max}

Chemical	Species	log(k)	Predicted	Lethal	Classi cation
D001556 (hexachlorocyclohexane)	59899 (walking cat sh)	3:4	0:97	1 (yes)	TP
C037925 (benthiocarb)	7965 (sea urchins)	0:9	0:2	0 (no)	TN
D026023 (permethrin)	378420 (bivalves)	0:7	0:96	1 (yes)	TP
D011189 (potassium chloride)	938113 (megacyclops viridis)	6:7	0:27	1 (yes)	FN
C427526 (carfentrazone-ethyl)	208866 (eudicots)	0:9	0:82	0 (no)	FP
D010278 (parathion)	201691 (green sun sh)	0:9	0:86	0 (no)	FP

Table 15
Example predictions by Complex FT HolE-DistMult (best model) for sampling strategy (e)

fects sub-KG. This required pre-processing of the KG, where it was divided into smaller parts such that the selected systems could perform the alignment. We used several standard ontologies to facilitate the transformation of the effect data into a knowledge graph. This involved not only automatic processes, but also an important amount of manual work.

Integrating more data into TERA involves the creation of mappings to the existing data. This is possible for a large amount of chemical datasets as Wikidata links multiple datasets, e.g., the chemical compound diethyltoluamide (wd:Q408389) has 35 distinct identi ers. Biological data, both taxonomic and effects, might be harder to align to TERA as these mappings are not available in Wikidata. Here is where ontology alignment systems play an important role to ll this gap.

The additional integrated data will give larger coverage of the domain, and thereby, improve model performance. However, adding more data will also increase the memory and time requirements of KGE models, which we have bypassed in this work by reducing TERA to only relevant parts.

Adding additional domain knowledge is also critical in other applications, such as using TERA for data access.

8.2. Performance of prediction models

We have shown that the ability to embed some structure types of different KGE models largely impact the prediction models. We see that some KGE models fail to capture the semantics of the chemicals and the species, which leads to similar performance to the one-

hot baselines. Moreover, in a few isolated cases the performance is reduced further which leads us to believe that the embeddings collapse in one or some dimensions, making it impossible to distinguish among entities.

We suspect that the even distribution of KGE models to embed KG_C (Table 14) in most settings is likely down to the structure of KG_C. This sub-KG has, unlike KG_S's tree structure, a forest structure, and models that can deal with trees (as in KG_S) fail here, e.g., an entity in KG_C can have multiple parents, but only one grand-parent. In this case, some models may create very similar or the same embeddings for the parent nodes.

9. Conclusions and future work

TERA is a novel knowledge graph which includes large amounts of data required by ecological risk assessment. We have also conducted an extensive evaluation of KGE embedding models in a novel and very challenging application domain, where we have shown the value of using TERA in an ecotoxicological effect prediction task. The fine-tuning optimization model architecture to adapt the KG embeddings to the prediction task has, to our knowledge, not been applied elsewhere.

9.1. Value for the ecotoxicology community

The creation of TERA is of great importance to future effect modelling and computational risk assessment approaches within ecotoxicology, whose strategic goal is designing and developing prediction models to assess the hazard and risks of chemicals and their mixtures where traditional laboratory data cannot easily be acquired.

A great effort in the hazard and risk assessment of chemicals is the reduction of regulatory-mandated animal testing. Wide-scale predictive approaches, as described here, answer a direct and current need for generalized prediction frameworks, that can aid in identifying especially sensitive species and toxic chemicals. At the Norwegian Institute for Water Research (NIVA), TERA will be used in this regard and will support several research projects.

In environmental risk assessment it is often unfeasible to assess the hazard and risk a chemical poses to a local species in the environment. These species may not be suitable for lab testing, or may even be endangered and thus are protected by national or international legislation. The currently presented work provides an in silico approach to predict the hazard to such species based on the taxonomic position of the species within the tree of life.

From an economic perspective, TERA and the prediction models are useful tools to evaluate new industrial chemicals during the synthetic in silico stage. Candidate chemicals can be evaluated for their potential environmental hazard, which is in line with the Green Chemistry initiatives by authorities such as the European Parliament or the US Environmental Protection Agency.

The effect prediction using TERA is also in line with a larger shift in ecological risk assessment towards the use of artificial intelligence [79]. We also believe the development of TERA contributes to a methodological change in the community, and encourages others to make their data interoperable.

9.2. TERA as background knowledge

As mentioned, in this work we use TERA directly in prediction models. However, TERA could be used as background knowledge to improve many emerging techniques for toxicity prediction (e.g., [64]). These methods often use chemical features, images, fingerprints and so on as input, and machine learning methods such as Convolutional Neural Networks and Random Forests as prediction models [80, 83]. These models are often uninterpretable, and the predictions lack domain explanations. TERA can also provide context for machine learning tasks such as pre-processing, feature extraction, transfer and zero/few-shot learning. Furthermore, the knowledge graph is a possible source for the (semantic) explanation of the predictions (e.g., [43]).

9.3. Benchmarking KG embedding models

We have shown that embedding TERA brings new challenges to state-of-the-art KGE models with respect to capturing the semantics of the chemicals and the species. Furthermore, as shown in Section 5.4 the sparsity-related measures indicate that TERA represent an interesting KG. KGE models could be benchmarked in a standard KG completion task or in a specific task such as the chemical effect prediction.

9.4. Value to the ontology alignment community

As mentioned in Section 5.2, there does not exist a complete and public alignment between ECOTOX species and the NCBI Taxonomy. Therefore the computed mappings can also be seen as a very relevant resource to the ecotoxicology community. The used alignment techniques achieve high scores for recall over the available (incomplete) reference mappings. However, aligning such large and challenging datasets requires preprocessing before ontology alignment systems can cope with them. We removed all nodes which did not share a word (or shared only a stop word) in labels across the two taxonomies. This quartered the size of ECOTOX and reduced NCBI Taxonomy 50 fold. However, the possible alignment between entities without labels is lost when reducing the dataset size. Thus, the alignment of ECOTOX and NCBI Taxonomy has the potential of becoming a new track of the Ontology Alignment Evaluation Initiative (OAEI) [51] to push the limits of large scale ontology alignment tools. Furthermore, the output of the different OAEI participants could be merged into a rich consensus alignment that could become the reference to integrate ECOTOX and NCBI Taxonomy.

9.5. Future work

We plan to extend TERA to include a larger part of ChEBI (which ChEMBL is a part of). ChEBI includes relevant data on the interaction between chemicals and species at a cellular level, which may be very important for chemical effect prediction. In this work we only consider effect data from ECOTOX as this is the largest data set available, however, the inclusion of e.g., TOXCAST [74] is in our interest. New sources will always bring more coverage of the domain and will improve TERA for prediction, as background knowledge, and for data access.

We plan to evaluate the effect prediction under different parts of TERA, i.e., which sources in TERA provide value and which do not contribute in terms of the effect prediction. A similar effort in exploring different KG crawling techniques has been explored in [66]. In a similar vain, we plan to evaluate how materialization, via OWL reasoning, of TERA's implicit triples affects prediction performance.

Finally, as mentioned already, some KGE models cannot deal with parts of the structure of TERA. An in-depth analysis of this is an interesting direction for future research. This could be solved by embedding the hierarchy separately, e.g., [49], or imposing restrictions on the embeddings, such as a minimum distance constraint.

9.6. Resources

We encourage feedback from domain researchers on extensions to TERA and associated tools.

A snapshot of TERA is available at

https://doi.org/10.5281/zenodo.3559865

This snapshot does not include data that is impractical to re-share (i.e., partial KG_C as described in Section 5). However, we include the full KG_E and KG_S.

All the material related to this project is available at

https://github.com/NIVA-Knowledge-Graph/

Source codes to create TERA are available in the TERA GitHub repository. The prediction models and data used for prediction can be found in the KGs_and_Effect_Prediction_2020 GitHub repository. The prediction models require the implementation of the KGE models from the KGE-Keras GitHub repository.

Acknowledgements

This work is supported by the grant 272414 from the Research Council of Norway (RCN), the MixRisk project (Research Council of Norway, project 268294), SIRIUS Centre for Scalable Data Access (Research Council of Norway, project 237889), Samsung Research UK, Siemens AG, and the EPSRC projects AnaLOG (EP/P025943/1), OASIS (EP/S032347/1), UK FIRES (EP/S019111/1) and the AIDA project (Alan Turing Institute).

References

[1] M. Abd Nikooie Pour, A. Algergawy, R. Amini, D. Faria, I. Fundulaki, I. Harrow, S. Hertling, E. Jiménez-Ruiz, C. Jonquet, N. Karam, A. Khiat, A. Laadhar, P. Lambrix, H. Li, Y. Li, P. Hitzler, H. Paulheim, C. Pesquita, T. Saveta, P. Shvaiko, A. Splendiani, É. Thiéblin, C. Trojahn, J. Vatascinová, B. Yaman, O. Zamazal, and L. Zhou. Results of the Ontology Alignment Evaluation Initiative 2020. In *15th International Workshop on Ontology Matching*, pages 92–138, 2020.

[2] A. Agibetov and M. Samwald. Global and local evaluation of link prediction tasks with neural embeddings. In *4th Workshop on Semantic Deep Learning (ISWC workshop)*, pages 89–102, 2018.

[3] A. Algergawy, M. Cheatham, D. Faria, A. Ferrara, I. Fundulaki, I. Harrow, S. Hertling, E. Jiménez-Ruiz, N. Karam, A. Khiat, P. Lambrix, H. Li, S. Montanelli, H. Paulheim, C. Pesquita, T. Saveta, D. Schmidt, P. Shvaiko, A. Splendiani, É. Thiéblin, C. Trojahn, J. Vatascinová, O. Zamazal, and L. Zhou. Results of the ontology alignment evaluation initiative 2018. In *13th International Workshop on Ontology Matching*, pages 76–116, 2018.

[4] A. Algergawy, D. Faria, A. Ferrara, I. Fundulaki, I. Harrow, S. Hertling, E. Jiménez-Ruiz, N. Karam, A. Khiat, P. Lambrix, H. Li, S. Montanelli, H. Paulheim, C. Pesquita, T. Saveta, P. Shvaiko, A. Splendiani, É. Thiéblin, C. Trojahn, J. Vatascinová, O. Zamazal, and L. Zhou. Results of the ontology alignment evaluation initiative 2019. In *14th International Workshop on Ontology Matching*, pages 46–85, 2019.

[5] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, and J. Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a uni ed framework, 2020.

[6] M. Alshahrani, M. A. Khan, O. Maddouri, A. R. Kinjo, N. Queralt-Rosinach, and R. Hoehndorf. Neuro-symbolic representation learning on biological knowledge graphs. *Bioinformatics*, 33(17):2723–2730, 2017.

[7] H. Arnaout and S. Elbassuoni. Effective searching of rdf knowledge graphs. *Journal of Web Semantics*, 48:66 – 84, 2018.

[8] T. Benson. *Principles of Health Interoperability HL7 and SNOMED*. Health Information Technology Standards. Springer London, 2012.

[9] K. Blagec, H. Xu, A. Agibetov, and M. Samwald. Neural sentence embedding models for semantic similarity estimation in the biomedical domain. *BMC Bioinformatics*, 20(1):178, Apr 2019.

[10] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery.

[11] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[12] P. Branco, L. Torgo, and R. P. Ribeiro. A Survey of Predictive Modeling on Imbalanced Domains. *ACM Comput. Surv.*, 49(2):31:1–31:50, 2016.

[13] A. Breit, S. Ott, A. Agibetov, and M. Samwald. OpenBioLink: a benchmarking framework for large-scale biomedical link prediction. *Bioinformatics*, 36(13):4097–4098, 04 2020.

[14] J. Chen, P. Hu, E. Jiménez-Ruiz, O. M. Holter, D. Antonyrajah, and I. Horrocks. OWL2Vec*: Embedding of OWL ontologies. *CoRR*, abs/2009.14654, 2020.

[15] J. Chen, E. Jimenez-Ruiz, I. Horrocks, D. Antonyrajah, A. Hadian, and J. Lee. Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision. In *European Semantic Web Conference (ESWC)*. Springer, 2021.

[16] X. Chen, M.-X. Liu, and G.-Y. Yan. Drug–target interaction prediction by random walk on the heterogeneous network. *Mol. BioSyst.*, 8:1970–1978, 2012.

[17] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[18] T. F. Coleman and J. J. Moré. Estimation of sparse jacobian matrices and graph coloring blems. *SIAM Journal on Numerical Analysis*, 20(1):187–209, 1983.

[19] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Web Semantics*, 6(4):309–322, 2008.

[20] J. David, J. Euzenat, F. Scharffe, and C. T. dos Santos. The alignment API 4.0. *Semantic Web*, 2(1):3–10, 2011.

[21] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. Convolutional 2D knowledge graph embeddings. *AAAI 2018*, 02 2018.

[22] J. A. Doering, S. Lee, K. Kristiansen, L. Evenseth, M. G. Barron, I. Sylte, and C. A. LaLone. In Silico Site-Directed Mutagenesis Informs Species-Speci c Predictions of Chemical Susceptibility Derived From the Sequence Alignment to Predict Across Species Susceptibility (SeqAPASS) Tool. *Toxicological Sciences*, 166(1):131–145, 07 2018.

[23] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610, 2014. Evgeniy Gabrilovich Wilko Horn Ni Lao Kevin Murphy Thomas Strohmann Shaohua Sun Wei Zhang Geremy Heitz.

[24] A. Z. Dudek, T. Arodz, and J. Gálvez. Computational Methods in Developing Quantitative Structure-Activity Relationships (QSAR): A Review. *Combinatorial Chemistry & High Throughput Screening*, 9(3):213–228, 2006.

[25] J. Euzenat and P. Shvaiko. *Ontology Matching, Second Edition*. Springer, 2013.

[26] D. Faria, E. Jiménez-Ruiz, C. Pesquita, E. Santos, and F. M. Couto. Towards annotating potential incoherences in bioportal mappings. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*, pages 17–32, 2014.

[27] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto. The agreementmakerlight ontology matching system. In R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, P. De Leenheer, and D. Dou, editors, *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 527–541, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[28] J. Fukuchi, A. Kitazawa, K. Hirabayashi, and M. Honma. A practice of expert review by read-across using QSAR Toolbox. *Mutagenesis*, 34(1):49–54, 01 2019.

[29] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, and C. Steinbeck. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic acids research*, 44(D1):214—9, 2016.

[30] K. Hayashi and M. Shimbo. On the equivalence of holographic and complex embeddings for link prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 554–559, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[31] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, and D. Tchekhovskoi. InChI, the IUPAC International Chemical Identi er. *Journal of Cheminformatics*, 7(1):23, 2015.

[32] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, and A. Zimmermann. Knowledge Graphs. CoRR, abs/2003.02320, 2020.

[33] E. Jiménez-Ruiz and B. Cuenca Grau. LogMap: Logic-Based and Scalable Ontology Matching. In 10th International Semantic Web Conference, pages 273–288, 2011.

[34] E. Jiménez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. B. Llavori. Logic-based assessment of the compatibility of UMLS ontology sources. J. Biomedical Semantics, 2(S-1):S2, 2011.

[35] E. Jiménez-Ruiz, B. Cuenca Grau, Y. Zhou, and I. Horrocks. Large-scale interactive ontology matching: Algorithms and implementation. In the 20th European Conference on Arti cial Intelligence (ECAI), pages 444–449. IOS Press, 2012.

[36] R. Kadlec, O. Bajgar, and J. Kleindienst. Knowledge base completion: Baselines strike back. CoRR, abs/1705.10744, 2017.

[37] S. Kim, E. E. Bolton, and S. H. Bryant. Similar compounds versus similar conformers: complementarity between PubChem 2-D and 3-D neighboring sets. Journal of Cheminformatics, 8(1):62, Nov 2016.

[38] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton. PubChem 2019 update: improved access to chemical data. Nucleic Acids Research, 47(D1):D1102–D1109, 10 2018.

[39] D. Kingma and J. Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations, 12 2014.

[40] M. Kulmanov, W. Liu-Wei, Y. Yan, and R. Hoehndorf. El embeddings: geometric construction of models for the description logic el++. In Proceedings of the 28th International Joint Conference on Arti cial Intelligence, pages 6103–6109. AAAI Press, 2019.

[41] C. LaLone, D. Villeneuve, H. Helgen, and G. Ankley. Sequence alignment to predict across-species susceptibility. SETAC Europe, Basel, SWITZERLAND, May 11 - 15, 2014.

[42] M. lare (Skolelaboratoriet i realfag ved Universitetet i Bergen). Smyr i ferskvann. Accessed 11.06.2020.

[43] F. Lécué and J. Wu. Semantic explanations of predictions. CoRR, abs/1805.10587, 2018.

[44] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2):167–195, 2015.

[45] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady 10:707, Feb 1966.

[46] X. Liang, D. Li, M. Song, A. Madden, Y. Ding, and Y. Bu. Predicting biomedical relationships using the knowledge and graph embedding cascade model. PLOS ONE, 14(6):1–23, 06 2019.

[47] G. A. Miller. Wordnet: A lexical database for english. Commun. ACM, 38(11):39–41, Nov. 1995.

[48] S. Mohamed, E. Muñoz, V. Novacek, and P.-Y. Vandenbussche. Loss functions in knowledge graph embedding models. In Workshop on Deep Learning for Knowledge Graphs, 2019.

[49] S. Mumtaz and M. Giese. Frequency-based vs. knowledge-based similarity measures for categorical data. AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2020.

[50] E. B. Myklebust, E. Jiménez-Ruiz, J. Chen, R. Wolf, and K. E. Tollefsen. Knowledge Graph Embedding for Ecotoxicological Effect Prediction. In Int'l Sem. Web Conf. (ISWC), 2019. Best Student Paper in the In-Use track.

[51] E. B. Myklebust, E. Jiménez-Ruiz, J. Chen, R. Wolf, and K. E. Tollefsen. Ontology alignment in ecotoxicological effect prediction. In 15th International Workshop on Ontology Matching, 2020.

[52] E. B. Myklebust, E. Jimenez-Ruiz, C. Jiaoyan, R. Wolf, and K. E. Tollefsen. Toxicological Effect and Risk Assessment (TERA) Knowledge Graph, November 2020. (Version 1.1.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4244313.

[53] M. Nayyeri, C. Xu, Y. Yaghoobzadeh, H. S. Yazdi, and J. Lehmann. Toward understanding the effect of loss function on then performance of knowledge graph embedding, 2019.

[54] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 327–333, 2018.

[55] M. Nickel, L. Rosasco, and T. A. Poggio. Holographic embeddings of knowledge graphs. CoRR, abs/1510.04935, 2015.

[56] NLM. Medical Subject Headings (MeSH) RDF, 2020. https://id.nlm.nih.gov/mesh/.

[57] C. S. Parr, N. Wilson, P. Leary, K. Schulz, K. Lans, L. Walley, J. Hammock, A. Goddard, J. Rice, and M. Studer. The encyclopedia of life v2: Providing global access to knowledge about life on earth., 2014.

[58] C. S. Parr, N. Wilson, P. Leary, K. S. Schulz, K. Lans, L. Walley, J. A. Hammock, A. Goddard, J. Rice, M. Studer, J. T. G. Holmes, and J. Robert J. Corrigan. The Encyclopedia of Life v2: Providing Global Access to Knowledge About Life on Earth. Biodiversity Data Journal, 2:e1079, 2014.

[59] R. Parthasarathi and A. Dhawan. Chapter 5 - in silico approaches for predictive toxicology. In A. Dhawan and S. Kwon, editors, In Vitro Toxicology, pages 91 – 109. Academic Press, 2018.

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[61] J. Pujara, E. Augustine, and L. Getoor. Sparsity and noise: Where knowledge graph embeddings fall short. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1751–1756, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[62] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. ACM Trans. Knowl. Discov. Data, 15(2):14:1–14:49, 2021.

[63] E. W. Sayers, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, M. Feolo, L. Y. Geer, W. Helmberg, Y. Kapustin, D. Landsman, D. J. Lipman, T. L. Madden, D. R. Maglott, V. Miller, I. Mizrachi, J. Ostell, K. D. Pruitt, G. D. Schuler, E. Sequeira, S. T. Sherry, M. Shumway, K. Sirotkin,

A. Souvorov, G. Starchenko, T. A. Tatusova, L. Wagner, E. Yaschenko, and J. Ye. Database resources of the National Center for Biotechnology Information. Nucleic Acids Research, 37(suppl_1):D5–D15, 10 2008.
[64] A. K. Sharma, G. N. Srivastava, A. Roy, and V. K. Sharma. Toxim: A toxicity prediction tool for small molecules developed using machine learning and chemoinformatics approaches. Frontiers in pharmacology, 8:880–880, Nov 2017.
[65] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng., 25(1):158–176, 2013.
[66] N. P. O. Skrindebakke. Understanding the Role of Background Knowledge in Predictions, 2020. Master's thesis.
[67] F. Z. Smaili, X. Gao, and R. Hoehndorf. Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics, 35(12):2133–2140, 2019.
[68] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In 16th International Conference on the World Wide Web, pages 697–706, 2007.
[69] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In International Conference on Learning Representations, 2019.
[70] M. Swain et al. PubChemPy: Python wrapper for the pubchem pug rest api., 2014. [Online; accessed 15.08.2019].
[71] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. Journal of the Royal Statistical Society. Series B (Statistical Methodology), 61(3):611–622, 1999.
[72] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. CoRR, abs/1606.06357, 2016.
[73] U.S. Environmental Protection Agency. Ecotox user guide: Ecotoxicology knowledgebase system. version 5.3., 2020.
[74] U.S. Environmental Protection Agency. ToxCast & Tox21 Summary Files from invitrodb_v3, 2020.
[75] D. Vrandecic and M. Krötzsch. Wikidata: a free collaborative knowledgebase. Commun. ACM, 57(10):78–85, 2014.
[76] A. Waagmeester, G. Stupp, S. Burgstaller-Muehlbacher, B. M. Good, M. Grif th, O. L. Grif th, K. Hanspers, H. Hermjakob, T. S. Hudson, K. Hybiske, S. M. Keating, M. Manske, M. Mayers, D. Mietchen, E. Mitraka, A. R. Pico, T. Putman, A. Riutta, N. Queralt-Rosinach, L. M. Schriml, T. Shafee, D. Slenter, R. Stephan, K. Thornton, G. Tsueng, R. Tu, S. Ul-Hasan, E. Willighagen, C. Wu, and A. I. Su. Wikidata as a knowledge graph for the life sciences. eLife, 9:e52614, Mar 2020.
[77] Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. IEEE Trans. Knowl. Data Eng., 29(12):2724–2743, 2017.
[78] E. Willighagen. InChIKey collision: the DIY copy/pastables, 2011.
[79] C. Wittwehr, P. Blomstedt, J. P. Gosling, T. Peltola, B. Raffael, A.-N. Richarz, M. Sienkiewicz, P. Whaley, A. Worth, and M. Whelan. Arti cial intelligence for chemical risk assessment. Computational Toxicology, page 100114, 2019.
[80] Y. Wu and G. Wang. Machine Learning Based Toxicity Prediction: From Chemical Structural Description to Transcriptome Analysis. International journal of molecular sciences, 19(8):2358, Aug 2018.
[81] Z. Wu, W. Lu, D. Wu, A. Luo, H. Bian, J. Li, W. Li, G. Liu, J. Huang, F. Cheng, and Y. Tang. In silico prediction of chemical mechanism of action via an improved network-based inference method. British Journal of Pharmacology, 173(23):3372–3385, 2016.
[82] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. CoRR, abs/1412.6575, 2015.
[83] H. Yang, L. Sun, W. Li, G. Liu, and Y. Tang. In silico prediction of chemical toxicity for drug design using machine learning methods and structural alerts. Frontiers in chemistry, 6:30, 2018.
[84] W. J. Youden. Index for rating diagnostic tests. Cancer, 3(1):32–35, 1950.
[85] Z. Zhang, J. Cai, Y. Zhang, and J. Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction, 2019.

Appendix A. Knowledge Graph Embedding Models

In this work, we use 9 KGE models of three major categories: decomposition models, geometric models, and convolutional models. The interested reader please refer to [62] for a comprehensive survey.

A.1. Notation

Throughout this section we use bold letters to denote vectors while matrices are denoted as. Common notation for all KGE models are k, k_n for the n-th norm, $\langle x; y\rangle$ for the inner product (dot product) between x and y, $[x; y]$ is the concatenation of x and y, $\bar{x}$ indicates the reshape of a one-dimensional vector into a two-dimensional image (not in HolE where it represent the complex conjugate), nally, $vec(x)$ reshapes a matrix into a one-dimensional vector.

The vector representation of an entity and a relation are noted as e_e and e_p, respectively. These vectors are either in R^k or C^k, where k is the embedding dimension.

A.2. Decomposition models

DistMult. Developed by [82] and shown to have state-of-the-art performance on link prediction tasks under optimal hyper-parameters [36]. This model represent the score of a triple as an Hadaman product (dot product) of the vectors representing the subject, predicate, and object of a triple.

$$S_{DistMult}(s; p; o) = \langle e_s; e_p; e_o\rangle \quad (19)$$

This model does not take the direction of the relation into account, i.e. $S_{DistMult}(s; p; o) = S_{DistMult}(o; p; s)$.

ComplEx. This model use the same scoring function as DistMult [72]. However, the entity vector representation are in the complex space $e_s, (e_p; e_o \in \mathbb{C}^k)$ and therefore, the drawback of lacking directionality in DistMult is solved.

$$\begin{aligned} S_{ComplEx}(s,p,o) &= \langle e_s, e_p, e_o \rangle \\ &= \langle \hat{A}(e_s) + i\acute{A}(e_s), \hat{A}(e_s) \rangle \\ &+ \langle i\acute{A}(e_s), \hat{A}(e_p) + i\acute{A}(e_o) \rangle \\ &= \langle \hat{A}(e_s), \hat{A}(e_p), \hat{A}(e_o) \rangle \\ &+ \langle \acute{A}(e_s), \hat{A}(e_p), \acute{A}(e_o) \rangle \\ &+ \langle \hat{A}(e_s), \acute{A}(e_p), \acute{A}(e_o) \rangle \\ &+ \langle \acute{A}(e_s), \acute{A}(e_p), \hat{A}(e_o) \rangle \end{aligned} \tag{20}$$

where $i = \sqrt{-1}$ and, $\hat{A}(x)$ and $\acute{A}(x)$ are the real and complex parts of x, respectively. We can easily see that $S_{ComplEx}(e_s, e_p, e_o) = S_{DistMult}(e_s, e_p, e_o)$ if $\acute{A}(e_s) = \acute{A}(e_p) = \acute{A}(e_o) = 0$.

HolE. The Holographic embedding model is described in [55], and use a circular correlation scoring function

$$S_{HolE}(s,p,o) = e_p^T [e_s \star e_o] \quad e_s \star e_o \tag{21}$$

$$= \mathcal{F}^{-1}[\overline{\mathcal{F}(e_s)} \quad \mathcal{F}(e_o)] \tag{22}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier transform and its inverse, for this model we use $\overline{x}$ as the elementwise complex conjugate, $\cdot$ denotes the Hadamard product (element-wise). HolE has been show to be equivalent to ComplEx [30], and therefore, we expect the performance to be similar.

A.3. Geometric models

TransE. The translational model has the scoring function [11]

$$S_{TransE}(s,p,o) = \|e_s + e_p \quad e_o\|_n. \tag{23}$$

Such that if (s,p,o) exists in the KG the relational embedding will translate the subject embedding close to the object embedding.

RotatE. This model is inspired by Euler's identity ($e^{iq} = \cos(q) + i\sin(q)$) and scores triples by rotating the relation embedding in complex space. RotatE has been shown to be ef cient of modelling symmetric, inverse and composite relations [69]. The scoring function of RotatE is de ned as

$$\begin{aligned} S_{RotatE}(s,p,o) &= \|e_s \quad e_p \quad e_o\|_n \\ &= \|e_s \quad (\cos(q_p) + i\sin(q_p)) \quad e_o\|_p \\ &= \|[\hat{A}(e_s)\cos(q_p) \quad \acute{A}(e_s)\sin(q_p) \quad \hat{A}(e_o) \\ &\quad ; \hat{A}(e_s)\sin(q_p) + \acute{A}(e_s)\cos(q_p) \quad \acute{A}(e_o)]\|_n. \end{aligned} \tag{24}$$

Here, we concatenate the real and complex parts of $e_s \quad e_p \quad e_o$. The modulus constraint of e_p is set equal to 1 and is therefore not included in the scoring function. See the original publication for details of derivation.

pRotatE. This model is described as a baseline for RotatE enabling comparison when including modulus information in the model versus limiting to phase information only [69]. pRotatE has the scoring function

$$S_{RotatE}(s,p,o) = \|C\sin(\frac{q_s + q_p \quad q_o}{2})\|_n \tag{25}$$

where $q_x = \angle e_x$ (phase of e_x) and C is the modulus constraint on e_s and e_o.

HAKE. The hierarchy-aware model use the modulus and the phase part of the embedding vectors [85]. Such that entities at the same level in the hierarchy is modelled using rotation, i.e., phase, and the entities at different levels are modelled using the distance from the origin, i.e., modulus. Therefore, the scoring function of HAKE is modelled in two parts

$$S_{pRotatE}(s,p,o) = \|\|e_s\| \quad |e_p| \quad |e_o|\|_n \tag{26}$$

$$+ \|\sin(\frac{q_s + q_p \quad q_o}{2})\|_1 \tag{27}$$

where $|\cdot|$ is the modulus of. The authors noted that a mixture bias can be added $\|\|e_s| \quad |e_p| \quad |e_o|\|_n$ to improved performance [85]. We omit these details here.

A.4. Convolutional models

The nal set of models used in this work are convolutional models. We denote convolutions between an image X and lters w is denoted as $X \ast w$. The models also use dense layers, which is denoted by transform matrices, e.g., W, note that this also includes bias, even though we do not explicit state it. Moreover, dropout

layers are used between every convolutional and dense layer.

ConvKB. The scoring function of ConvKB [54] use a single convolutional layer and a single dense layer

$$S_{\text{ConvKB}}(s;p;o) = f(\text{vec}(f([\mathbf{e}_s;\mathbf{e}_p;\mathbf{e}_o] \quad w))\mathbf{W}); \tag{28}$$

where $\text{vec}(x)$ reshapes x to a 1-dimensional vector. w is the convolution filters. $\mathbf{W}$ is the transformation matrix for the output dense layer. ConvKB can easily be extended to use multiple convolution and dense layers.

ConvE. In contrast to ConvKB, ConvE [21] only perform convolution over the subject and predicate *image* (concatenated and reshaped) and multiples the output dense layer with the object vector as such

$$S_{\text{ConvE}}(s;p;o) = f(\text{vec}(f([\overline{\mathbf{e}_s};\overline{\mathbf{e}_p}] \quad w))\mathbf{W})\mathbf{e}_o^T \tag{29}$$

where $\overline{\mathbf{x}}$ reshapes $\mathbf{x}$ into a 2-dimensional *image*. Here, the last dimension of $\mathbf{W}$ is equal to the embedding dimension. This model can also be extended with multiple convolution and dense layers, however, [21] found that this did not yield improved results.

A.5. Loss functions

Work on KGE models usually define loss functions specific to the models. However, as show in [48, 53] the choice of loss function has a huge impact on model performance. In this work we use four loss functions. We experimented with other loss functions, *e.g.*, absolute/square error, however, these did not materialize in improved results.

To optimize a loss function we need to generate negative examples. Under the local closed world assumption we replace the object of each true triple with all entities and sample negative examples from this set [23], *i.e.*, we sample from $fs;p;o^0g \ 2 \ KG;o^0 \ 2 \ E$. This can be expanded to the stochastic local closed world assumption, which corrupt both the subject and the object of true triples (illustrated by Fig. 3 in [5]). The number of negative samples sampled per positive sample is controlled by a hyper-parameter. However, [36] show that the largest possible number is favorable.

Pointwize hinge. The objective of pointwize losses minimize the scores of negative triples and maximize the score of positive triples.

$$L_{H_1} = \sum_{\mathbf{t} 2 X} [g \quad y_{\mathbf{t}} S(\mathbf{t})]_+ \tag{30}$$

where X is the set of positive and negative triples, y is the triple label (1 for false and 1 for true) and $S(\mathbf{t})$ is the score of triple $\mathbf{t}$. g is the margin hyper-parameter. $[x]_+$ is the positive part of x.

Pointwize logistic. In contrast to hinge loss, logistic loss applies a larger non-linear loss to predictions that are further away from the true label.

$$L_{L_1} = \sum_{t 2 X} \log(1 + \exp(\quad y_{\mathbf{t}} S(\mathbf{t})) \tag{31}$$

Pairwise hinge. The objective of pairwise loss functions is to maximize the distance (in score) between a positive and a negative triple.

$$L_{H_2} = \sum_{\mathbf{t}^+ 2 X^+} \sum_{\mathbf{t} \ 2 X} [g + S(\mathbf{x} \ ) \quad S(\mathbf{x}^+)]_+ \tag{32}$$

where X^+ and X are the sets of positive and negative triples, respectively. g is the margin hyper-parameter, which for pairwise hinge loss represents the maximum score discrepancy between a positive and negative score.

Pairwise logistic. Akin to the move from pointwize to pairwize hinge, pairwize logistic maximizes the distance between positive and negative triples, however, in a non-linear way

$$L_{L_2} = \sum_{\mathbf{t}^+ 2 X^+} \sum_{\mathbf{t} \ 2 X} \log(1 + \exp(S(\mathbf{x} \ ) \quad S(\mathbf{x}^+)): \tag{33}$$

A.6. Implementation

We have implemented the KGE models in Keras [17] and the model codes are available at https://github.com/NIVA-Knowledge-Graph/KGE-Keras. This enables us to easily use the KGE models as components in other models as described in Section 6.