UiO : **University of Oslo**

Erik Bryhn Myklebust

Ecotoxicological Effect Prediction using a Tailored Knowledge Graph

Thesis submitted for the degree of Philosophiae Doctor

Department of Informatics
Faculty of Mathematics and Natural Science

Section for Environmental Data Science
Norwegian Institute for Water Research

SIRIUS Centre for Scalable Data Access

UNIVERSITAS OSLOENSIS
MDCCCXI

2022

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2554

ISSN 1501-7710

Print production: Graphics Center, University of Oslo.

To the catcher of this rock.

Abstract

The use of background knowledge in prediction tasks has great potential to enhance predictive ecotoxicology, but presents challenges in the integration of data from a vast array of disparate sources.

This thesis considers the task of integrating large disparate data sources within ecotoxicology into a domain specific knowledge graph. This knowledge graph is then used as background knowledge for a ecotoxicological effect prediction task, a prerequisite for ecological risk assessment where the goal is to assess the health of an ecosystem. Historically, the adverse effects are observed from laboratory experiments. However, due to the enormous number of potential species and chemicals this is impractical both monetarily and ethically. Therefore, *in silico* methods for effect prediction and extrapolation have been proposed. These methods are highly specialised and may only consider single species and small groups of chemicals. In this thesis a higher level approach is taken.

Considering the most important ecological data sources, and by using powerful Semantic Web tools and custom mappings, we aggregate and transform these data sources into a knowledge graph called the Toxicological Effect and Risk Assessment Knowledge Graph (TERA). TERA consists of species hierarchies and traits with links to external sources, chemical classifications and functional hierarchies, and the adverse effect data from the largest database of such data. Tools are developed to aid the use of TERA in the ecotoxicological community.

Knowledge graph embeddings have been used for link prediction in knowledge graphs. We apply common knowledge graph embedding models to TERA which create dense vector representations of entities. Moreover, these representations are used in various models to perform binary (alive/dead) or continuous (effect concentration) predictions. We evaluate multiple data sampling strategies of increasing difficulty. We show that the use of knowledge graph embeddings as background knowledge improve the results. Furthermore, this advantage increases with data sampling difficulty. We analyse the vector representations for patterns that can explain the variation among models.

Finally, we develop simple ways of analysing the knowledge graph to perform explanations of the predictions. These methods include qualitative and quantitative methods. The quantitative methods provide trust in predictions while qualitative methods aid domain experts with decision making.

This thesis provides a framework for integrating knowledge graphs within ecological risk assessment and especially, biological effect prediction. This will aid researchers and policy makers in providing valuable data insights to provide read across, support effect predictions and provide valuable data insights.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo. The research presented here was conducted at the University of Oslo and at the Norwegian Institute for Water Research, under the supervision of Ernesto Jiménez-Ruiz (2018-2022), Jiaoyan Chen (2019-2022), Raoul Wolf (2019-2022), Knut Erik Tollefsen (2019-2022), Grunde Løvoll (2018-2019), Martin Georg Skjæveland (2018-2019) and Martin Giese (2018-2022). This work was supported by the Norwegian Research Council through grant 272414.

The thesis is a collection of three papers. These papers all use a tailored knowledge graph and knowledge graph embeddings to perform different prediction tasks within the domain of ecotoxicology. The papers are proceeded by an introductory chapter presenting the motivation and hypothesis studied in this thesis, a background chapter giving necessary preliminaries for the understanding of the papers, and a state-of-the-art chapter describing existing methods and applications related to the current work. Thereafter, the main contributions of the papers are presented. The implications to the ecotoxicological, semantic web communities, and machine learning communities are discussed. Finally, the concluding remarks puts the work into a grander context and dwells on further improvements.

Acknowledgements. I would like to thank my supervisors for their guidance and support. Moreover, I would like to raise special gratitude to Ernesto Jiménez-Ruiz, without you this thesis would not have been possible. My employer, the Norwegian Institute for Water Research (NIVA), during the writing of this work was at most supportive and facilitated the work in the best way possible. I would like to thank the other PhD students (and others) at NIVA for the endless coffee breaks and quarantine quizzes.

I would also like to thank the SIRIUS center at the University of Oslo for including me in many scientific and extracurricular activities, and for contributing to student well being overall. I was lucky enough to be able to attend the SIRIUS mentoring program in 2019/2020. This program gave substantial insight into industrial problems and enabled large personal growth and for that I am grateful.

During a research visit in the spring of 2019, the Department of Computer Science at the University of Oxford was outstanding in their hospitality.

I would also like to thank my family for their support even though they have no clue what I've been doing for all these years.

And finally, thanks to all the fish who sacrificed their lives in the name of science!

Erik Bryhn Myklebust, Oslo, September 2022

List of Papers

Paper I

Erik B. Myklebust, Ernesto Jiménez-Ruiz, Jiaoyan Chen, Raoul Wolf, Knut Erik Tollefsen, “Knowledge Graph Embedding for Ecotoxicological Effect Prediction”. In: Ghidini C. et al. (eds) The Semantic Web – ISWC 2019. ISWC 2019. Lecture Notes in Computer Science, vol 11779. Springer, Cham. DOI: 10.1007/978-3-030-30796-7_30.

Paper II

Erik B. Myklebust, Ernesto Jiménez-Ruiz, Jiaoyan Chen, Raoul Wolf, Knut Erik Tollefsen, “Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings”. Semantic Web 13 (2022) 299–338. DOI: 10.3233/SW-222804

Paper III

Erik B. Myklebust, Ernesto Jiménez-Ruiz, Jiaoyan Chen, Raoul Wolf, Knut Erik Tollefsen, “Understanding Adverse Biological Effect Predictions using Knowledge Graphs”. Under review.

Contents

List of Figures

List of Tables

Glossary

AI

artificial intelligence. 1, 42, 43

API

application programming interface. 44, 157–161

CWA

closed-world assumption. 7

DL

deep learning. 22

ECOTOX

the ECOTOXicological Knowledgebase. 26, 27, 37

EOL

Encyclopedia of Life. 32

ERA

environmental risk assessment. xiii, 1, 2, 5–7, 26, 31, 37, 40, 41, 44

GCN

graph convolutional network. xiii, 2, 15, 18, 19, 22, 24, 39

HTML

HyperText Markup Language. 8

IRI

International Resource Identifier. xiii, 8, 9

KG

knowledge graph. xiv, 1–4, 7, 10–27, 31–33, 35–43, 155–161

KGE

knowledge graph embedding. xiv, 2–4, 15–18, 20, 21, 23, 24, 31–33, 35–37, 39–41

KGEM

knowledge graph embedding model. xvii, 2–4, 7, 11–13, 15, 20–22, 24, 25, 31–44, 161

LOD

Linked Open Data. 2, 37, 38, 40

MeSH

Medical Subject Headings. 27

ML

machine learning. 1, 21, 22, 29, 30, 35, 37, 40, 43

MLP

multilayer perceptron. xvii, 34

MoA

mode-of-action. 6, 7

NIVA

the Norwegian Institute for Water Research. 5

NLP

natural language processing. 17, 20, 24, 36

OM

ontology matching. 32, 38, 40, 43

OWA

open-world assumption. 7, 13

OWL

The Web Ontology Language. 8, 10

PCA

Principal Component Analysis. 33

QSAR

quantitative structure-activity relationship. 4, 6, 7, 29, 30, 36, 37, 41

R-GCN

relational graph convolutional network. 18

RDF

Resource Description Framework. xiii, 7–10, 13, 18, 21, 155, 157, 160, 161

RDFS

Resource Description Framework Schema. 8, 10

SAR

structure-activity relationships. 29, 30

SPARQL

SPARQL Protocol and RDF Query Language. 8, 157–160

SVM

support-vector machine. 11, 33, 39, 41

SW

Semantic Web. xiii, 7, 9, 13, 35, 38, 40, 43, 44

SWT

Semantic Web Technologies. 1, 2, 4, 7, 8, 37, 38, 43

TERA

The Toxicological and Risk Assessment Knowledge Graph. xvii, 3, 4, 31–44, 155, 157–161

UMLS

Unified Medical Language System. 27

URI

Uniform Resource Identifier. xiii, 7–10

URL

Uniform Resource Locator. 7

XAI

explainable artificial intelligence. 21

XML

Extensible Markup Language. xiii, 9

Chapter 1

Introduction

There is great potential in background knowledge to inform prediction tasks. Relevant data sources might be disparate and require extensive work to unify. The work in this thesis unified several data sources relevant to ecotoxicology and used these as background knowledge to solve a prediction task; namely, biological effect prediction.[1]

Environmental risk assessment (ERA) is an important tool for understanding and reducing the risks of chemical pollution. It is concerned with assessing the risk of substances on plants, fungi, and animals in an ecosystem. These individual assessments requires large amounts of data, including abiotic environmental samples, but most crucially, biological effect data, *i.e.*, how substances influence organisms. This data is time consuming and costly to produce; furthermore, there are ethical implications of performing experiments on test organisms (10-18 million test organism used per year in the European Union; Busquet et al. 2020). Therefore, models have been developed to extrapolate from existing data. These models are usually based on derived chemical properties. This keeps the models simple and interpretable; however, it puts strict limits on the applicability domain. The domain for a typical model is on the order of 10-100 compounds and one species or genus. We investigate the possibility of integrating large data sources to increase the applicability domain of models. Moreover, the methods we present here are domain-agnostic and can be applied to other low-resources domains.

The large disparate data sources tackled in this thesis needs to be unified and using Semantic Web Technologies (SWT) and knowledge graphs (KGs) is an increasingly popular direction for doing so. Automated tools exist for partially tackling the integration; however, this still requires manual curation to increase coverage and exactness.

Artificial intelligence (AI) and the subfield machine learning (ML) are large, fast moving, research fields. However, the majority of research is conducted on benchmark datasets and applications on real data is limited. This thesis provides several real world applications within the biological effect prediction domain based on existing real, noisy data.[2] The use of integrated data sources and SWT in ML models has the potential to improve predictions on these noisy data.

[1] This thesis is concerned with non-human biological effect prediction, *i.e.*, ecotoxicological effect prediction.

[2] Here, the noise comes from variance in laboratory experiments.

1.1 Motivation

The initial idea for the project arose from the need for homogeneous data in ERA. Life sciences in general are dependent on large disparate datasets (Baralis and Fiori 2008) and have not yet invested in unified frameworks for biological data.

In ecotoxicology, a sub-domain of ecology, there is a large amount of these disparate sources. The sources with the largest discrepancy are biological, or taxonomic, data. Here, domain experts are not unified in the description of *e.g.*, species, which has the consequential error that experimental data is mislabeled.[3] We aim to lessen this problem through the unification of data sources. Thereafter, we apply the unified data to a real problem, effect prediction using knowledge graph embeddings (KGEs).

Knowledge graph embedding models (KGEMs) are plentiful, (relatively) simple, and flexible in implementation and unlike other methods (*e.g.*, graph convolutional networks (GCNs)), their computational complexity scale linearly with dataset size (Yuan, N. Gao, and Xiang 2019). In addition, the learned embeddings from a KGEM can be used for explanation of predictions, either alone or along side the KG. Finally, the use of KGEMs will give insights into how these models behave on data other than benchmark datasets. These are motivations in themselves, but with positive results from the prediction modelling, we can, to some degree, mitigate large experimental expenses and reduce the use of test organisms.

1.2 Hypothesis and Objectives

Based on the motivation we formulate a hypothesis that encapsulates the objectives of this thesis.

The integration of disparate data sources in ecotoxicological research will aid data access and a prediction task, namely biological effect prediction, through the use of knowledge graph embedding models.

To validate the hypothesis we define partial objectives:

Objective 1. *Identify relevant sources within the domain.* Not all sources used in current ecological risk assessment pipelines may be relevant in the described use case. Moreover, the use of SWT and Linked Open Data (LOD) effectively unbounds the data sources available which is not practical in this use case.

Objective 2. *Integrate the sources, including creating or gathering mappings between them.* The use of existing mappings is preferred; however, these are not always available and mapping tools need to be applied.

[3] Not from the experimenter perspective, but from a unification perspective.

Objective 3. *Create several validation strategies for the prediction task.* This is often an overlooked task and results are frequently skewed by improper validation strategies.

Objective 4. *Apply KGE within the prediction task to identify suited KGEMs.* The use of KGEMs has exploded recently; however, the applications tested with these models are fairly limited and exploring an extensive catalog of models is important for the validation of this work.

Objective 5. *Develop a novel prediction model based on the idea of fine-tuning embeddings.* In contrast to (most) other work with KGEMs, the prediction task presented in this thesis is an out-of-the-KG task (see Section 3.2.3); therefore, novel task-specific tuning methods can be explored.

Objective 6. *Use the KG (and embeddings) properties to provide quantitative and qualitative explanations for predictions.* As ecotoxicological predictions are usually made using few pieces of data (Chary, Boyer, and Burns 2021); therefore, these methods are necessary to increase confidence in the methods. In cases where models are both black boxes and uncertain, methods to interpret these models need to be developed. The use of KGs to aid in this process is an emerging research field.

These objectives are covered across Papers I to III. Paper I cover Objectives 1, 2, and 4, Paper II builds on this and covers Objectives 3 and 5, and extends on Objectives 1 and 2 in addition to a vast validation of Objective 4. Paper III covers Objective 6 and expands on Objective 3 with a modified prediction task.

1.3 Contributions

The scientific contributions of this thesis are as follows:

1. The Toxicological and Risk Assessment Knowledge Graph (TERA) integrates the highly valued data sources in the ecotoxicological domain. This integration required a large amount of manual annotation and transformation. Moreover, the use of state-of-the-art alignment tools and external sources was essential. This KG serves as the backbone of the thesis and is essential for the consequent contributions.

2. Popular KGEMs are evaluated in the ecotoxicological effect prediction ranking task with four sampling strategies representing different unknown aspects of the prediction.

3. A fine-tuning method is developed to work in conjunction with the standard KGEMs to tailor the KGEs to the prediction task.

4. The prediction task is expanded from ranking (binary) to the prediction of effect concentrations which proves the ability of background knowledge to contribute to the expansion of application domain of effect predicting models.

5. Methods to gain insight into the predictions are created based on both the symbolic and the latent representation of TERA. These methods can be interpreted by domain experts (during development) or end users of the prediction models.

1.4 Structure of the Thesis

This thesis is a collection of three papers which commonalities are the use of a tailored KG, KGE, and prediction models to perform ecotoxicological effect prediction. The remainder of this thesis is organised as follows:

Chapter 1 introduces the problem this thesis aims to resolve. The motivation, hypothesis, objectives, and contributions are described.

Chapter 2 contains necessary background information to aid in the understanding of the subsequent chapters. It introduces the aspects of environmental risk assessment and the key concepts of adverse effects. SWTs are introduced, focusing on the management of ontology-enriched KGs. Thereafter, prediction models used in the thesis are introduced followed by a high level introduction to KGEs (detailed in Section 3.2).

Chapter 3 gives an overview of existing work in the field of *domain knowledge graphs* and investigates the varying domains where knowledge graphs have been used. Furthermore, KGEMs and their applications are described. Moreover, the two prevalent techniques (quantitative structure-activity relationship (QSAR) models and read-across) for ecotoxicological effect predictions are described and some prevalent work is highlighted.

Chapter 4 summarizes the most important findings of the papers. A timeline of the evolution of content and methods through the papers is also presented.

Chapter 5 contains a discussion related to the presented material in Chapter 3. The limitations and implications of the research performed over the three papers are discussed.

Chapter 6 presents the concluding remarks and the potential future improvements which can address the limitations of the presented work.

Chapter 2

Preliminaries

This chapter introduces concepts to facilitate the understanding of the consequent chapters.

2.1 Ecological Risk Assessment

Authorities demand ERAs to prove that no harm is expected by a substance if applied in a environmental context. Therefore, a methodology for performing good and consistent risk assessments are needed. Figure 2.1 shows a simplified version of a risk assessment pipeline, based on the one used at the Norwegian Institute for Water Research (NIVA) (Tollefsen 2018). This pipeline is used to assess whether a given ecosystem will be influence by *e.g.*, runoff from nearby farmland. This is done through several components:

Exposure is the chemicals quantities detected in the environment, through *e.g.*, water samples.

Effects are results from laboratory experiments with reference species and chemicals. See the next section.

Risk Quotient is the risk excerpted on single species, or taxa, by single chemicals, *i.e.*, the ratio between exposure and effect.

Hot spots identification are where the thresholds of effects and safe exposures in terms of adverse effects of regulatory relevance is documented.

Susceptibility of a species is the fact of being likely or liable to be influenced or harmed by a stressor, *e.g.*, a chemical.

Risk Driver is a chemical with the highest risk quotient, *i.e.*, where exposure exceeds the safety thresholds with the largest magnitude.

Finally, a simplistic risk assessment of the total ecosystem is the accumulation of individual risk quotients. Moreover, more complex risk assessment might take other factors into account such as importance of a species in the ecosystem, *e.g.*, in the food chain.

2.2 Adverse Effects and Prediction

As seen in the previous section, effect data[1] is highly important for ERA with large coverage in species or chemicals (Larras et al. 2022; Rohr, Salice, and

[1] *Effects* is used in short for both adverse and chemical effect throughout the thesis.

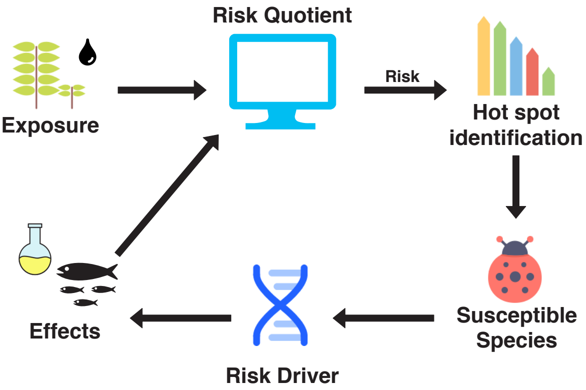


Figure 2.1: A simplified ERA pipeline. **Exposure** is the level of chemicals detected; **Effects** are the measures of adverse responses in an organism due to exposure to one or more chemicals; **Risk Quotient** is the ratio between **Exposure** and **Effects**; **Hot Spots** are identified by where the **Risk Quotient** is largest; **Susceptible Species** are those which are most at risk for a particular compound; **Risk Drivers** are chemical with the highest risk quotient (above safety thresholds). Permission to reuse under the Creative Commons Attribution License (CC BY 4.0). Published by IOS Press. Myklebust, Jiménez-Ruiz, et al. 2022.

Nisbet 2016). These effects can be related to mortality, fertility or reduction in growth etc. (Van Leeuwen 1995). In this work, we are most concerned with mortality as it is binary[2], which makes prediction simpler and reduces ambiguity. However, the effects are more abstract as laboratory experiments have endpoints as results. These endpoints describe the experimental results, *e.g.*, the LC_{50} is the lethal concentration (LC) to half (50%) of the population (Forfait-Dubuc et al. 2012). This method will account for discrepancies between organisms as long as the population is large enough (Liess et al. 2016).

To create sufficient coverage in the biological effects a high number of animal experiments are often required. However, this is monetary and ethically challenging and, therefore, *in silico* methods have been developed to aid in extrapolation of existing data. An example of these are QSAR models (presented further in Section 3.3; *c.f.*, Tsakovska, Diukendjieva, and Worth 2022). These models describe the mode-of-action (MoA)[3] or adverse effect of an observable toxicity in a taxon, species or species group (Grant, Combs, and Acosta 2010). They vary in complexity from linear models with few features, *e.g.*, octanol-water partition coefficient (logP), to fingerprinting of chemicals and convolutional models (Breiman 2001; Cover and Hart 1967; Fujita and Winkler 2016; Geppert

[2] For the individual, not the population.

[3] Functional or anatomical change in organism due to chemical exposure.

et al. 2008; Hansch et al. 1962; Sakai et al. 2021). Other approaches, such as clustering, are also popular with ecotoxicological modelling (Hasan et al. 2019); however, they are not considered in this thesis as we focus on supervised methods.

The largest drawback of these QSAR methods is the applicability domain. The applicability domain concerns the range of physcio-chemical properties, which effect or biological responses and which species (groups) the prediction has been designed for. The applicability domain is usually small and limited to one or a few species and ten to hundreds of chemicals, much smaller than is needed to cover the large combination of chemical-species pairs necessary for ERA (Weaver and Gleeson 2008). Lastly, a QSAR model can only be adapted to single (or limited groups) of MoAs and adverse effects, meaning that a comprehensive risk assessment would require multiple QSAR models to produce good results. Moreover, the reliance on *ad-hoc* feature selection slows down the required development further.

2.3 Semantic Web Technologies

Large amounts of metadata are available on the Web. However, these data are not necessarily inter-operable. SWT aim at integrating these vast disparate data sources into unified frameworks. Loosely from Greek, semantic, translates to meaning. This emphasizes that each piece of information should be unique and identifiable.[4] The full stack of SWT is shown in Figure 2.2. A few parts of the stack are critical for the thesis and are introduced below.

Foundationally, the idea is to identify *things* (animate and inanimate objects, ideas, concepts, actions, *anything*) as unique entities instead of representing them by ambiguous strings. The Resource Description Framework (RDF) is the preferred framework to describe structured data in the Semantic Web (SW). RDF is used to describe graphs of data, *i.e.*, a set of entities (or resources) connected by labeled directed edges. The definition of an RDF graph (or knowledge graph) will follow below.

In contrast to many other datastores, RDF is based on the open-world assumption (OWA). In simple terms, in the OWA a statement might be true irrespective of the known truth of the statement. Contrastly, the closed-world assumption (CWA) assumes all knowledge is known; therefore, any statement not known (or not in a database) is false. Moreover, as a consequence of OWA, we cannot be sure all information is present in a KG. As a side note, KGEMs assume a closed world, this is necessary for the creation of false facts during training, *i.e.*, false statements created by modifying parts of a true statement.

As mentioned, the entities in RDF are defined as unique things, using Uniform Resource Identifiers (URIs), a generalization of Uniform Resource Locators (URLs). A URI has a number of building blocks (Hitzler, Krötzsch, and Rudolph 2009, p. 23):

[4] Note that, we will limit this section to technologies used in the thesis, SWT is a vast field of research and is too large to cover here.

scheme:[//authority]path[?query][#fragment]

where the parts of the URI express:

scheme is the type of the URI, *e.g.*, `http`. In applications, schemes can be used to indicate the use of the URI, *e.g.*, `rdf`.

authority equivalent to *domain* on the Web and is optional, *e.g.*, `example.org`.

path is the core of the URI. These typically contain hierarchies of information separated by /, *e.g.*,/path/to/important/information.

query is optional and can provide non-hierarchical information not contained in path. On the Web, a query is used to provide parameters.

fragment is used to provide a sub-part of the resource. *e.g.*, pointing to a section in a HyperText Markup Language (HTML) file.

Note that, not all characters are allowed in URIs. Latin letters and numbers are allowed in most positions while the extension to International Resource Identifiers (IRIs) allows the use of language specific characters.

In addition to resources described by URIs, RDF can also include data values, called literals. These values are strings which can be associated with a value type tag, *e.g.*, `"fish"@en` and `"fisk"@no` are both labels for the entity `Fish` in English and Norwegian, respectively.

In addition to RDF we use several other SWT in this work:

RDFS. For now we have only considered entities in RDF. Resource Description Framework Schema (RDFS) enables the definition of classes which contain entities. Class assertions are defined though `rdf:type` edges in the graph. It follows that subsumption, property domain/range can be defined in RDFS.

OWL (The Web Ontology Language) is the *de facto* language for defining ontologies in RDF. This enables the definition of logical constraints/requirements on the data, *e.g.*, Disjointness.

SPARQL (SPARQL Protocol and RDF Query Language) is the query language to query RDF data. The queries are based on graph patterns which are matched with the target graph.

Blank nodes are nodes in RDF without a URI. This enables simple modeling of many-valued relationships, *e.g.*, Value-unit pairs. See Figure 3 in Paper II.

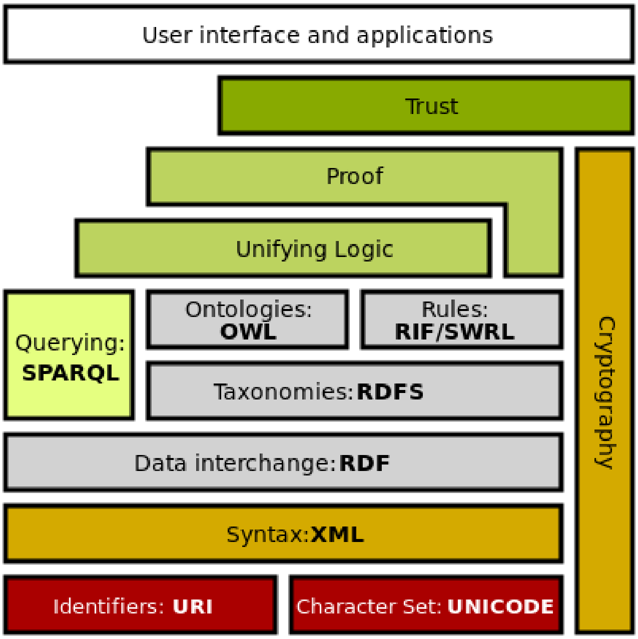


Figure 2.2: The building blocks of the SW. The foundation is made up of *Unicode*, the standard character set, *URI/IRI* identifiers, Extensible Markup Language (XML), and *Namespaces* which defined URI scopes. XML Query and *Schema* form the basis of the RDF language which can express more complex knowledge such as *Ontologies*. These can be enriched with *Rules*, *Logic*, and *Proof* (*e.g.*, source). The entire stack can be *signed* and *encrypted* to provide the *Trust* needed. Reuse permitted under the Creative Commons CC0 1.0 Universal Public Domain Dedication. Inspired by Obitko 2007.

2.4 Ontology Alignment

Ideally, all data sources should use the same schema or terminology when talking about the same or interlinking domains; however, this is not the case. Therefore, methods for aligning sources of different schemas are necessary. This is the problem of ontology matching or alignment. This aims at using the structure and literals of an ontology or RDF graph to match resources (classes, instances, relationships) from one schema to another.

This is useful when dealing with incomplete data, as in our use case for ontology alignment, matching a limited taxonomy to a complete taxonomy[5]

Many tools exist for performing ontology alignment, *e.g.*, Faria, Pesquita, et al. 2013; Jiménez-Ruiz and Cuenca Grau 2011; Jiménez-Ruiz, Cuenca Grau, Zhou, et al. 2012. The Ontology Alignment Evaluation Initiative (OAEI) (Abd

[5] In this work, the NCBI Taxonomy is assumed to be complete and the *de facto* organisation of species.

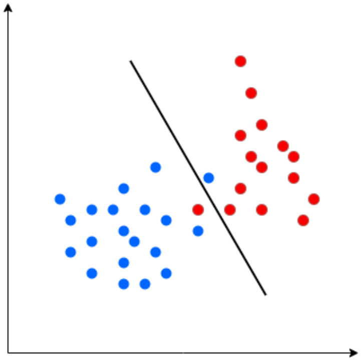


Figure 2.3: Supervised learning. Based on labeled data (blue and red), the boundary is chosen to minimize errors.

Nikooie Pour et al. 2020; Algergawy, Cheatham, et al. 2018; Algergawy, Faria, et al. 2019) is an yearly event where ontology alignment tools are evaluated on real-world tasks.

2.5 Ontology-enhanced Knowledge Graph

The use of KGs have been popularized over the last years (Hogan et al. 2020; Kroetsch and Weikum 2016; Paulheim 2017). Google is credited with coining the term which was used to describe FreeBase (Bollacker et al. 2008a) when it was acquired. This definition of a KG consists of triples on the form $\langle s, p, o \rangle$, where s, p, o is the subject, predicate, and object of the triple (*e.g.*, $\langle \texttt{Salmon}, \texttt{endemicTo}, \texttt{AtlanticOcean} \rangle$). No formal restrictions is put on s, p, or o in this definition. However, in this work we require $\langle s, p, o \rangle$ to be an RDF triple, where the subject is an instance or a class, the predicate is a property, and the object an instance, a class, or a literal (string, numbers, etc.). [6] In addition, the KG entities (classes, instances, properties) are described by URIs.

An ontology-enhanced KG (Arenas et al. 2014) can be split into an ABox (assertions) and TBox (terminology). The TBox contains triples using RDFS concerning, *e.g.*, class subsumption, property domain/range; and OWL constructors such as disjointness, equivalence, and property inverse. The ABox contains assertions among instances, including OWL equality and inequality, and semantic type definitions.

2.6 Supervised Learning

Machine learning can be divided into two main categories, unsupervised and supervised. In supervised learning prediction targets corresponding to the input

[6] $\mathcal{E}$ is the set of all classes and entities and $\mathcal{R}$ is the set of all relations.

are available while in unsupervised learning they are not. The prediction models in this work are strictly supervised models. In supervised learning, a mapping from an input to an output is learned from examples as seen in Figure 2.3. The creation of this mapping is done by a model, these can vary from simple linear models to highly non-linear neural networks. More complex models are able to capture more information from the input to use in the prediction; however, the drawback of this can be overfitting, *i.e.*, the model learns individual input-output pairs and does not generalize well.

The supervised learning methods used in Papers I to III are

Neural networks consist of a series of linear transforms are applied to the input. A non-linear function can be applied to the transforms to make them non-linear, *e.g.*, ReLU ($f(x) = \max(0, x)$). A loss function is calculated based on how *wrong* the predictions are. Thereafter, corrections to each transform is calculated and applied thought a process called back-propagation (Rumelhart and McClelland 1987). This process is iterated until the updates become minuscule and the loss function has reached a local minima.

SVMs (support-vector machines) are normally used in classification problems where the goal is to maximize the distance between the decision boundary (class hyperplane) and the individual points. Regression can be performed using a SVM by considering the margin from the decision boundary as a real number and not a strict boundary. SVMs are strictly linear; however, applying a (non-linear) kernel can be done to make SVMs non-linear, *i.e.*, the kernel projects the input non-linearly into a space where the problem becomes linear (and smooth; Cortes and Vapnik 1995; Drucker et al. 1996). In this work, we use a radial basis function kernel defined as $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$ where $\gamma = 1/2\sigma$, and $\mathbf{x}$ and $\mathbf{x}'$ are two different datapoints.

In addition to these two prediction models we use KGEMs to represent the KGs in vector space.

2.7 Knowledge Graph Embedding Models

Symbolic data, such as KGs are not easy to use directly in machine learning and, therefore, embedding models have been developed whose task is to create vector representations of entities and relations in the KG while maintaining its structure.[7] The KGEM used in this thesis have their origin in the task of link prediction, *i.e.*, assuming the KG is incomplete and predicting the missing facts. Models such as RDF2Vec (Ristoski and Paulheim 2016) perform walks over the KG to create sentences. Thereafter, *e.g.*, Word2Vec (Mikolov et al. 2013a) is

[7] Structure of embeddings is subjective and task specific, and will vary greatly from model to model.

used to create vector representations from the sentences. Chapter 3.2 will cover these and other methods in detail.

The KGEM used in Papers I to III assigns a score to each triple ($S(t)$) in the KG which is proportional to the probability (P) that the triple exists in the KG, *i.e.*,

$$S(t) \propto P(t \in KG), \tag{2.1}$$

The models are trained in a supervised manner using score and loss functions shown in Appendix II.A in Paper II. The KG only consists of positive facts (where probability equals 1). Moreover, supervision cannot be applied in such a case; therefore, the models need to create negative triples. This is done by corrupting positive triples by replacing the subject and/or object in a positive triples by a random entity from the KG. The ratio of negative to positive triples is usually $\gg 1$. This is done to *fill* the latent space around the positive triples.

Chapter 3

State of the art

This chapter will cover work related to this thesis. Firstly, domain KGs are introduced. Thereafter, KGEMs and applications are discussed. Finally, ecotoxicological effect prediction methods from the literature are presented.

Each section will include a postamble regarding the relevance of the presented work to the thesis work.

3.1 Domain Applications of Knowledge Graphs

A crucial component of the work in this thesis revolves around a domain-specific KG. Therefore, this chapter will cover prominent domain-specific KG applications.

3.1.1 Domain-Specific Knowledge Graphs

To solve domain problems, in this context, there is a need for domain-specific KGs. In other words, generic KGs are not enough to solve specific problems.

There is no consensus on the definition of a KG; but, Section 2.5 defined the (generic) KG based on RDF that is used throughout this thesis. This definition is based on the OWA, are domain-independent and is part of the foundation of the SW. Furthermore, domain-specific KGs also have no concise definition. One can think of them as vehicles for the definition of complex domains (Fan et al. 2017; Li et al. 2020; Yuan, Jin, et al. 2020) or the process of enriching a domain ontology (Kejriwal 2019). Abu-Salih 2021 provides a inclusive definition of the term *domain KG*:

> Domain Knowledge Graph is an explicit conceptualisation to a high-level subject-matter domain and its specific subdomains represented in terms of semantically interrelated entities and relations.

This definition includes important aspects. Conceptualisation through a predefined ontology which can capture generic or specific domain of interest. The definition also ensures the subject-matter is firmly contextualized to address the specific knowledge. Finally, it depicts the domain KG as a labeled graph with entities and relations between these. This last property is largely connected to the definition of a generic KG introduced previously.

3.1.2 Knowledge Graph Applications

This section will present the use of KGs in wast areas of science.

There are far more KG applications that are not covered here, and therefore survey papers are a valuable resource. A few domain specific surveys include:

- Amin and Bhattacharyya 2019 presents the vast use of KGs in the biomedical domain;
- Q. Guo et al. 2020 presents the extensive use of KGs in recommender systems;
- Sani 2020 has compiled a survey on cybersecurity applications of KGs;
- H. L. Nguyen, Vu, and Jung 2020 looks into KG fusion and the use in smart systems.

We will present a few detailed examples from these surveys and other applications described in them. Firstly, this section will focus on chemistry, biology, computer science, and other technical subjects as these are mostly relevant to this thesis. Other applications of KGs will be touched upon towards the end.

Chemistry. Chemical modelling is a large area of research and Farazi et al. 2020 has developed the ontology OntoKin and then populated a KG with content based on the ontology using linked open data. The usefulness of the KG is demonstrated on a query use-case to identify variations between 10 different mechanisms related to a specific reaction.

Another domain where the data is complex and plentiful is in the petroleum industry. Here, the major concern is retrieval of relevant information. Huang, Y. Wang, and Yu 2020 have therefore integrated a KG with other structured and semi-structured data to improve intelligent search. This enables the user to easier express intent of the search in such a complex domain. Moreover, the integration of semantics makes results easier to understand, enhancing the quality of the search.

Computer Science. Kiesling et al. 2019 designed the evolving SEPSES KG to detect vulnerabilities in computer systems. The KG can be linked to local sources of data and evolves as new information becomes available. The authors demonstrate the use of the KG in two use-cases: assessing vulnerability and detecting intrusions.

Fu et al. 2019; Nayak, Kesri, and Dubey 2020 both construct KGs related to software development. The former relates documentation and requirements to extract test cases from the KG. The latter relates to development rather than testing and has integrated data from Stack Overflow[1] and Wikipedia[2], in addition to crowdsourcing. Using this KG in a ranking model improves metrics by up to 35% (Fu et al. 2019).

Anomaly detection is important in many domains. In Aumayr, M. Wang, and Bosneag 2019 the authors construct a KG which reduce complex telecom incident reporting. This KG is constructed using natural language processing tools. In a use-case, the KG is shown to provide information dense reporting and improve

[1] https://stackoverflow.com/

[2] https://www.wikipedia.org/

effectiveness over natural language reports. Another form of anomaly is fraud detection. Zhan and Yin 2018 enhances a neural network approach of predicting fraud in loan applications using a KG. The KG is generated from transaction data before *word2vec* is used to generate vector representations for the fraud prediction model. Albeit, fraud methods change over time and, therefore, automated methods for extracting knowledge is necessary (Zhan and Yin 2018). The authors demonstrate their method on a real use-case with good results. Unfortunately, this KG's usefulness is limited due to the quality. This is a product of the challenging data in this application (Zhan and Yin 2018).

Food. KGs are also being used for nutritional purposes. FoodKG (Haussmann et al. 2019) is comprised of recipes and other food related information. FoodKG tries to solve the specificity of other food related ontologies which are usually related to a sub-domain, *e.g.*, production. The KG helps consumers find healthy recipes along with the source information. FoodKG enables users to ask for recipes using the ingredients at hand at not only finding ingredients based on recipes. To further usability, tools have been developed which enables question answering over FoodKG.

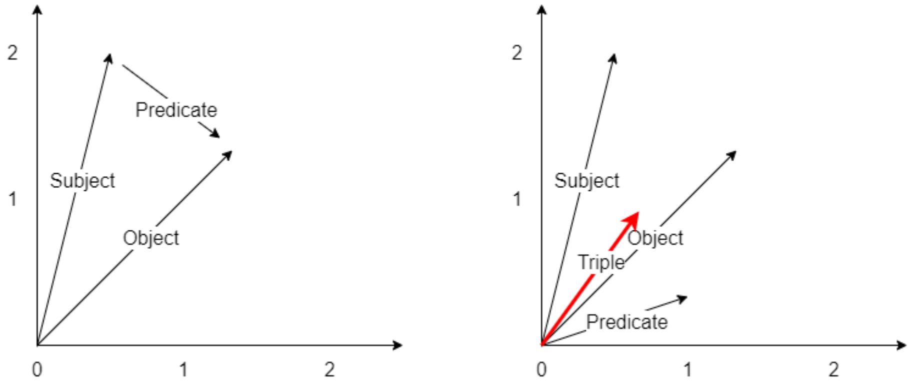
Products. Large retailers/auction-house, like eBay, use KGs to organise products (Noy et al. 2019). These KGs are automatically extracted to create hierarchies of products. The main product KG contain around 100 million products and 1 billion triples. The use cases of this are plentiful. One can link entities in the KG to real world items and make recommendations, *e.g.*, a customer interested in *Lionel Messi* memorability might also be interested in *Argentina* memorability. This link can be made based on the KG.

This section has introduced several domain-specific KGs and these provide a wast array of different datasets and approaches to creating KGs. The KG introduced in this thesis has plenty in common regarding construction and use-case as presented work, *e.g.*, biomedicine (Amin and Bhattacharyya 2019), anomaly detection (Aumayr, M. Wang, and Bosneag 2019), or chemistry (Farazi et al. 2020). Furthermore, the areas of software testing (Nayak, Kesri, and Dubey 2020), recommender systems (Q. Guo et al. 2020), or product definitions (Noy et al. 2019), influenced the design decisions of the KG constructed in this thesis.

3.2 Knowledge Graph Embeddings

This section will cover the KGEMs used across Papers I to III. In addition to these, alternative methods such as sentence-based methods and GCNs are introduced as they can be alternatives to KGEMs. Section 5.3 discusses why KGEMs was rather used. Finally, applications of KGEs within and outside KGs are described.

Over the three papers included in this thesis we use a selection of nine KGEM. All models follow the same basic formulation as described in Section 2.7 and further details are provided in Appendix II.A.



(a) Translational KGE (TransE).

(b) Distance multiplication (DistMult).

Figure 3.1: Visual representation of TransE (Bordes, Usunier, et al. 2013a) and DistMult (B. Yang et al. 2015). (a): The score for a triple is the distance between *Subject* + *Predicate* and *Object*. (b): Vector *Triple* is the result for multiplying *Subject*, *Predicate*, and *Object*. The score of the triple is then the sum of the elements for the vector.

These nine models are classified into three categories: geometric, decomposition, and convolutional.

The geometric models include TransE (Bordes, Usunier, et al. 2013a), HAKE (Z. Zhang et al. 2019a), and RotatE and its variation pRotatE (Sun et al. 2019a). TransE is the base of a whole family of models using various translations from subject to object in a triple (*e.g.*, via hyperplanes; Z. Wang et al. 2014). TransE is the simplest of these modelling the translation as a straight line, *i.e.*, $\mathbf{e}_s + \mathbf{e}_p \approx \mathbf{e}_o$, where $\mathbf{e}_x \in \mathbb{R}^k$. This is seen in Figure 3.1a.

The hierarchical aware model, HAKE, as the name suggests, takes the hierarchical structure into account, which is very relevant for our particular use case and KG. This is done by considering each level in the hierarchy at a fixed modulus from the (embedding space) origin and each entity at a level as a phase rotation. For long property chains (*e.g.*, hierarchies in the form of `subClassOf` chains) this will create a radially structured embeddings with each level in the chain at a set radius from the origin while other properties (*e.g.*, `age`) will create small embedding sub-hierarchies spanning (sideways) from entities in the main hierarchies.

The use of a rotation is also central in RotatE, where the relation is rotated in complex space using the Euler identity, by representing the relation as $\cos(\theta_p) + i\sin(\theta_p)$. This enables strong performance on symmetric, inverse, and composite relations. pRotatE removes the modulus information from RotatE. This was done by the authors in Sun et al. 2019a to create a simpler baseline.

The decomposition models used in the papers of this thesis are DistMult (B. Yang et al. 2015), ComplEx (Trouillon et al. 2016), and HolE (Nickel, Rosasco, and Poggio 2015). DistMult uses a simple inner-product to model the triple probability, *i.e.*, $\langle \mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \rangle$ as seen in Figure 3.1b. ComplEx uses the same score function; but, with complex vector representations. This enables ComplEx to model inverse relationships uniquely, *i.e.*, $\langle \mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \rangle \not\equiv \langle \mathbf{e}_o, \mathbf{e}_p, \mathbf{e}_s \rangle$.

Holographic embeddings, or HolE, use circular correlation between the vector representations of the subject and object which is then multiplied with the representation of the relations to model the score. Circular correlation can be calculated in frequency space as multiplication of conjugated vectors, which makes the calculation relatively simple.

Finally, the convolutional models applies deep learning to the problem of KGE. ConvE (Dettmers et al. 2018) and ConvKB (D. Q. Nguyen et al. 2018) use the same convolutional operation with slightly different architectures. ConvKB applies convolution to the concatenation of subject, predicate and object in a triple. ConvE only applies convolution to the concatenations of subject and predicate and lets the object vector be used in an inner-product with the convolutional output.

These models represent just a small selection of the models that have been developed in the last years. However, they are representative and are often used as baselines against which new models are compared.

3.2.1 Sentence-Based methods

This class of models takes inspiration from natural language processing (NLP). However, these methods require data in the form of natural language, *i.e.*, sentences. Disregarding the downstream NLP model, each method has unique ways of extracting sentences from KGs or ontologies.

Firstly, RDF2Vec (Ristoski and Paulheim 2016) performs random walks over the KG. A random walk is created by starting at one node (entity) in the graph and selecting predicates connected to this entity in a random manner. This is the continued until the walk is of desired length. If a dead end (*e.g.*, leaf node) in the KG is reached during the walk, the method jumps randomly to another part of the KG, and continues from there. The random walks are then used in the word2vec algorithm (Mikolov et al. 2013a) to create entity and predicate embeddings.

node2vec (Grover and Leskovec 2016) is similar to RDF2vec. However, node2vec prioritizes the graph neighbourhoods which is claimed to create richer embeddings. This is done by adjusting the transitional probability between nodes based on neighbourhoods. node2vec is optimized using stochastic gradient decent (SGD) over an objective that preserves the neighbourhoods. Moreover, the neighbourhood biased sentences can also be used in word2vec as in RDF2vec.

These methods have shown great performance on simple graphs and KGs. However, these methods are not able to capture logical constructs which are found in ontologies.

Onto2Vec (Smaili, X. Gao, and Hoehndorf 2018) and OPA2Vec (Smaili, X. Gao, and Hoehndorf 2019) are two methods which create low-dimensional representations of concepts and entities in an ontology. The axioms of the ontology are used to create a corpus in Onto2Vec before word2vec is applied to this. OPA2Vec extends this by including lexical information (*e.g.*, `rdf:label` or `rdf:comment`) in the creation of the corpus. Both methods were validated on the Gene Ontology (GO; Ashburner et al. 2000; Seth Carbon et al. 2020) on the protein-protein interaction prediction task among others (Smaili, X. Gao, and Hoehndorf 2018). As both methods treat the axioms as individual sentences, it is hard to explore large graph structures, *e.g.*, hierarchies which is comprised of long `rdf:subClassOf` chains.

This drawback is addressed by the authors of OWL2Vec (and extention OWL2Vec*; J. Chen, Hu, Jimenez-Ruiz, et al. 2020; Holter et al. 2019) by transforming the ontology into an RDF graph and applying RDF2Vec, *e.g.*, axiom $B \sqsubseteq A$ is projected to RDF triples $\langle \mathtt{B}, \mathtt{rdfs:subClassOf}, \mathtt{A} \rangle$ and $\langle \mathtt{A}, \mathtt{rdfs:subClassOf}^{-}, \mathtt{B} \rangle$ ($\mathtt{p}^{-}$ denotes the inverse of $\mathtt{p}$). These projections enable OWL2Vec to outperform above mentioned methods on membership and subsumption prediction.

3.2.2 Graph Convolution Networks

Kipf and Welling 2017 introduces the notion of GCNs for the task of node classification. This can be considered a semi-supervised problem as labels are only available for select nodes and needs to be propagated across the graph. A graph convolutional layer is described as:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}), \tag{3.1}$$

where $\tilde{A}$ is the adjacency matrix of the graph with self connections, $\tilde{D}$ is a diagonal matrix with the column wise sum of $\tilde{A}$ on the diagonal, W is the layer weights, and σ is the activation function. The input layer $H^{(0)}$ is graph feature matrix X. A number of layers can be stacked to create a model.

Kipf and Welling 2017 demonstrated how a two layer GCN performs on citation network prediction. Baselines are consistently outperformed by GCNs.

For multi-relational KGs, GCNs can be extend to a variation called relational graph convolutional networks (R-GCNs) (Schlichtkrull et al. 2017). In essence, this extension introduces relation specific weights to (3.1) and accumulates the results based on the neighbourhood around each node.

relational graph convolutional network (R-GCN)s can be used for entity classification similar to GCNs or used as a feature encoder in a link prediction task as show in Figure 3.2. Here, a link prediction model (*e.g.*, DistMult) acts as the decoder. Schlichtkrull et al. 2017 show that R-GCNs performs well on entity classification and link prediction, depending on the benchmark datasets used for validation. Extending the methods with more powerful decoders, *e.g.*, ComplEx, might yield state-of-the-art results across a vaster array of datasets.

3.2.3 Applications

This section will describe applications of KGEs. These can be classified into two sub-categories. Either the application is applied within the KG (it acts only on

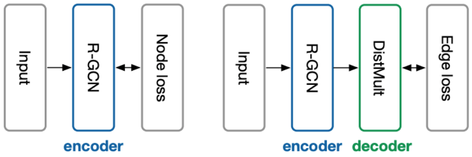


Figure 3.2: Left: Entity classification: each node of the graph is classified. Right: Link prediction: DistMult (B. Yang et al. 2015) is used to predict triple scores (see Equation 2.1) based on the GCN output (Schlichtkrull et al. 2017). Reprinted by permission from Springer Nature: European Semantic Web Conference, Lecture Notes in Computer Science, Schlichtkrull et al. 2017.

elements in the KG) or outside to support other tasks (also called background knowledge).

3.2.3.1 Within-KG Applications

The subsequent sections will describe applications within the KG. This relates to KG refinement tasks, such as completion.

Link Prediction. Link prediction is used to complete KGs, *i.e.*, predict missing facts. The task is to predict the object of a triple given the subject and predicate or predict the subject given the predicate and object. This task has been extensively studied *e.g.*, Bordes, Usunier, et al. 2013a; Lin, Z. Liu, et al. 2015; Nickel, Rosasco, and Poggio 2015; Z. Wang et al. 2014. Link prediction has also been used in biomedical KGs to discover new protein-protein interactions which has aided in the development of new drugs (*c.f.*, S. K. Mohamed, Nováček, and Nounu 2019).

To evaluate models on the link prediction task, the learned models' scoring function is applied to the full set of candidate triples, *i.e.*, $\langle \texttt{Subject}, \texttt{Predicate}, ? \rangle$ where ? is any entity in the KG. These scores are then used together with the ground truth(s).[3] The performance of this task is usually measured in *Hits@k* (normal values for k in the literature are $1, 5, 10$). The metric value is 1 if the real entity is in the top k highest scoring entities, 0 otherwise.

An extension of the link prediction task is relation extraction where the predicate is to be predicted based on a known subject and object.

In the case of link prediction, it is assumed that there exists a fact in the KG with particular subject and predicate. However, this is not necessarily true and this can be resolved with triple classification.

Triple Classifications. In link prediction the assessment is on an entity level, while triple classification assesses the whole triple as one. The creation of prediction

[3] In this context, a ground truth is a triple existing in the KG that has been holdout for validation.

models to assess the correctness of a triple has been extensively studied, *e.g.*, Lin, Z. Liu, et al. 2015; Socher et al. 2013; Z. Wang et al. 2014. This again can be done by applying the learned models' scoring function to candidate triples. The metric used for assessing the performance of a model in this task is most often accuracy (micro and macro), and a threshold needs to be defined to maximize this. The micro-accuracy can be improved by optimizing thresholds per predicate. As the model prediction is a relative probability, ranking metrics (*e.g.*, mean average precision) can also be used (S. Guo et al. 2016).

Entity Classification. Entity classification is a special case of link prediction where the goal is to categorize entities into classes (*e.g.*, `rdf:type` relations). If the KG contains encoded entity types the same methods and metrics as above can be used (Bordes, Usunier, et al. 2013a; Nickel, Tresp, and Kriegel 2012). One the other hand, if entity classification is a separate task (`rdf:type` triples are not present in the KG), it cannot be treated as link prediction. In this case, one can classify entities in other ways, *e.g.*, by clustering the entity embeddings (Gad-Elrab et al. 2020).

Entity Resolutions. During the construction of large KGs (*e.g.*, constructing Wikidata from Wikipedia), there are possibilities for duplicates, *e.g.*, `DEET` and `diethyltoluamide`.[4] One can use KGEs to assess the similarity between entities and de-duplicate KGs (Bordes, Usunier, et al. 2013a; Glorot et al. 2013). Firstly, this can be a special case of triple classification if *e.g.*, `owl:sameAs` relations are present in the KG. This was the case considered in Bordes, Usunier, et al. 2013a where triples $\langle x, \texttt{owl:sameAs}, y\rangle$ are classified to determine whether x and y refer to the same entity.

In the other case, where equivalence axioms are not present in the KG, entity resolution can be assess using the entity vector representation. The similarity between two entities x and y is defined as $\exp(-||\mathbf{e}_x - \mathbf{e}_y||_2^2/\sigma)$, where $\mathbf{e}_x$ is the representation of x and σ is a user defined normalization factor (Nickel, Tresp, and Kriegel 2011).

3.2.3.2 Out-of-KG Applications

This section will describe a few tasks where the KG supports an external prediction task.

Natural Language Tasks. KGs and learned embeddings can help remove ambiguity from natural language statements. A prominent application is relation extraction where, *e.g.*, inflections, can complicate the extraction of knowledge. *e.g.*, in the sentence *Ridley Scott directed Blade Runner* the entities `RidleyScott` and `BladeRunner` are detected, and `DirectorOf` should be the predicted relation. Many studies have tried to take advantage of KGs in this task; however, mostly to automatically generate labeled data (Hoffmann et al. 2011; Jiang et al. 2016; Mintz et al. 2009; Riedel, Yao, and McCallum 2010). Methods have also been proposed to jointly learn a NLP model and a KGEM (aligned to corpus) to

[4] Insect repellent.

improve performance in this task, *e.g.*, Riedel, Yao, McCallum, and Marlin 2013. This way the language and KGE models can inform each other by textual mentions and link prediction (or triple classification).

A related task is question answering, where the task is to give answers to natural language questions that are supported by one or more triples in a KG. KGE can be used in this task by modelling low-dimensional representations of questions and answers and applying a similarity function which scores candidate answers (Bordes, Chopra, and Weston 2014; Bordes, Weston, and Usunier 2014). The KGE are then learned using this similarity function and example questions and answers. Bordes, Chopra, and Weston 2014; Bordes, Weston, and Usunier 2014 show that this method produces promising results without extra lexical information or rules.

Recommender Systems. The goal of a recommender system is to advise the user of potential choices. However, in many situations the latent representation of the user-item interaction can be sparse. A KG can be leveraged to solve these issues. F. Zhang et al. 2016 proposed to use the textual (descriptions), visual (images, *e.g.*, book or movie cover), and structural information in the KG to augment recommendations. Combining the structural knowledge learned using a KGEM, and textual and visual representations learned with autoencoders enables ranking of items that a users prefer over others.

Explainability. Most deep learning systems are black boxes and, therefore, not explainable. KGs, and semantic web technologies in general, can provide semantic explanations for predictions. These explanations are required in high stakes domains, *e.g.*, health care, where doubt in predictions is undesirable.

Explainable artificial intelligence (XAI) is a broad term which incorporates explainable, transparent, interpretable, or comprehensible ML methods. Methods have been proposed to increase the understanding of models by technical analysis of ML methods (*c.f.*, Adadi and Berrada 2018; Gilpin et al. 2019). Cherkassky and Dhar 2015 argue that the use of these methods can never achieve explainability and that explainability is highly dependent on the domain in question. Therefore, domain centric KGs and ontologies could be a large part of truly explainable ML methods (Holzinger, Biemann, et al. 2017; Holzinger, Kieseberg, et al. 2018).

Explaining supervised learning methods with KGs can be done by mapping inputs to entities in the KG. *e.g.*, Sarker et al. 2017 map objects in images to ontology classes, and use DL-Learner (Bühmann, Lehmann, and Westphal 2016) to extract class expressions which function as explanations. Similar methods exists where RDF triples are extracted from the image and mapped to DBPedia (P. Wang et al. 2015).

In terms of explainability of link prediction by KGEMs, a few methods exist. W. Zhang et al. 2019 have developed CrossE which in addition to creating KGE for entities and relations, create embeddings for the interaction between them. This enables explanation of the link predictions made by CrossE by looking at closed paths (based on embeddings) between the subject and object in a triple. In other words, the reliability of a prediction is based on the number of similar patterns in the KG. d'Amato, Masella, and Fanizzi 2021 build on

CrossE by using a semantic measure to search for similar paths and patterns which justify the link prediction. Similarly, Rossi, Firmani, et al. 2022 find which combination of facts that has enabled this particular prediction. This method has the benefit that it can be applied to any KGEMs. However, this can lead to explanation that might not be immediately understandable, *e.g.*, ⟨`Billy_Halop`, `acted_in`, `Hell's_Kitchen`⟩ has the sufficient explanation:

- ⟨`Billy_Halop`, `acted_in`, `The_Angels_Wash_Their_Faces`⟩;
- ⟨`Billy_Halop`, `acted_in`, `On_Dress_Parade`⟩;
- ⟨`Billy_Halop`, `acted_in`, `On_Dress_Parade`⟩

Here, `acted_in` is the dominant predicate, but needless to say, an actor does not necessary act in one movie since they acted in three others. Therefore, more investigation is needed to determine how these triples relate.

In the domain of recommendations, KGs are also used. Y. Zhu et al. 2021 use a (encoded) KG in conjunction with a neural logic model (user-item interaction) to explain e-commerce recommendations. The encoding of the KG can be done with a KGEM or GCN. The neural logic model enables a user-specific rules to be mined from the KG. These two components create rich recommendations for each user as well as delivering faithful explanations for these.

3.2.3.3 Low-resource Learning

ML and deep learning (DL) has been successfully applied in a vast array of research problems. However, these methods rely on large labeled datasets. Annotation of these datasets is resource intensive. In low-resource learning the aim is to create robust prediction models with limited amounts of annotated training data (Hedderich et al. 2021; Zoph et al. 2016).

The use of KGs to aid in low resource problems such as few- and zero-shot learning is an emerging trend. Few- and zero-shot refers to the number of samples seen by a model during training (Xian et al. 2017). Consequently, zero-shot tries to predict on samples whose labels has not been seen during training. The use of the KG in few- and zero-shot learning can be divided into four methods (J. Chen, Geng, et al. 2021):

Mapping-based methods aim at creating a projection into a vector space where classification is based on some distance metric (cosine or euclidean distance).

Data augmentation is the notion of creating more data from existing data. For zero-shot learning, data augmentation using a KG aims at creating features for the unseen classes.

Propagation-based methods as the name suggest propagate features from seen to unseen classes via the KG.

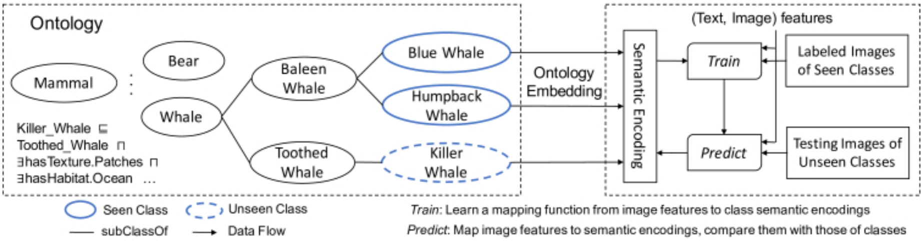


Figure 3.3: Zero-shot learning framework with ontology embedding (J. Chen, Lecue, et al. 2020). Here, an ontology of animals is used; however, this framework can be adapted to other tasks in *train* and *predict*. Published by IJCAI Organization. © 2020 International Joint Conferences on Artificial Intelligence Organization.

Class Features is a class of methods which use the KG contexts together with the model input to create richer features.

Examples of these methods are presented below.

Mapping-based. J. Chen, Lecue, et al. 2020 proposes a new method for the use of ontology embeddings in two zero-shot tasks, animal image classification and question answering. The framework is shown in Figure 3.3. This shows an example of how an image of a *Killer Whale* can be correctly classified using examples of *Blue* and *Humpback Whales.*

For text classification, Q. Chen et al. 2021 proposes to use existing KGs in a new zero-shot learning method. This methods is applied to a real-world dataset of tweets related to COVID-19 (classes *Advice, China, Mask, News, Transportation, USA, Vaccine*). This method is based on the language model BERT (Devlin et al. 2018) (and the sentence extension, S-BERT; Reimers and Gurevych 2019), to create S-BERT-KG. Figure 3.4 shows the methodology. This method is based on a pretrained S-BERT model and provided embedding of the KG. Each possible label (of a sentence) is represented in the S-BERT embedding space, which is projected into the KGE space using provided projection matrix (optimized previously). For unseen classes projected into the KGE space a similarity measure is used for classification. The proposed method performs $\sim 10\%$ above previous state-of-the-art.

Data augmentation. Leveraging prior knowledge is key to improving zero-shot learning. In contrast to previous methods using textual (or simple taxonomies) prior knowledge, Geng et al. 2021 take advantage of the increased expressively of an ontology. The method proposed, OntoZSL, incorporates prior knowledge using a text-aware ontology embedding. This ontology embeddings can then again be used to synthesize training samples from unseen classes. *e.g.*, an image of a *zebra* can be synthesized by using the knowledge of how it relates (in appearance) to *horses.* Simplified, a *zebra* is a *horse* with black and white stripes.

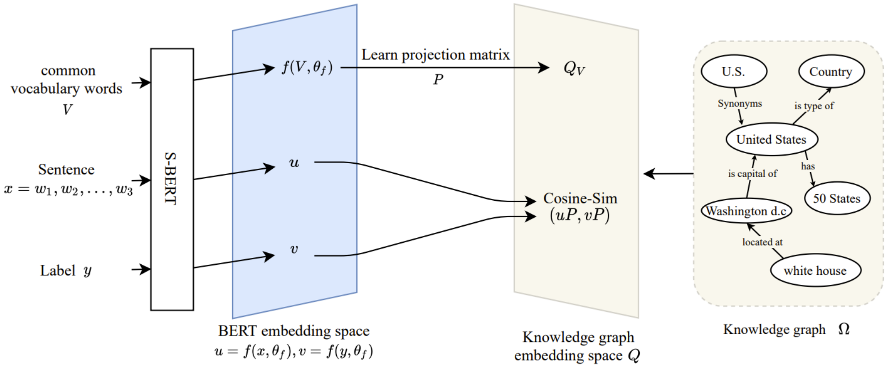


Figure 3.4: The S-BERT-KG architecture (Q. Chen et al. 2021). The BERT embedding space and KGE space is aligned such that unseen classes can be classified using similarity measures. © 2021 IEEE.

The proposed methods has been shown to increase prediction performance over image classification and KG completion tasks.

Propagation-based. Question answering is an area where KGs can improve results substantially. Bosselut, Bras, and Choi 2020 proposes to replace static KGs for contextual and dynamic KGs. An example of this can be seen in Figure 3.5. The contextual KG is extracted using *Commonsense Transformers* (COMET) (Bosselut, Rashkin, et al. 2019). This contextual knowledge can be used to reason about different questions with regards to a situation. The proposed method improve performance by $\sim 10\%$ over *e.g.*, the large language model GPT-2 (Radford et al. 2019). However, there are still improvements to be made to match performance of supervised BERT language models (Devlin et al. 2019).

Class Features. Amador-Domínguez et al. 2021 proposes a new way of using an ontology to initialize KGE approaches. As shown in Figure 3.6, the method embeds the lexical information from the ontology and KGs using Word2Vec (Mikolov et al. 2013a). The ontological embedding is concatenated with the lexical embedding of the entity. If an entity is described by multiple words, the mean of the vectors is used (Socher et al. 2013). Then the initialized vectors in the KGEMs are learned word embeddings enriched with ontology embeddings.

In addition to type assertions, most relations have domain or range restrictions and this method is able to inject this crucial information into KGEMs. The initialization technique show up to a 9% increase in performance for selected relations in WordNet (Miller 1995). Similar results are also shown for Freebase (Bollacker et al. 2008a). As KGEMs are expensive to retrain if new entities are added, this method can provide an approximate embedding for unseen entities that are more robust than purely lexical embeddings.

This section has introduced several KGE methods and applications of these. The introduced KGEMs are extensively used throughout this thesis, while NLP-based methods and GCNs are introduced as alternatives to them. In addition,

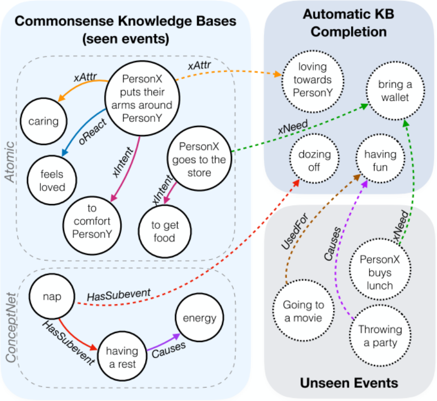


Figure 3.5: Left: static KG approach; right: dynamic, contextual KG approach. The dynamic KG is generated using *Commonsense Transformers* (COMET) (Bosselut, Rashkin, et al. 2019). Reprinted from Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi, COMET: Commonsense Transformers for Automatic Knowledge Graph Construction, 4762–4779, 2019, Creative Commons Attribution 4.0 International (CC-BY-4.0) from Association for Computational Linguistics.

this section introduced methods for embedding ontologies as this is relevant for the ontology-enhanced KG used throughout the thesis.

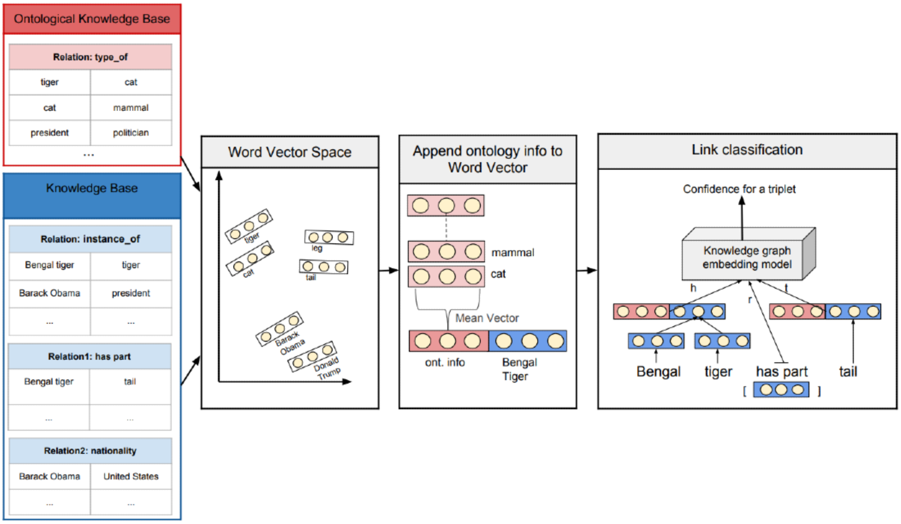
The prediction task in this thesis can be tacked either as and within- and out-of-KG problem. Therefore, this section introduced several applications for both. Link prediction, triple classification, entity classification and entity resolution are different problems where KGEMs are used and, thus, relevant for this particular part of the presented thesis work. However, this thesis take the out-of-KG approach as this gives some benefits, *e.g.*, task specific fine tuning.

Explainability of black-box models using KGs is presented as this is a central topic in a paper included in this thesis. An overview of the methods available were presented and this section focused on advances in explanations of KGEMs. The work regarding explanations in this thesis are heavily inspired by these works.

Finally, the prediction task tacked in this thesis concerns a low-resource domain and, therefore, several methods for increasing performance on these tasks were presented. The work in this thesis falls within *mapping-based* approaches; however, other methods were presented to aid in contextualization.

3.3 Ecotoxicological Effect Prediction Models

This chapter will first cover data integration efforts within the ecotoxicological community; thereafter, prominent methods for performing biological effect



Relation: type_of	
tiger	cat
cat	mammal
president	politician
...	

Relation: instance_of	
Bengal tiger	tiger
Barack Obama	president
...	...

Relation1: has part	
Bengal tiger	tail
...	...

Relation2: nationality	
Barack Obama	United States
...	...

Figure 3.6: The methodology for injecting ontological information into a KG completion task (Amador-Domínguez et al. 2021). Ontological and KG embeddings are combined to improve link prediction performance. Reprinted from Information Sciences, 564, Elvira Amador-Domínguez, Emilio Serrano, Daniel Manrique, Patrick Hohenecker, Thomas Lukasiewicz, An ontology-based deep learning approach for triple classification with out-of-knowledge-base entities, 85-102, 2021, with permission from Elsevier.

prediction.

3.3.1 Data Interoperability in Ecotoxicology

Animal testing is still the norm when assessing the toxicity of compounds on organisms albeit with substantial ethical and economical issues. Therefore, powerful methods that makes use of existing effect data are needed. These methods rely on high quality data and efforts have been made to incorporate data from various sources.

The system used in this thesis for extracting effect data is the ECOTOXicological Knowledgebase (ECOTOX) (Olker et al. 2022), which is one of the largest publicly available sources of such data. EnviroTox (Connors et al. 2019) is another such database concerning aquatic species; albeit, a lot smaller than ECOTOX and, therefore, not included further in this thesis.

ECOTOX contains over one million toxicity tests from 50,000 references concerning about 12,000 chemicals (Olker et al. 2022). The tests in ECOTOX are unfortunately biased toward certain species (and to some degree compounds) which are important for regulatory purposes; however, in a ERA applications these might not represent the ecosystem fully (Duffy, Dunlap, and Godduhn

2014). Furthermore, ECOTOX relies on proprietary identifiers, *e.g.*, CAS numbers for species. Another toxicity database, albeit smaller, ToxRefDB (Watford et al. 2019), aims at partially solving these issues by aligning vocabulary to Unified Medical Language System (UMLS) (Bodenreider 2004) and Medical Subject Headings (MeSH) (Rogers 1963). This enables the use of external data sources when performing predictive ecotoxicology. As these systems are based on (semi)-automatic extraction from publication the reliability of the data has been criticized, *e.g.*, Plunkett, Kaplan, and Becker 2015 claim that the evaluation of the original study source and that the biological significance of responses are lacking and, therefore, manual verification of ToxRefDB data is needed on a per study bases, partially defeating the purpose of an large integrated database. Improving the automated extractions can help mitigate some of these concerns (Hoff 2020). However, it is unlikely to abolish the issues completely.

In the domain of drug discovery the use of KG and ontologies (*e.g.*, Lin, Mehta, et al. 2017), which help to get rid of inconsistencies, is embraced. However, these ontologies are humancentric (and human curated) and the variation in the ecology domain is much larger. R.-L. Wang, Edwards, and Ives 2019 have mapped chemical toxicity to semantic mappings; however, in a fairly limited domain so far (19 chemical-species phenotypic profiles). The sheer size of the target domain is a major hurdle to overcome in ecotoxicological data integration.

3.3.2 Read-across

Read-across methods are developed to extrapolate from data-rich to data-poor compounds using similarity measures. Here, the data-rich compound is named source and the data-poor compound is named target for simplicity.

Figure 3.7 shows an example read-across task. A simple case would be to read-across from sources A, B, D to target C. This is simple as we have the most available information about the compound neighbours. But more often than not, both C and D are missing data and the toxicity needs to be estimated for both.

3.3.2.1 Expert Judgment

Firstly, a read-across can be defined thought expert judgment. Two main categories (with sub-categories) of approaches can be used by experts (Daston et al. 2014):

1. Analogue approach:

   a) Identical toxicants. A biotransformation of target compound results in the same toxicants as sources and, therefore, the same effect.

   b) Identical mode-of-action of adverse effect. The source and target compounds belong to a group that exhibit the same or similar toxicity in the organism.

2. Category approach:

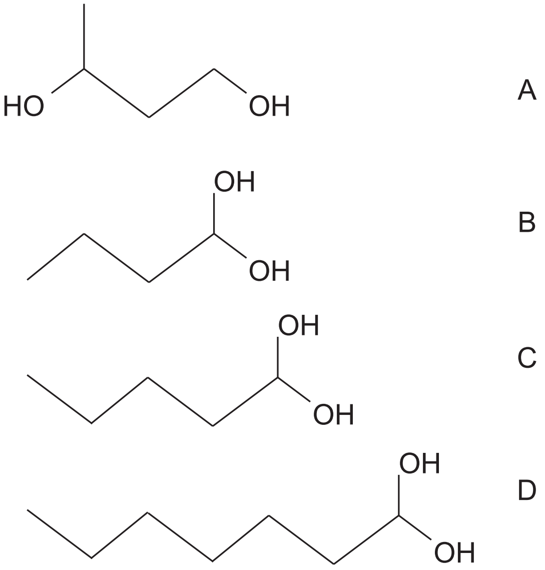


Figure 3.7: Read-across example. A, B, C, and D are chemical compounds. A read-across task would be to predict the toxicity of C, given the toxicity of A, B and D.

- Property trends. A group of compounds exhibit a trend in an observed toxicity tied to a chemical property.
- Non-observed property trends. Other than the observation property changes in a group of compounds. This is a weaker relationship than the directly observed property trend from above.

These expert judgements are excellent for rather small datasets. However, with the increasing amount of available data, automated methods based on chemical and biological similarity are becoming popular.

3.3.2.2 Chemical Similarity

Using the (structural) similarity among chemicals is the most used read-across method. Therefore, many methods have been developed to express chemical similarity. Similarity among molecular fingerprints is a well used method.

A topological fingerprint can be generated for compounds either in two- or three-dimensions by many methods, *e.g.*, Hosoya et al. 1999; Randić 1997; Wiener 1947. These methods have predefined binary markers which correspond to some topological property of the compound. The representation of a compound is then the set of these markers. Similarity between fingerprints are usually calculated with Tanimoto index (Tanimoto 1958) (also called Jaccard index in other fields).

Certain dissimilar chemicals can exhibit very similar toxicity properties; therefore, invalidating these similarity measures for the purpose of read-across

(Low et al. 2013; Mahmoud and Yousef 2019). Hence, the supplementing of chemical similarity with taxonomic similarity is emerging as a useful approach (*e.g.*, Grimm et al. 2016).

3.3.2.3 Taxonomic Similarity

There are no analogues to QSAR models for species due to the lack of defining features of organisms. Therefore, other methods, not dependent on *ad-hoc* features, need to be used. The use of mode-of-action or gene commonality is a viable option for taxonomic read-across (H. Zhu 2016). In addition, methods for extracting similarity through sequence analysis have been developed, *e.g.*, SeqAPASS (Lalone et al. 2016). However, taxonomic read-across is still a undeveloped research field.

Chemical and taxonomic similarity can be used in combination with both experimental and computational methods to increase the domain coverage of biological effect prediction.

3.3.3 Quantitative structure-activity relationship models

QSAR models (*c.f.*, Tsakovska, Diukendjieva, and Worth 2022) are based on analysis of the chemical structure and which biological processes the chemical exposure disturbs. Early models predicted systemic toxicity or adversity, *e.g.*, mortality, thereafter, developments were made to predict the effect on internal processes that have an effect on the whole organism (Kubinyi 2002).

3.3.3.1 Linear Models

Classical QSAR models are usually based on linear regression analysis of the relationship between the chemical and bioactivity. These models predict the potency (*e.g.*, bioactivity) of chemicals based on the change in physical properties (Fujita and Winkler 2016; Hansch et al. 1962). The development of such models is largely based on data from medicinal or toxicological research (Muratov et al. 2020; Olker et al. 2022). Therefore, the compounds used share common moieties which limit the models applicability domains. These models are ideal for use in medicinal drug development and synthesis as one can easily see the results of perturbing the chemical structure (Hansch et al. 1962).

The main drawback of these classical QSAR models is the assumption that similar chemical structures exhibit (linearly) similar bioactivity (Muratov et al. 2020). In fact, the applicability domain is defined by the structure-activity relationships (SAR) continuity. This continuity is defined where a change in functional-group (*e.g.*, R-group for amino-acids) has a linear effect on the activity (Peltason and Bajorath 2007). If a large potency change is observed, the perturbation is outside the applicability domain of the models. However, recently the use of ML and other non-linear methods have attempted to mitigate these issues.

3.3.3.2 Machine Learning Based Models

To solve the problem of SAR discontinuity, non-linear models have been developed. Many ML methods are used in the context of QSAR modelling, *e.g.*, nearest neighbour methods (Cover and Hart 1967), random forests (Breiman 2001), and support vector machines (Geppert et al. 2008).

Highly non-linear methods such as neural networks are less common due to the relatively small datasets. Nevertheless, efforts have been made to extract features from compound structures using graph convolutional networks (Sakai et al. 2021), but these methods have not yet matured in toxicity prediction.

This section presented work central to the ecotoxicological parts of this thesis. The integration of biological test data is of high importance to this thesis, and this section included the most prominent work regarding this. Thereafter, methods for extrapolation of these biological effect data were presented. Throughout this thesis the complexity of methods used increase, and therefore, this section presented methods from simple (and explainable) linear regression and read-across, to powerful (and black-box) ML methods.

Chapter 4

Summary of Research

This chapter presents extended abstracts for Papers I to III. Table 4.1 contains the shared and exclusive aspects of each paper.

4.1 Paper I - Knowledge Graph Embedding for Ecotoxicological Effect Prediction

In this work we introduced the use case of effect prediction using KGs, and the importance of this in downstream applications like ERA.

This work introduced TERA within the use case of effect prediction. The sources used in version one of TERA are NCBI (Taxonomy; Sayers et al. 2008), ChEBI (chemicals; Hastings, Owen, et al. 2016, PubChem (chemical similarity; Kim, J. Chen, et al. 2018), and ECOTOX (effect data; Olker et al. 2022). In addition, we used LogMap (Jiménez-Ruiz and Cuenca Grau 2011; Jiménez-Ruiz, Cuenca Grau, Zhou, et al. 2012) and Wikidata (Vrandecic and Krötzsch 2014) to align the disparate sources.

The effect prediction modelling approach used in this paper is to model mortality as a binary state[1] where effects defined as mortal in ECOTOX are treated as 1 and others as 0.

We developed two baseline models to compare to the use of KGE. First, a symbolic model based on distances in the KG is developed. This used chemical similarity as defined in PubChem, and species similarity based on taxonomic distance. The prediction was then based on the closest points in terms of chemical and species similarity. Second, a neural network with chemicals and species one-hot encoded, *i.e.*, vectors uniquely defining each chemical or species. Finally, we defined a model based on three KGEM (TransE, DistMult, and HolE). This model takes the architecture from the second baseline and replaces the one-hot encoding with KG entity vectors learned from the KGEM. The prediction and KGE models were optimized jointly.

The effect data was split 70%/30% for training and testing. We used 10 fold cross validation to train the models. We showed that the symbolic baseline model is not suited for this problem, lacking 20% behind the second baseline in terms of F_1-score (harmonic mean of precision and recall). The embedding based models improved slightly over the neural network baseline in terms of F_1-score, however, we saw larger improvements for F_2-score (recall is weighted twice that of precision) indicating that the embedding based models are able to catch more lethal effects.

[1] This is true for individuals but not necessarily for populations.

This paper provided a proof-of-concept for further work using KGs in toxicology. Moreover, a large amount of work was done to integrate different data sources into TERA and implementations of the models.

4.2 Paper II - Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings

This paper builds directly on Paper I with refined data and modeling choices, and validation strategies.

We extend TERA with further resources, the Encyclopedia of Life (EOL), MeSH, and ChEMBL, and made further use of ontology matching (OM) tools (adding AML for alignment) and Wikidata to align them more efficiently. In that regard, a full rewrite of the TERA libraries was completed. This introduced larger modularity and enabled faster and better integration of new sources.

In Paper I, we used three KGEMs and no mixing between them. In this paper, we make use of nine state-of-the-art and commonly used KGEMs. We implemented these in a separate library in such a way that they can be easily used as components in larger model architectures. Furthermore, we allowed cross use of the models in the two separate TERA parts: species and chemical KGs, respectively. This resulted in 81 model combinations. In contrast to Paper I, we perform a large scale parameter search for each of the models.

We introduced a fine-tuning model architecture which enabled us to initialize a model with pre-trained KGEs and finetune them to fit to the task at hand.

In Paper I, we only used one validation strategy, unknown species-chemical combinations.[2] However, some entities are substantially more used than others, therefore, the results can be shifted w.r.t. each entity. Therefore, we introduced three new validation strategies: no intersection between the set of chemicals in training and testing; no intersection between the set of species in training and testing; and finally, the combination of the previous two. These strategies gave a more realistic view (w.r.t. other work in ecotoxicological effect prediction) on the prediction performance of the given models. Furthermore, they increased the separation in performance between models such that they could be ranked easier than in Paper I where the performance was very similar among KGEMs.

We found that with parameter tuning the (one-hot neural network) baseline performs equal to KGEs for the simplest validation strategy (unknown pairs). Moreover, the gap between the baseline and KGEs increased with increasing difficulty, unknown species, then unknown chemicals, and finally, with both unknown. We found that the finetuning architecture improved results in the difficult settings (chemical and species unknown) up to 10% over non-finetuned KGEs.

Finally, we analyzed the trends of which KGEMs perform well in all settings. We found that ComplEx performed well in all scenarios, albeit being surpassed in a few cases. To explain why, we looked at how models explain variance

[2] All chemicals and species seen by the models during training; however, new combinations was used for validation

(using Principal Component Analysis (PCA)) and we found that there is a negative correlation between explained variance and predictive performance. This indicated that certain KGEMs perform poorly due to introduced noise from the training process.

These findings can be used to develop further specialized KGEMs or fine tuning architectures which might be better at taking advantage of the unique structure of the data in TERA.

4.3 Paper III - Adverse Biological Effect Extrapolation and Explanation using Knowledge Graphs

This paper accumulated the knowledge from previous papers in terms of KG sources and KGEMs that are most appropriate to embedding TERA. In this paper, we took a pragmatic approach from the ecotoxicological community. We moved away from a classification problem to a regression one, predicting chemical concentrations causing 50% mortality. The importance of this is apparent as the number of quality (and trustworthy) data points are reduced tenfold from Paper II. The move to continuous prediction also introduces a lot of noise into the model; therefore, a decision was made to move to a less noise prone model, like a SVM. SVMs do not perform state-of-the-art in this task compared to a neural network approach, but are preferable as they greatly reduce the variance between different random perturbations in data splits.

We showed that introducing KGE for regression effect prediction improved results by up to 40% depending on the data splits over a similar baseline as in Paper II.

To emphasize the usefulness of KGE in this task, we developed qualitative and quantitative methods to gain insight. First, we explored the entities and their relations, which are similar to prediction entities. This can give a domain expert the possibility to validate whether the model has sufficient knowledge to make an educated prediction. Second, two quantitative methods where developed, both based on the KG density in the neighborhood of prediction entities. Prediction of error based on neighborhood density gave an indication of confidence in the model. Finally, as an extension of this, we used regulatory categories for hazard (based on chemical concentrations intervals) to assert if the KG is suitable for use in this prediction task.

These developments will enable the use of complex models in the toxicological community without the loss of interpretability, which is highly important for regulatory applications.

	Paper I	**Paper II**	**Paper III**
TERA KG Parts	ECOTOX ChEBI PubChem NCBI Taxonomy	ECOTOX ChEBI PubChem NCBI Taxonomy EOL ChEMBL MeSH	Same as in Paper II.
KGE models	TransE DistMult HolE	TransE DistMult HolE ComplEx RotatE pRotatE HAKE ConvKG ConvE	ComplEx
Prediction models	Symbolic Baseline Shallow MLP Same KGEM	Deep MLP Mixed KGEM Hyperparameter tuning	SVM
Evaluation	*Classification* Unknown pair 70/30% split	*Classification* Unknown pair Unknown species Unknown chemical Both Unknown 5-fold validation	*Regression* Organic compounds Group gap-filling 5-fold validation
Analysis		Explained Variance	KG density Shared facts Error estimation

Table 4.1: This table shows the common and distinct parts related to Papers I to III. The rows refer to datasets include in **TERA**, the **KGEM** evaluated, the **Prediction models** used, the **Evaluation** and **Analysis** performed, and finally, the methods used to gain **Prediction insights**. Here, *shallow* and *deep* multilayer perceptrons (MLPs) refer to the number of neural network layers, *deep* has > 1 layers. Moreover, *same KGEMs* indicates that the same KGEM is used to embed $KG_{species}$ and $KG_{chemical}$ while mixed uses all combinations to find the ideal one.

Chapter 5

Discussion

This chapter will first discuss how the work presented in Chapter 3 relates to the work conducted in this thesis. Thereafter, each of Papers I to III are discussed in the context of the main research fields encompassed by this thesis: SW and ML, and ecotoxicology. Finally, the limitations and outlook of the work is presented.

5.1 State of the Art

Chapter 3 introduced domain specific KGs and their applications, along with an extensive introduction to KGEMs and a handful of important applications of these. Finally, we introduced *in silico* methods for extrapolation and prediction of adverse toxicological effect. The following will relate the most influential sources from Chapter 3 to Papers I to III.

5.1.1 Domain Knowledge Graphs

Section 3.1.1 presented several domain specific KGs. The papers in this thesis rely on such a KG through-out and the choices made during the construction of TERA are not unique for this particular task.

Kiesling et al. 2019 is an example of the evolving KG: as we obtain new knowledge it is necessary to update the KGs while being able to revert if demanded. In ecology, it is often needed to update our understanding as it evolves. Moreover, Fu et al. 2019; Nayak, Kesri, and Dubey 2020 relate directly to the effect part of TERA where tests (experiments in TERA and software tests) are documented.

We directly use chemical data in TERA and Farazi et al. 2020 create a foundation ontology for this purpose which makes interoperability between sources easier. This, used in conjunction with intelligent search (*e.g.*, Huang, Y. Wang, and Yu 2020), enables users to be more efficient when working in complex domains, such as ecotoxicology.

Aumayr, M. Wang, and Bosneag 2019; Zhan and Yin 2018 follow a similar approach as throughout this thesis where a KG is used to express knowledge of the domain, followed by a KGE approach before using a neural network to detect anomalies. The targets of Paper I and Paper II can be thought of in a similar manner as it also concerns binary prediction.

Defining products in KGs can be very useful, as demonstrated by Haussmann et al. 2019. The chemical part of TERA contains several product categories and their components, *e.g.*, insect repellents which are usually a mix of *diethyltoluamide* and *ethanol*. This can be very useful in data access applications.

5.1.2 Knowledge Graph Embedding Applications

Section 3.2.3 presented several KGE applications. We presented both within-KG and out-of-KG applications as the prediction problem in this thesis can be treated as both.

For the within-KG task, we can treat effect prediction either as link prediction or triple classification, *e.g.*, $\langle \texttt{DEET}, \texttt{affects}, \texttt{?} \rangle$ (where `?` is any taxon) or $\langle \texttt{DEET}, \texttt{affects}, \texttt{Daphnia magna} \rangle \in \{True, False\}$.

Most out-of-KG tasks use a KG as background knowledge to inform predictions. However, Riedel, Yao, McCallum, and Marlin 2013 jointly trained a KGEM and NLP model by aligning corpora (words). This is in line with the work in Paper II where we fine-tune the KGEM by jointly training it with the effect prediction task.

Initially, Paper I considers the effect prediction as a recommender system and, as in F. Zhang et al. 2016, the output is sparse and a KG needs to inform the recommendations. We take a different approach to representing the individual data sources; however, the goal is the same. This sparsity is linked to low-resource learning where the goal is to create robust models for large domains with small amounts of training data (Hedderich et al. 2021; Zoph et al. 2016). In effect prediction, we have large amounts of metadata but a relatively small amount of training samples. This was discussed in detail in Paper I. The hope is that by using KGEs we are able to increase the value of low-resource species or chemicals, *i.e.*, where only a few laboratory experiments are available.

The different types of low-resource learning can be used in different ways. Amador-Domínguez et al. 2021 initialize KGEMs with existing word embedding which is similar to the fine-tuning approach used in Paper II. Moreover, Geng et al. 2021 took advantage of an ontology to augment the data. The KGEs used in Papers I to III act as augmentations in increasing the representation dimension of the entities, *i.e.*, the relations between entities in the embedding space create augmentations from low- to high-resource entities. Similarly, J. Chen, Lecue, et al. 2020 used mathematical operations to add entities together, *e.g.*, we could potentially create the entity representation of `Daphnia magna` by adding the vector representations of `Daphnia` and `magna`.

Our approach defines the effect prediction problem as a combination of within- and out-of-KGs, *i.e.*, the effect data is included within TERA, but it is excluded from the KGE process. Then the effect data plays a key role in the fine-tuning of the KGEs. This creates dynamic embeddings which is also used in Bosselut, Bras, and Choi 2020.

5.1.3 Biological Effect Extrapolation

As mentioned in Section 3.3.1, the general drawback of current effect prediction methods is the limited applicability domain. More specifically, linear QSAR models perform well as they are applied to very similar chemicals, where toxicity is linearly transferable between compounds. However, out-of-domain predictions will yield non-linear errors which is far from ideal; albeit, the relevance of this depends on downstream applications. One way of partially mitigating this would be data integration. Databases, such as ECOTOX and ToxRefDB, are created from extracting experimental data directly from studies. However, the (semi)-automated processes used can introduce errors and discrepancies as demonstrated in Paper III where the standard deviation of experiments is discussed. KGs and ontologies could potentially offer a solution to the data integrating problem by requiring published data to cohere to a given format. This would speed up model development and open the possibilities for other, more complex methods.

Increasing complexity and the use of other ML techniques (*e.g.*, Geppert et al. 2008) mitigate some of the linear QSAR problems. Nevertheless, these methods require larger amounts of training data to be effective and might suffer the same fate as linear models with sufficiently different compounds.

Read-across methods take a different approach to extrapolation. Using expert judgments can reduce the overall error in extrapolations; however, this is not viable for larger datasets. Using objective similarity measures is one of the leading techniques to perform read-across (Daston et al. 2014). The objectivity removes bias in the model and feature design, and enables agnostic evaluation which is more in line with modern model evaluation strategies. This has enabled models using chemical similarity to achieve very good results (*e.g.*, Hosoya et al. 1999; Randić 1997; Wiener 1947). However, on the other side, biological similarity is more difficult to define and, therefore, accuracy of these methods is less than desired (Grimm et al. 2016).

5.2 Contribution Summary

We have presented three papers which address the task of ecotoxicological effect prediction, an important prerequisite for ERA. These papers have shown the benefits of integrating relevant disparate data sources into a unified system, in our case, a KG called TERA. The sources in TERA were integrated semi-automatically, using SWT and LOD to facilitate the alignment of the sources. Some data sources lack proper documentation[1] and, therefore, manual transformation of data through scripts was necessary. The performance gains from using KGE in the prediction task are substantial and the KG enables explanations and insights into the predictions.

Paper I provided an introduction and a proof of concept for using KGE in ecotoxicological effect prediction. However, large parts of the background work in creating TERA are omitted in this paper. Paper II builds on Paper I by first giving a detailed explanation of the creation of TERA. Followed by improved data sampling techniques which reflect *in silico* experiments from the literature. Finally, extensive evaluation of nine KGEMs in conjunction with simple and complex neural networks and the fine-tuning of embeddings. Paper III was aimed at the ecotoxicological community where we put more emphasis on known example cases. We also used TERA for the practical purpose of explaining the

[1] or the documentation is not machine readable.

predictions. For simplicity, only one KGEM was used: ComplEx, based on the results of Paper II.

5.3 Semantic Web and Machine Learning

The SW community are large proponents of real use-cases (*e.g.*, the In-use track at the International Semantic Web Conference; Ghidini et al. 2019), and in this work, we have explored a novel use-case for the SW community. Firstly, we have exploited available open LOD sources, *e.g.*, Wikidata, to integrate the not-linked data sources necessary for ecotoxicological effect prediction. Secondly, we have shown that this use-case strains OM systems on commodity hardware, and needs some pre-processing to enable effective alignment of the sources.

Paper I introduced TERA and the use-case of ecotoxicological effect prediction. This is a novel real world application using established SWT. We constructed TERA with what we deem as the bare minimum of sources and demonstrated how to align them through open data and SW tools. This proof of concept was neglecting the data selection process where we naively split training and testing data using unique chemical-species pairs. However, this actually led to data leakage as some chemicals and/or species are non-proportionally used in experiments. Therefore, for certain chemicals or species the predictions are perfect, while for those less commonly used ones the models are much less informed and performed worse. Paper II remedies this by creating three additional sampling strategies in addition to the one used in Paper I. These improved validation strategies enabled us to distinguish models easier in Paper II than in Paper I, where the three KGEMs performed very similarly.

Paper II presented, to our knowledge, the most extensive evaluation of KGEMs on a prediction task. This shows that KGEMs have a large application domain. We showed that certain KGEMs fail completely in embedding TERA; either the species or chemical sub-graphs, or both. The unique structure of TERA is the main cause as it is in majority hierarchical KG (most relations link hierarchical levels). Methods such as HAKE (Z. Zhang et al. 2019a) were created for similar tasks where the hierarchy is important; however, the chemical part (KG_C) of TERA does not have simple tree structures (like KG_S, the species part, does), but the hierarchy resembles a forest structure. This poses the problem of separating entities of the KG in the latent space, *e.g.*, *Benzamides* is a sub-class of *Amides*, *acids*, and *hydrocarbons*, and these are all sub-classes of *organic compounds*. Most KGEMs cannot handle such structures, and the three super-class embeddings will *collapse* into one. We discussed a solution to the problem which would be to add a minimum distance restriction to the entity embeddings. This ensures that the entities remain separate in the latent space. However, this does not account for entities which are the same, *e.g.*, in alignment tasks, and therefore, more investigation into these mechanisms is needed.

As mentioned, Paper III is aimed at the ecotoxicological community where explainability is valued highly. This is evident in the relatively simple models used in current *in silico* systems. Linear regression or decision tree models retain the desired explainability; however, they lack the capacity of state-of-the-art methods. We used the KGEM that performed best in Paper II (ComplEx) in order to limit the complexity of the experiments. We also omitted the fine-tuning architecture for the same reason and settled on a SVM as it is robust to noisy data. For certain data settings (from Paper II) we showed improved performance over the baseline; however, in the chemical gap-filling task, both the baseline and KGEMs performs worse than random ($R^2 < 0$). Therefore, chemical gap-filling is a harder task than species gap-filling and cannot be based on metadata directly. This can be remedied by using derived chemical features in addition to KGE. This is not an option for species as it is hard to create defining features. Metadata in the form of a KG might be the best option.

To underline these findings we introduced two methods — one quantitative and one qualitative — to gain partial explanations of the prediction of the SVM used in this paper. The quantitative method is based on the density of the KG neighbourhood which gives an indication of how much the training data has informed the model decision. This can increase confidence in the predictions. We also extend this quantitative method to predict the error of a given prediction. This is also analogous to a prediction confidence. These methods can easily be extended to other prediction tasks, using one, two (as in this work), or more entities involved in the prediction task.

The qualitative method presented in Paper III is especially relevant in this use-case as it can help domain experts assess the prediction confidence. This method presents facts (triples) in the KG that were relevant for the individual prediction. This way an expert can assess whether the prediction was based on correct domain knowledge (from the KG) or if further investigation is necessary, *e.g.*, if the presented facts are too generic (*e.g.*, $\langle$`FatheadMinnow`, `endemicTo`, `AquaticEnvironment`$\rangle$), a domain expert might hesitate in including the prediction in downstream applications. This method will just become better as more information is added to the KG.

We have not previously discussed the use of GCNs (Kipf and Welling 2017) in this work. Unfortunately, GCNs require node features to create meaningful connections between nodes in the graph and, as mentioned above, it is difficult to define numerical features for species, and especially classes (taxa), in the KG. In addition, GCNs require the use of adjacency matrices to represent the graph and due to the immense size of TERA this would not fit on reasonable hardware. Therefore, in a sense, KGEMs are more generic and can be applied to a wider spectrum of tasks. We also found that apart from a few works in protein-protein interactions in drug discovery, KGEMs had not been extensively explored for non-graph[2] based prediction tasks.

A large part of the work in this thesis was the implementation of TERA. The details of the construction and implementation of TERA were largely omitted in Papers I to III. In Appendix A, a detailed description of the creation of TERA and the functionality of the accompanying tools is described. Some sources of data, such as PubChem (Kim, J. Chen, et al. 2018), in TERA are

[2] *i.e.*, not classification of nodes, (predicate) link prediction, etc.

already in formats preferred by SW tools. However, other sources are legacy databases with only human readable documentation, and are prime candidates for bringing into the LOD universe. These converted data sources could simplify the implementations of front end systems used for data access today. The NCBI Taxonomy is a prime example of this. A taxonomy organised in tabular form is not ideal both from practical (implementation) and philosophical perspectives. Moreover, as species hierarchies are strictly defined by humans and are more fluid in reality, tabular organisation makes it harder to deal with anomalies. Interest in evolving and versioned KGs is increasing, *e.g.*, Pernischova et al. 2019 aim at predicting the cascading effects in a KG based on changes made.

Most implementations of KGEMs are not modular as they are designed to be used in link prediction tasks. We create a modular KGEM implementation that enables individual KGEMs to be part of a larger ML architecture.Other applications of this modality could be though of, *e.g.*, jointly learning multiple KGEMs to increase performance.

The joint learning architecture developed in Paper I was used to fine-tune pretrained KGE in Paper II. This can mitigate the potential delta when solely using KGEMs while keeping the benefits of KGEMs as mentioned above. The idea of fine-tuning models for a specific task is commonly used in image recognition, but the lack of potential fine-tuning use-cases, using KGs, means that this area has not been extensively explored. Effect prediction could be considered as a benchmark for other KGEs fine-tuning architectures in the future.

The collection of papers have introduced a novel use-case to the semantic web community. We have applied selected SW tools to aid in the creation of TERA, a KG that covers large parts of the ERA and effect prediction domain. This KG enabled us to find the limits of popular OM tools and has shown how certain KGEMs fail when presented with specific KG structures.

5.4 Ecotoxicology

This section will discuss the implication of the papers on ecotoxicological effect prediction and ERA communities. Ecology is a research field with long traditions and established norms. The use of models in effect prediction is not new; however, the adoption of new technology has not been as fast as in other natural science fields.

We have seen a shift toward data driven modelling and assessment methods (*e.g.*, Tollefsen 2018) and the papers presented in this thesis help move this forward. However, in the literature we have seen very specific models only concerning a handful of chemicals (*e.g.*, 137 compounds; Sushko 2011) and/or single species. The methodology presented in this thesis certainly does not outperform these specific methods; however, using KGs will enable the use of larger modelling domains, and accelerate model development overall. As more data and newer ML techniques becomes readily available, the move from small, specific models to larger, generic models is just on the horizon. We have demonstrated this through the three papers presented in this thesis.

Firstly, in Paper I we present the proof of concept that KGs can aid in the prediction of biological effects. We identified the most important data sources in the domain and created a few binary prediction models based on a KG created from these. We presented a baseline which was strictly based on existing methodologies. This involved using chemical similarity, in the form of fingerprints (881 bit representation of a chemical) and the distance between species in the KG. Thereafter, a prediction is made based on the most similar chemicals and species. We showed that in this binary setting this method is outperformed by a simple (without prior-knowledge) machine learning model. We expanded on this model with the use of KGE, and increase the performance further. As this is a binary task, the end goal of the method would be ranking of chemicals which would need to be explored further, by either more specific models or laboratory experiments. The model developed in Paper I only consider mortality and, therefore, we improved upon the methodology to include more experimental endpoints in subsequent papers.

In Paper II, we expanded TERA with more sources which improved data access in domain tasks. For the purpose of the study, we used endpoints related to mortality; however, the model can be extended to others. This study also includes far more datapoints than the original proof of concept study. This enabled the use of more representative data sampling strategies. These are directly related to chemical and/or species gap-filling, and we show that including background knowledge in these settings is not just beneficial, but necessary.

Paper III explored a use-case directly relatable to other work in adverse effect prediction, *i.e.*, read-across or QSAR models. We use a SVM in conjunction with a KGEM to predict the chemical concentrations that causes mortality to a portion of the population (50% in this study). This task is more complex in nature as laboratory experiments have large variability. In fact, around a quarter of the data gathered from TERA to perform this task had standard deviations of one order of magnitude or larger. We mitigated this by only considering experiments with more than three results (all other variables being equal) which are averaged to create training and testing data. This, in addition to using a predictor which is robust to noise, SVMs, limits the variability of results. When integrating these methodologies into ERA pipelines we can limit the variability further by using data curated by experts, as certainly some of the effect data are purely noise (*e.g.*, wrong unit in the database entry). Equally important for the integration of this methodology is the ability of experts to trust the output of the models. Therefore, we created methods for gaining insight into the prediction. Assessing the density of (meta)data in TERA can help identify parts of the KG which could benefit from additional data. We showed that there is a weak correlation between the density and model accuracy which is a move in the right direction; however, more investigation is needed for this to be a fully fledged explanatory method. We used these density metrics to further predict the error of an individual prediction. We showed that this model for predicting errors is able to produce results with an average error less that the uncertainty in the effect data. Moreover, with more curated data and, therefore, lower variability in the effect data we can expect this method to perform even better.

5.5 Limitations and Opportunities

Overall, the papers presented have increased our understanding of the use of background knowledge in biological effect prediction tasks. However, the work has also revealed some limitations with this approach.

Firstly, there are always more data to be added to TERA. We have added what we deemed to be the most relevant data for effect prediction and, therefore, many potential sources were omitted. Adding additional sources will require alignment and the payoff (in prediction performance) versus time spent, using either manually or automated techniques remains unknown.

In Paper II, we showed that in certain prediction scenarios the chemical KG did not impact predictions. This comes down to low correlation between chemical classifications (hierarchy) and chemical features (not in TERA). These features have a larger impact on toxicity prediction. This is not the case for the species taxonomy, as species do not have inherent features.[3] Therefore, species features generated from the KGEMs can be used to expand existing domain models.

The models created in this work are black boxes and are not explainable as they are. This is a big hurdle for AI in ecotoxicological research in general, where interpretable models are highly valued. We explored a few methods for mitigating this in Paper III; however, this area largely remains unexplored.

In the pursuit of generic models with large applicability domains, competitive results with specific models is not achievable. However, we have limited ourselves to only consider non-literal KGs in prediction and, as mentioned, the chemical features will have a large impact on predictions. The difference in performance might be solved by adding more data to TERA. But, increasing the size of TERA without adding more effect data can lead to lower performance due to added noise.

The use of AI in ecotoxicological research is an emerging trend and this work falls into this largely unexplored area. Moreover, this work is separated from previous *in silico* methods by its use of external data and increased applicability domain. This use of external (background) data is also a significant research area in the broader AI fields. This work fits into the intersection of these two areas, and is a significant contribution to both.

[3] Individuals does at a point in time; however, models work at a higher abstraction.

Chapter 6

Conclusion and Future Outlook

This thesis has presented three papers in the novel intersection of SWT, ML, and ecotoxicological effect prediction. The papers have individually shown interesting results going from KG construction to proof-of-concept and extensive evaluation and, finally, a relatable domain task and moving toward explanations of the predictions.

The commonality and contrasts of the papers have been presented and the impacts of the papers on the fields of SWT, ML, and ecotoxicological effect prediction have been discussed.

Impact on the Ecological Domain. This thesis has expanded on the idea of increasing the use of AI in ecological research. The papers have shown that the integration of disparate data sources improves prediction models in addition to the natural benefits of data integration. We have shown that it is possible to create generic effect prediction models; albeit, this generality reduces performance compared to specific models. The ecotoxicological research community has to a large degree avoided these complex, generic models due to the lack of explainability. In the final paper we explore methods for mitigating this concern.

Impact on Machine Learning. We have extensively evaluated KGEMs on this specific use case. Moreover, this has exposed drawbacks of certain models as they fail completely to capture the information in TERA. This is to be expected to some degree as these are generic models and, therefore, we have developed a fine-tuning strategy to reduce some of these drawbacks.

Impact on Semantic Web Research. As a collection, the papers presented in this thesis have shown the use of SWT in a novel use case. As indicated in the discussion, this use case can be included in the evaluation of SW tools. We have discussed the use of automated KG creation tools; however, due to the ambiguity of the ecological data we did not explore these further. Albeit, the development of such tools could be accelerated by the manually curated TERA. Furthermore, we have shown that alignment of sources in TERA is challenging and the use of TERA to evaluate the scalability of these OM tools is worth considering. Finally, the ecotoxicological effect prediction task is a good prediction task where representation of the KG is key, without being part of the prediction itself. Therefore, it can be used as a benchmark to test latent representations created by KGEMs.

Future Outlook. There are certainly avenues that were not explored in this thesis. We have limited use to the present dataset based on TERA, which means that the fine-tuning models have not been evaluated for other tasks. In addition, as mentioned in Paper II certain properties of TERA means that the embeddings might suffer from a collapse (equivalence of entity embeddings) in latent space.

Both of these problems could be solved by modifying existing or developing new KGEMs as mentioned in Paper II.

TERA has great potential to be an important domain tool. However, this will require integration into current pipelines for ecotoxicology domain experts which might not be versed in SW tools or bio-informatics in general. The integration of TERA and the prediction models in ERA pipelines is important to test the validity and integratability of the presented work.

Resources. All code and data (where licences permit) related to this thesis is openly available at https://github.com/NIVA-Knowledge-Graph and reuse is encouraged. This included the repositories to produce results for all papers and application programming interfaces (APIs) and scripts to create TERA (described in Appendix A).

Bibliography

Abd Nikooie Pour, M. et al. (2020). “Results of the Ontology Alignment Evaluation Initiative 2020”. In: *15th International Workshop on Ontology Matching*, pp. 92–138.

Abu-Salih, B. (2021). “Domain-specific knowledge graphs: A survey”. In: *Journal of Network and Computer Applications* vol. 185, p. 103076. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2021.103076. URL: https://www.sciencedirect.com/science/article/pii/S1084804521000990.

Adadi, A. and Berrada, M. (Sept. 2018). “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. In: *IEEE Access* vol. PP, pp. 1–1. DOI: 10.1109/ACCESS.2018.2870052.

Agency, U. E. P. (2004). *Chemical Hazard Classification and Labeling: Comparison of OPP Requirements and the GHS*. https://www.epa.gov/sites/production/files/2015-09/documents/ghscriteria-summary.pdf.

Agibetov, A., Jiménez-Ruiz, E., et al. (2018). “Supporting shared hypothesis testing in the biomedical domain”. In: *J. Biomedical Semantics* vol. 9, no. 1, 9:1–9:22.

Agibetov, A. and Samwald, M. (2018). “Global and Local Evaluation of Link Prediction Tasks with Neural Embeddings”. In: *4th Workshop on Semantic Deep Learning (ISWC workshop)*, pp. 89–102.

— (2020). “Benchmarking neural embeddings for link prediction in knowledge graphs under semantic and structural changes”. In: *J. Web Semant.* vol. 64, p. 100590. DOI: 10.1016/j.websem.2020.100590. URL: https://doi.org/10.1016/j.websem.2020.100590.

Algergawy, A., Cheatham, M., et al. (2018). “Results of the Ontology Alignment Evaluation Initiative 2018”. In: *13th International Workshop on Ontology Matching*, pp. 76–116.

Algergawy, A., Faria, D., et al. (2019). “Results of the Ontology Alignment Evaluation Initiative 2019”. In: *14th International Workshop on Ontology Matching*, pp. 46–85.

Ali, M. et al. (2020). *Bringing Light Into the Dark: A Large-scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework*. arXiv: 2006.13365 [cs.LG].

Alshahrani, M. et al. (2017). “Neuro-symbolic representation learning on biological knowledge graphs”. In: *Bioinformatics* vol. 33, no. 17, pp. 2723–2730.

Amador-Domínguez, E. et al. (2021). “An ontology-based deep learning approach for triple classification with out-of-knowledge-base entities”. In: *Inf. Sci.* vol. 564, pp. 85–102. DOI: 10.1016/j.ins.2021.02.018. URL: https://doi.org/10.1016/j.ins.2021.02.018.

Amin, E. and Bhattacharyya, P. (2019). “Survey on Generic and Biomedical Knowledge Graph”. In:

Animal and Plant Health Inspection Service, U. D. o. A. (2021). “Annual Report Animal Usage by Fiscal Year: Total Number of Animals Research Facilities Used in Regulated Activities (Column B)”. In:

Arenas, M. et al. (2014). “Enabling Faceted Search over OWL 2 with SemFacet”. In: *OWLED*.

Arnaout, H. and Elbassuoni, S. (2018a). “Effective Searching of RDF Knowledge Graphs”. In: *Web Semantics: Science, Services and Agents on the World Wide Web* vol. 48, no. 0. ISSN: 1570-8268.

— (2018b). “Effective searching of RDF knowledge graphs”. In: *Journal of Web Semantics* vol. 48, pp. 66–84. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2017.12.001. URL: http://www.sciencedirect.com/science/article/pii/S1570826817300677.

Ashburner, M. et al. (May 2000). “Gene Ontology: tool for the unification of biology”. In: *Nature Genetics* vol. 25, no. 1, pp. 25–29. DOI: 10.1038/75556. URL: https://doi.org/10.1038/75556.

Aumayr, E., Wang, M., and Bosneag, A.-M. (2019). “Probabilistic Knowledge-Graph based Workflow Recommender for Network Management Automation”. In: *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–7. DOI: 10.1109/WoWMoM.2019.8793049.

B. Cuenca Grau et al. (2008). “OWL 2: The Next Step for OWL”. In: *J. Web Semantics* vol. 6, no. 4, pp. 309–322.

Baird, D. J. and Barata, C. (1998). “Variability in the Response of Daphnia Clones to Toxic Substances: Are Safety Margins Being Compromised?” In: *Diversification in Toxicology — Man and Environment*. Ed. by Seiler, J. P., Autrup, J. L., and Autrup, H. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 399–406. ISBN: 978-3-642-46856-8.

Baralis, E. and Fiori, A. (2008). “Exploring Heterogeneous Biological Data Sources”. In: *2008 19th International Workshop on Database and Expert Systems Applications*, pp. 647–651. DOI: 10.1109/DEXA.2008.116.

Benson, T. (2012). *Principles of Health Interoperability HL7 and SNOMED*. Health Information Technology Standards. Springer London. ISBN: 9781447128014. URL: https://books.google.no/books?id=CNcCcdUsoOYC.

Blagec, K. et al. (Apr. 2019). “Neural sentence embedding models for semantic similarity estimation in the biomedical domain”. In: *BMC Bioinformatics* vol. 20, no. 1, p. 178. ISSN: 1471-2105. DOI: 10.1186/s12859-019-2789-2. URL: https://doi.org/10.1186/s12859-019-2789-2.

Bodenreider, O. (Jan. 2004). “The Unified Medical Language System (UMLS): integrating biomedical terminology”. eng. In: *Nucleic acids research* vol. 32, no. Database issue. 32/suppl_1/D267[PII], pp. D267–D270. ISSN: 1362-4962. DOI: 10.1093/nar/gkh061. URL: https://doi.org/10.1093/nar/gkh061.

Bollacker, K. et al. (2008a). “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. Vancouver, Canada: Association for Computing Machinery, pp. 1247–1250. ISBN: 9781605581026. DOI: 10.1145/1376616.1376746. URL: https://doi.org/10.1145/1376616.1376746.

— (2008b). “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. Vancouver, Canada: Association for Computing Machinery, pp. 1247–1250. ISBN: 9781605581026. DOI: 10.1145/1376616.1376746. URL: https://doi.org/10.1145/1376616.1376746.

Bordes, A., Chopra, S., and Weston, J. (2014). *Question Answering with Subgraph Embeddings*. arXiv: 1406.3676 [cs.CL].

Bordes, A., Usunier, N., et al. (2013a). “Translating Embeddings for Modeling Multi-relational Data”. In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pp. 2787–2795.

— (2013b). “Translating Embeddings for Modeling Multi-relational Data”. In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pp. 2787–2795.

Bordes, A., Weston, J., and Usunier, N. (2014). *Open Question Answering with Weakly Supervised Embedding Models*. arXiv: 1404.4326 [cs.CL].

Bosselut, A., Bras, R. L., and Choi, Y. (2020). *Dynamic Neuro-Symbolic Knowledge Graph Construction for Zero-shot Commonsense Question Answering*. arXiv: 1911.03876 [cs.CL].

Bosselut, A., Rashkin, H., et al. (2019). *COMET: Commonsense Transformers for Automatic Knowledge Graph Construction*. arXiv: 1906.05317 [cs.CL].

Bradbury, S. P. (1995). “Quantitative structure-activity relationships and ecological risk assessment: an overview of predictive aquatic toxicology research”. In: *Toxicology Letters* vol. 79, no. 1. Decision Subtances Methodologies for Human Health Risk Assessment of Toxic Substances, pp. 229–237. ISSN: 0378-4274. DOI: https://doi.org/10.1016/0378-4274(95)03374-T. URL: https://www.sciencedirect.com/science/article/pii/037842749503374T.

Branco, P., Torgo, L., and Ribeiro, R. P. (2016). “A Survey of Predictive Modeling on Imbalanced Domains”. In: *ACM Comput. Surv.* vol. 49, no. 2, 31:1–31:50.

Breiman, L. (Oct. 2001). “Random Forests”. In: *Machine Learning* vol. 45, no. 1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

Breit, A. et al. (Apr. 2020). “OpenBioLink: a benchmarking framework for large-scale biomedical link prediction”. In: *Bioinformatics* vol. 36, no. 13, pp. 4097–4098. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa274. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/13/4097/33458979/btaa274.pdf. URL: https://doi.org/10.1093/bioinformatics/btaa274.

Bühmann, L., Lehmann, J., and Westphal, P. (2016). “DL-Learner - A framework for inductive learning on the Semantic Web”. In: *Web Semantics: Science, Services and Agents on the World Wide Web* vol. 39, pp. 15–24. ISSN: 1570-8268. DOI: http://dx.doi.org/10.1016/j.websem.2016.06.001. URL: http://www.sciencedirect.com/science/article/pii/S157082681630018X.

Busquet, F. et al. (Mar. 2020). "New European Union statistics on laboratory animal use – what really counts!" In: *ALTEX - Alternatives to animal experimentation* vol. 37, no. 2, pp. 167–186. DOI: 10.14573/altex.2003241. URL: https://www.altex.org/index.php/altex/article/view/1755.

Chary, M., Boyer, E. W., and Burns, M. M. (May 2021). "Diagnosis of Acute Poisoning using explainable artificial intelligence". en. In: *Comput Biol Med* vol. 134, p. 104469.

Chen, J., Jiménez-Ruiz, E., et al. (2021). "Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision". In: *European Semantic Web Conference (ESWC).* Springer.

Chen, J., Geng, Y., et al. (2021). *Low-resource Learning with Knowledge Graphs: A Comprehensive Survey.* arXiv: 2112.10006 [cs.LG].

Chen, J., Hu, P., Jimenez-Ruiz, E., et al. (2020). *OWL2Vec*: Embedding of OWL Ontologies.* arXiv: 2009.14654 [cs.AI].

Chen, J., Hu, P., Jiménez-Ruiz, E., et al. (2020). "OWL2Vec*: Embedding of OWL Ontologies". In: *CoRR* vol. abs/2009.14654. arXiv: 2009.14654. URL: https://arxiv.org/abs/2009.14654.

Chen, J., Lecue, F., et al. (2020). *Ontology-guided Semantic Composition for Zero-Shot Learning.* arXiv: 2006.16917 [cs.AI].

Chen, J., Lécué, F., et al. (2018). "Knowledge-based transfer learning explanation". In: *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning.*

Chen, Q. et al. (2021). "Zero-shot Text Classification via Knowledge Graph Embedding for Social Media Data". In: *IEEE Internet of Things Journal*, pp. 1–1. DOI: 10.1109/JIOT.2021.3093065.

Chen, X., Liu, M.-X., and Yan, G.-Y. (2012). "Drug–target interaction prediction by random walk on the heterogeneous network". In: *Mol. BioSyst.* vol. 8 (7), pp. 1970–1978. DOI: 10.1039/C2MB00002D. URL: http://dx.doi.org/10.1039/C2MB00002D.

Cherkassky, V. and Dhar, S. (2015). "Interpretation of Black-Box Predictive Models". In: *Measures of Complexity: Festschrift for Alexey Chervonenkis.* Ed. by Vovk, V., Papadopoulos, H., and Gammerman, A. Cham: Springer International Publishing, pp. 267–286. ISBN: 978-3-319-21852-6. DOI: 10.1007/978-3-319-21852-6_19. URL: https://doi.org/10.1007/978-3-319-21852-6_19.

Chollet, F. et al. (2015). *Keras.* https://github.com/fchollet/keras.

Coleman, T. F. and Moré, J. J. (1983). "Estimation of Sparse Jacobian Matrices and Graph Coloring Blems". In: *SIAM Journal on Numerical Analysis* vol. 20, no. 1, pp. 187–209. DOI: 10.1137/0720013. URL: https://doi.org/10.1137/0720013.

Connors, K. A. et al. (2019). "Creation of a Curated Aquatic Toxicology Database: EnviroTox". In: *Environmental Toxicology and Chemistry* vol. 38, no. 5, pp. 1062–1073. DOI: https://doi.org/10.1002/etc.4382. eprint: https://setac.onlinelibrary.wiley.com/doi/pdf/10.1002/etc.4382. URL: https://setac.onlinelibrary.wiley.com/doi/abs/10.1002/etc.4382.

Cortes, C. and Vapnik, V. (1995). “Support-Vector Networks”. In: *Machine Learning*, pp. 273–297.

Cover, T. M. and Hart, P. E. (1967). “Nearest neighbor pattern classification”. In: *IEEE Trans. Inf. Theory* vol. 13, pp. 21–27.

d’Amato, C., Masella, P., and Fanizzi, N. (2021). “An Approach Based on Semantic Similarity to Explaining Link Predictions on Knowledge Graphs”. In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. WI-IAT ’21. Melbourne, VIC, Australia: Association for Computing Machinery, pp. 170–177. ISBN: 9781450391153. DOI: 10.1145/3486622.3493956. URL: https://doi.org/10.1145/3486622.3493956.

Daston, G. et al. (Nov. 2014). “SEURAT: Safety Evaluation Ultimately Replacing Animal Testing—Recommendations for future research in the field of predictive toxicology”. In: *Archives of Toxicology* vol. 89, no. 1, pp. 15–23. DOI: 10.1007/s00204-014-1421-5. URL: https://doi.org/10.1007/s00204-014-1421-5.

David, J. et al. (2011). “The Alignment API 4.0”. In: *Semantic Web* vol. 2, no. 1, pp. 3–10.

Dettmers, T. et al. (Feb. 2018). “Convolutional 2D knowledge graph embeddings”. In: *AAAI 2018*.

Devlin, J. et al. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805.

— (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Doering, J. A. et al. (July 2018). “In Silico Site-Directed Mutagenesis Informs Species-Specific Predictions of Chemical Susceptibility Derived From the Sequence Alignment to Predict Across Species Susceptibility (SeqAPASS) Tool”. In: *Toxicological Sciences* vol. 166, no. 1, pp. 131–145. ISSN: 1096-6080. DOI: 10.1093/toxsci/kfy186. eprint: https://academic.oup.com/toxsci/article-pdf/166/1/131/26183887/kfy186.pdf. URL: https://doi.org/10.1093/toxsci/kfy186.

Dong, X. L. et al. (2014). “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*. Evgeniy Gabrilovich Wilko Horn Ni Lao Kevin Murphy Thomas Strohmann Shaohua Sun Wei Zhang Geremy Heitz, pp. 601–610. URL: http://www.cs.cmu.edu/~nlao/publication/2014.kdd.pdf.

Drucker, H. et al. (1996). “Support Vector Regression Machines”. In: *Advances in Neural Information Processing Systems*. Ed. by Mozer, M., Jordan, M., and Petsche, T. Vol. 9. MIT Press. URL: https://proceedings.neurips.cc/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf.

Duchi, J., Hazan, E., and Singer, Y. (July 2011). “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *J. Mach. Learn. Res.* vol. 12, pp. 2121–2159. ISSN: 1532-4435.

Dudek, A. Z., Arodz, T., and Gálvez, J. (2006). “Computational Methods in Developing Quantitative Structure-Activity Relationships (QSAR): A

Review". In: *Combinatorial Chemistry & High Throughput Screening* vol. 9, no. 3, pp. 213–228. ISSN: 1386-2073/1875-5402. DOI: 10.2174/138620706776055539.

Duffy, L. K., Dunlap, K. L., and Godduhn, A. R. (Sept. 2014). "Bias, complexity, and uncertainty in ecosystem risk assessment: pharmaceuticals, a new challenge in scale and perspective". In: *Environmental Research Letters* vol. 9, no. 9, p. 091004. DOI: 10.1088/1748-9326/9/9/091004. URL: https://doi.org/10.1088/1748-9326/9/9/091004.

Euzenat, J. and Shvaiko, P. (2013). *Ontology Matching, Second Edition.* Springer. ISBN: 978-3-642-38720-3.

Fan, Y. et al. (Mar. 2017). "DKGBuilder: An Architecture for Building a Domain Knowledge Graph from Scratch". In: pp. 663–667. DOI: 10.1007/978-3-319-55699-4_42.

Farazi, F. et al. (2020). "Knowledge Graph Approach to Combustion Chemistry and Interoperability". In: *ACS Omega* vol. 5, no. 29. PMID: 32743209, pp. 18342–18348. DOI: 10.1021/acsomega.0c02055. eprint: https://doi.org/10.1021/acsomega.0c02055. URL: https://doi.org/10.1021/acsomega.0c02055.

Faria, D., Jiménez-Ruiz, E., et al. (2014). "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*, pp. 17–32.

Faria, D., Pesquita, C., et al. (2013). "The AgreementMakerLight Ontology Matching System". In: *On the Move to Meaningful Internet Systems: OTM 2013 Conferences.* Ed. by Meersman, R. et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 527–541. ISBN: 978-3-642-41030-7.

Forfait-Dubuc, C. et al. (May 2012). "Survival data analyses in ecotoxicology: critical effect concentrations, methods and models. What should we use?" en. In: *Ecotoxicology* vol. 21, no. 4, pp. 1072–1083.

Fu, D. et al. (2019). "Enhancing Semantic Search of Crowdsourcing IT Services using Knowledge Graph". In: *SEKE.*

Fujita, T. and Winkler, D. A. (2016). "Understanding the Roles of the "Two QSARs"". In: *Journal of Chemical Information and Modeling* vol. 56, no. 2. PMID: 26754147, pp. 269–274. DOI: 10.1021/acs.jcim.5b00229. eprint: https://doi.org/10.1021/acs.jcim.5b00229. URL: https://doi.org/10.1021/acs.jcim.5b00229.

Fukuchi, J. et al. (Jan. 2019). "A practice of expert review by read-across using QSAR Toolbox". In: *Mutagenesis* vol. 34, no. 1, pp. 49–54. ISSN: 0267-8357. DOI: 10.1093/mutage/gey046. eprint: https://academic.oup.com/mutage/article-pdf/34/1/49/28015863/gey046.pdf. URL: https://doi.org/10.1093/mutage/gey046.

Futia, G. and Vetrò, A. (2020). "On the Integration of Knowledge Graphs into Deep Learning Models for a More Comprehensible AI—Three Challenges for Future Research". In: *Information* vol. 11, no. 2. ISSN: 2078-2489. DOI: 10.3390/info11020122. URL: https://www.mdpi.com/2078-2489/11/2/122.

Gad-Elrab, M. H. et al. (2020). "ExCut: Explainable Embedding-Based Clustering over Knowledge Graphs". In: *The Semantic Web – ISWC 2020.*

Ed. by Pan, J. Z. et al. Cham: Springer International Publishing, pp. 218–237. ISBN: 978-3-030-62419-4.

Geng, Y. et al. (2021). *OntoZSL: Ontology-enhanced Zero-shot Learning*. arXiv: 2102.07339 [cs.AI].

Geppert, H. et al. (2008). "Support-Vector-Machine-Based Ranking Significantly Improves the Effectiveness of Similarity Searching Using 2D Fingerprints and Multiple Reference Compounds". In: *Journal of Chemical Information and Modeling* vol. 48, no. 4. PMID: 18318473, pp. 742–746. DOI: 10.1021/ci700461s. eprint: https://doi.org/10.1021/ci700461s. URL: https://doi.org/10.1021/ci700461s.

Ghidini, C. et al., eds. (2019). *The Semantic Web – ISWC 2019*. Springer International Publishing. DOI: 10.1007/978-3-030-30796-7. URL: https://doi.org/10.1007/978-3-030-30796-7.

Gilpin, L. H. et al. (2019). *Explaining Explanations: An Overview of Interpretability of Machine Learning*. arXiv: 1806.00069 [cs.AI].

Glorot, X. et al. (2013). *A Semantic Matching Energy Function for Learning with Multi-relational Data*. arXiv: 1301.3485 [cs.LG].

Grant, R., Combs, A., and Acosta, D. (2010). "Experimental Models for the Investigation of Toxicological Mechanisms". In: Oxford: Elsevier. ISBN: 978-0-08-046884-6.

Grimm, F. A. et al. (2016). "A chemical–biological similarity-based grouping of complex substances as a prototype approach for evaluating chemical alternatives". In: *Green Chem.* vol. 18 (16), pp. 4407–4419. DOI: 10.1039/C6GC01147K. URL: http://dx.doi.org/10.1039/C6GC01147K.

Grover, A. and Leskovec, J. (2016). *node2vec: Scalable Feature Learning for Networks*. arXiv: 1607.00653 [cs.SI].

Guo, Q. et al. (2020). "A Survey on Knowledge Graph-Based Recommender Systems". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1. DOI: 10.1109/TKDE.2020.3028705.

Guo, S. et al. (Nov. 2016). "Jointly Embedding Knowledge Graphs and Logical Rules". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 192–202. DOI: 10.18653/v1/D16-1019. URL: https://aclanthology.org/D16-1019.

Hansch, C. et al. (Apr. 1962). "Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients". In: *Nature* vol. 194, no. 4824, pp. 178–180. ISSN: 1476-4687. DOI: 10.1038/194178b0. URL: https://doi.org/10.1038/194178b0.

Harrow, I. et al. (2017). "Matching disease and phenotype ontologies in the ontology alignment evaluation initiative". In: *J. Biomed. Semant.* vol. 8, no. 1, 55:1–55:13. DOI: 10.1186/s13326-017-0162-9. URL: https://doi.org/10.1186/s13326-017-0162-9.

Hasan, M. N. et al. (Aug. 2019). "Assessment of Drugs Toxicity and Associated Biomarker Genes Using Hierarchical Clustering". en. In: *Medicina (Kaunas)* vol. 55, no. 8.

Hastings, J., Dumontier, M., et al. (June 2010). “Representing chemicals using OWL, description graphs and rules”. In: URL: http://hdl.handle.net/10204/4919.

Hastings, J., Owen, G., et al. (2016). “ChEBI in 2016: Improved services and an expanding collection of metabolites”. In: *Nucleic acids research* vol. 44, no. D1, pp. 214–9.

Haussmann, S. et al. (Oct. 2019). “FoodKG: A Semantics-Driven Knowledge Graph for Food Recommendation”. In: pp. 146–162. ISBN: 978-3-030-30795-0. DOI: 10.1007/978-3-030-30796-7_10.

Hayashi, K. and Shimbo, M. (July 2017). “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)*. Vancouver, Canada: Association for Computational Linguistics, pp. 554–559. DOI: 10.18653/v1/P17-2088. URL: https://www.aclweb.org/anthology/P17-2088.

Hedderich, M. A. et al. (2021). *A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios*. arXiv: 2010.12309 [cs.CL].

Heller, S. R. et al. (2015). “InChI, the IUPAC International Chemical Identifier”. In: *Journal of Cheminformatics* vol. 7, no. 1, p. 23. ISSN: 1758-2946.

Hitzler, P., Krötzsch, M., and Rudolph, S. (2009). *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

Hoff, D. (Nov. 2020). “Utilizing automated and semi-automated data analytic tools for curating data in the ECOTOX Knowledgebase”. In: DOI: 10.23645/epacomptox.13256435.v1. URL: https://epa.figshare.com/articles/presentation/Utilizing_automated_and_semi-automated_data_analytic_tools_for_curating_data_in_the_ECOTOX_Knowledgebase/13256435.

Hoffmann, R. et al. (June 2011). “Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 541–550. URL: https://aclanthology.org/P11-1055.

Hogan, A. et al. (2020). “Knowledge Graphs”. In: *CoRR* vol. abs/2003.02320. URL: https://arxiv.org/abs/2003.02320.

Holter, O. M. et al. (2019). *Embedding OWL Ontologies with OWL2Vec*.

Holzinger, A., Biemann, C., et al. (2017). *What do we need to build explainable AI systems for the medical domain?* arXiv: 1712.09923 [cs.AI].

Holzinger, A., Kieseberg, P., et al. (Aug. 2018). “Current Advances, Trends and Challenges of Machine Learning and Knowledge Extraction: From Machine Learning to Explainable AI: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings”. In: pp. 1–8. ISBN: 978-3-319-99739-1. DOI: 10.1007/978-3-319-99740-7_1.

Hosoya, H. et al. (1999). “Topological Index and Thermodynamic Properties. 5. How Can We Explain the Topological Dependency of Thermodynamic Properties of Alkanes with the Topology of Graphs?” In: *Journal of Chemical*

Information and Computer Sciences vol. 39, no. 2, pp. 192–196. DOI: 10.1021/ci980058l. eprint: https://doi.org/10.1021/ci980058l. URL: https://doi.org/10.1021/ci980058l.

Huang, S., Wang, Y., and Yu, X. (Aug. 2020). “Design and Implementation of Oil and Gas Information on Intelligent Search Engine Based on Knowledge Graph”. In: *Journal of Physics: Conference Series* vol. 1621, p. 012010. DOI: 10.1088/1742-6596/1621/1/012010.

Jana, G. et al. (2020). “Quantitative structure-toxicity relationship: An “in silico study” using electrophilicity and hydrophobicity as descriptors”. In: *International Journal of Quantum Chemistry* vol. 120, no. 6, e26097. DOI: https://doi.org/10.1002/qua.26097.

Jiang, X. et al. (Dec. 2016). “Relation Extraction with Multi-instance Multi-label Convolutional Neural Networks”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 1471–1480. URL: https://aclanthology.org/C16-1139.

Jiménez-Ruiz, E. and Cuenca Grau, B. (2011). “LogMap: Logic-Based and Scalable Ontology Matching”. In: *10th International Semantic Web Conference*, pp. 273–288.

Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., et al. (2011). “Logic-based assessment of the compatibility of UMLS ontology sources”. In: *J. Biomedical Semantics* vol. 2, no. S-1, S2.

Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., et al. (2012). “Large-scale Interactive Ontology Matching: Algorithms and Implementation”. In: *the 20th European Conference on Artificial Intelligence (ECAI)*. Montpellier, France: IOS Press, pp. 444–449.

Jupp, S. et al. (Jan. 2014). “The EBI RDF platform: linked open data for the life sciences”. In: *Bioinformatics* vol. 30, no. 9, pp. 1338–1339. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt765. URL: https://doi.org/10.1093/bioinformatics/btt765.

Kadlec, R., Bajgar, O., and Kleindienst, J. (2017). “Knowledge Base Completion: Baselines Strike Back”. In: *CoRR* vol. abs/1705.10744. arXiv: 1705.10744. URL: http://arxiv.org/abs/1705.10744.

Kejriwal, M. (2019). *Domain-Specific Knowledge Graph Construction*. Springer.

Kiesling, E. et al. (Oct. 2019). “The SEPSES Knowledge Graph: An Integrated Resource for Cybersecurity”. In: pp. 198–214. ISBN: 978-3-030-30795-0. DOI: 10.1007/978-3-030-30796-7_13.

Kim, S., Bolton, E. E., and Bryant, S. H. (Nov. 2016). “Similar compounds versus similar conformers: complementarity between PubChem 2-D and 3-D neighboring sets”. In: *Journal of Cheminformatics* vol. 8, no. 1, p. 62.

Kim, S., Chen, J., et al. (Oct. 2018). “PubChem 2019 update: improved access to chemical data”. In: *Nucleic Acids Research* vol. 47, no. D1, pp. D1102–D1109. ISSN: 0305-1048.

Kingma, D. and Ba, J. (Dec. 2014). “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations*.

Kipf, T. N. and Welling, M. (2017). *Semi-Supervised Classification with Graph Convolutional Networks.* arXiv: 1609.02907 [cs.LG].

Kroetsch, M. and Weikum, G. (2016). *Journal of Web Semantics: Special Issue on Knowledge Graphs.* URL: http://www.websemanticsjournal.org/index.php/ps/announcement/view/19.

Kubinyi, H. (2002). "From Narcosis to Hyperspace: The History of QSAR". In: *Quantitative Structure-Activity Relationships* vol. 21, no. 4, pp. 348–356. DOI: https://doi.org/10.1002/1521-3838(200210)21:4<348::AID-QSAR348>3.0.CO;2-D. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/1521-3838\%28200210\%2921\%3A4\%3C348\%3A\%3AAID-QSAR348\%3E3.0.CO\%3B2-D. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/1521-3838%5C%28200210%5C%2921%5C%3A4%5C%3C348%5C%3A%5C%3AAID-QSAR348%5C%3E3.0.CO%5C%3B2-D.

Kulmanov, M. et al. (2019). "EL embeddings: geometric construction of models for the description logic EL++". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence.* AAAI Press, pp. 6103–6109.

LaLone, C. et al. (2014). "Sequence alignment to predict across-species susceptibility." In: *SETAC Europe, Basel, SWITZERLAND, May 11 - 15,*

Lalone, C. et al. (June 2016). "Sequence Alignment to Predict Across Species Susceptibility (SeqAPASS): A web-based tool for addressing the challenges of cross-species extrapolation of chemical toxicity". In: *Toxicological Sciences* vol. 153, kfw119. DOI: 10.1093/toxsci/kfw119.

Larras, F. et al. (Apr. 2022). "A critical review of effect modeling for ecological risk assessment of plant protection products". In: *Environmental Science and Pollution Research.* ISSN: 1614-7499. DOI: 10.1007/s11356-022-19111-3. URL: https://doi.org/10.1007/s11356-022-19111-3.

Lécué, F. and Wu, J. (2018). "Semantic Explanations of Predictions". In: *CoRR* vol. abs/1805.10587. arXiv: 1805.10587. URL: http://arxiv.org/abs/1805.10587.

Lehmann, J. et al. (2015). "DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia". In: *Semantic Web* vol. 6, no. 2, pp. 167–195.

Levenshtein, V. I. (Feb. 1966). "Binary Codes Capable of Correcting Deletions, Insertions and Reversals". In: *Soviet Physics Doklady* vol. 10, p. 707.

Li, Y. et al. (2020). "Domain Specific Knowledge Graphs as a Service to the Public: Powering Social-Impact Funding in the US". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining.* New York, NY, USA: Association for Computing Machinery, pp. 2793–2801. ISBN: 9781450379984. URL: https://doi.org/10.1145/3394486.3403330.

Liang, X. et al. (June 2019). "Predicting biomedical relationships using the knowledge and graph embedding cascade model". In: *PLOS ONE* vol. 14, no. 6, pp. 1–23. DOI: 10.1371/journal.pone.0218264. URL: https://doi.org/10.1371/journal.pone.0218264.

Liess, M. et al. (Sept. 2016). "Predicting the synergy of multiple stress effects". In: *Scientific Reports* vol. 6, no. 1, p. 32965. ISSN: 2045-2322. DOI: 10.1038/srep32965. URL: https://doi.org/10.1038/srep32965.

Lin, Y., Liu, Z., et al. (2015). “Learning Entity and Relation Embeddings for Knowledge Graph Completion”. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI’15. Austin, Texas: AAAI Press, pp. 2181–2187. ISBN: 0262511290.

Lin, Y., Mehta, S., et al. (Nov. 2017). “Drug target ontology to classify and integrate drug discovery data”. In: *Journal of Biomedical Semantics* vol. 8, no. 1, p. 50. ISSN: 2041-1480. DOI: 10.1186/s13326-017-0161-x. URL: https://doi.org/10.1186/s13326-017-0161-x.

Low, Y. et al. (Aug. 2013). “Integrative Chemical–Biological Read-Across Approach for Chemical Hazard Classification”. In: *Chemical Research in Toxicology* vol. 26, no. 8, pp. 1199–1208. DOI: 10.1021/tx400110f. URL: https://doi.org/10.1021/tx400110f.

Maaten, L. van der and Hinton, G. (2008). “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* vol. 9, no. 86, pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

Mahmoud, R. S. and Yousef, A. H. (2019). “Using Molecular Fingerprints as Descriptors in Toxicity Prediction: A Survey”. In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2649–2654. DOI: 10.1109/BIBM47256.2019.8982990.

Mikolov, T. et al. (2013a). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL].

Mikolov, T. et al. (2013b). “Efficient Estimation of Word Representations in Vector Space”. In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Bengio, Y. and LeCun, Y. URL: http://arxiv.org/abs/1301.3781.

Miller, G. A. (Nov. 1995). “WordNet: A Lexical Database for English”. In: *Commun. ACM* vol. 38, no. 11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: https://doi.org/10.1145/219717.219748.

Mintz, M. et al. (Aug. 2009). “Distant supervision for relation extraction without labeled data”. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, pp. 1003–1011. URL: https://aclanthology.org/P09-1113.

Mohamed, S. K., Nováček, V., and Nounu, A. (Aug. 2019). “Discovering protein drug targets using knowledge graph embeddings”. In: *Bioinformatics* vol. 36, no. 2, pp. 603–610. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz600. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/2/603/31962922/btz600.pdf. URL: https://doi.org/10.1093/bioinformatics/btz600.

Mohamed, S. et al. (2019). “Loss Functions in Knowledge Graph Embedding Models”. In: *Workshop on Deep Learning for Knowledge Graphs*.

Mukerjee, M. (2004). “Speaking for the Animals: A Veterinarian Analyzes the Turf Battles That Have Transformed the Animal Laboratory”. In: *Scientific American*.

Mumtaz, S. and Giese, M. (2021). "Hierarchy-based Semantic Embeddings for Single-valued & Multi-valued Categorical Variable". In: *Journal of Intelligent Information Systems.* (in press). ISSN: 0925-9902.

Muratov, E. N. et al. (2020). "QSAR without borders". In: *Chem. Soc. Rev.* vol. 49 (11), pp. 3525–3564. DOI: 10.1039/D0CS00098A. URL: http://dx.doi.org/10.1039/D0CS00098A.

Myklebust, E. B., Jiménez-Ruiz, E., et al. (2019a). "Integrating Semantic Technologies in Environmental Risk Assessment: A Vision". In: *29th Annual Meeting of the Society of Environmental Toxicology and Chemistry (SETAC).*

Myklebust, E. B., Jiménez-Ruiz, E., et al. (2019b). "Knowledge Graph Embedding for Ecotoxicological Effect Prediction". In: *Int'l Sem. Web Conf. (ISWC).* Best Student Paper in the In-Use track.

— (2020). "Ontology alignment in ecotoxicological effect prediction". In: *15th International Workshop on Ontology Matching.*

— (2022). "Prediction of adverse biological effects of chemicals using knowledge graph embeddings". In: *Semantic Web* vol. 13. 3, pp. 299–338. ISSN: 2210-4968. DOI: 10.3233/SW-222804. URL: https://doi.org/10.3233/SW-222804.

Myklebust, E. B., Jimenez-Ruiz, E., et al. (Nov. 2020). *Toxicological Effect and Risk Assessment (TERA) Knowledge Graph.* Version 1.1.0. (Version 1.1.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4244313. DOI: 10.5281/zenodo.4244313. URL: https://doi.org/10.5281/zenodo.4244313.

Nayak, A., Kesri, V., and Dubey, R. K. (2020). "Knowledge Graph Based Automated Generation of Test Cases in Software Engineering". In: *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD.* CoDS COMAD 2020. Hyderabad, India: Association for Computing Machinery, pp. 289–295. ISBN: 9781450377386. DOI: 10.1145/3371158.3371202. URL: https://doi.org/10.1145/3371158.3371202.

Nayyeri, M. et al. (2019). *Toward Understanding The Effect Of Loss function On Then Performance Of Knowledge Graph Embedding.* arXiv: 1909.00519 [cs.AI].

Nguyen, D. Q. et al. (2018). "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)*, pp. 327–333.

Nguyen, H. L., Vu, D. T., and Jung, J. J. (2020). "Knowledge graph fusion for smart systems: A Survey". In: *Information Fusion* vol. 61, pp. 56–70. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2020.03.014. URL: https://www.sciencedirect.com/science/article/pii/S1566253519307729.

Nickel, M., Rosasco, L., and Poggio, T. A. (2015). "Holographic Embeddings of Knowledge Graphs". In: *CoRR* vol. abs/1510.04935. arXiv: 1510.04935. URL: http://arxiv.org/abs/1510.04935.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). "A Three-way Model for Collective Learning on Multi-relational Data". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning.*

ICML'11. Bellevue, Washington, USA: Omnipress, pp. 809–816. ISBN: 978-1-4503-0619-5.

— (Apr. 2012). "Factorizing YAGO: Scalable Machine Learning for Linked Data". In: *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*, pp. 271–280. DOI: 10.1145/2187836.2187874.

Nikolova, N. and Jaworska, J. (2003). "Approaches to Measure Chemical Similarity – a Review". In: *QSAR & Combinatorial Science* vol. 22, no. 9-10, pp. 1006–1026.

NLM (2020). *Medical Subject Headings (MeSH) RDF.* https://id.nlm.nih.gov/mesh/.

Noy, N. et al. (July 2019). "Industry-Scale Knowledge Graphs: Lessons and Challenges". In: *Commun. ACM* vol. 62, no. 8, pp. 36–43. ISSN: 0001-0782. DOI: 10.1145/3331166. URL: https://doi.org/10.1145/3331166.

Obitko, M. (2007). *Semantic Web Architecture.* URL: https://obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html.

Olker, J. H. et al. (2022). "The ECOTOXicology Knowledgebase: A Curated Database of Ecologically Relevant Toxicity Tests to Support Environmental Research and Risk Assessment". In: *Environmental Toxicology and Chemistry* vol. 41, no. 6, pp. 1520–1539. DOI: https://doi.org/10.1002/etc.5324. eprint: https://setac.onlinelibrary.wiley.com/doi/pdf/10.1002/etc.5324. URL: https://setac.onlinelibrary.wiley.com/doi/abs/10.1002/etc.5324.

Parr, C. S. et al. (2014a). "The Encyclopedia of Life v2: Providing Global Access to Knowledge About Life on Earth". In: *Biodiversity Data Journal* vol. 2, e1079. ISSN: 1314-2836. DOI: 10.3897/BDJ.2.e1079.

— (2014b). "The Encyclopedia of Life v2: Providing Global Access to Knowledge About Life on Earth". In: *Biodiversity Data Journal* vol. 2, e1079. ISSN: 1314-2836. DOI: 10.3897/BDJ.2.e1079.

Parthasarathi, R. and Dhawan, A. (2018). "Chapter 5 - In Silico Approaches for Predictive Toxicology". In: *In Vitro Toxicology.* Ed. by Dhawan, A. and Kwon, S. Academic Press, pp. 91–109. ISBN: 978-0-12-804667-8. DOI: https://doi.org/10.1016/B978-0-12-804667-8.00005-5. URL: http://www.sciencedirect.com/science/article/pii/B9780128046678000055.

Paulheim, H. (2017). "Knowledge graph refinement: A survey of approaches and evaluation methods". In: *Semantic Web* vol. 8, pp. 489–508.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* vol. 12, pp. 2825–2830.

Peltason, L. and Bajorath, J. r. (2007). "SAR Index: Quantifying the Nature of Structure-Activity Relationships". In: *Journal of Medicinal Chemistry* vol. 50, no. 23. PMID: 17902636, pp. 5571–5578. DOI: 10.1021/jm0705713. eprint: https://doi.org/10.1021/jm0705713. URL: https://doi.org/10.1021/jm0705713.

Pernischova, R. et al. (Oct. 2019). "Toward Predicting Impact of Changes in Evolving Knowledge Graphs". In:

Plunkett, L. M., Kaplan, A. M., and Becker, R. A. (2015). "Challenges in using the ToxRefDB as a resource for toxicity prediction modeling". In: *Regulatory Toxicology and Pharmacology* vol. 72, no. 3, pp. 610–614. ISSN:

0273-2300. DOI: https://doi.org/10.1016/j.yrtph.2015.05.013. URL: https://www.sciencedirect.com/science/article/pii/S0273230015001129.

Pradeep, P. et al. (2016). “An ensemble model of QSAR tools for regulatory risk assessment”. In: *Journal of cheminformatics* vol. 8, pp. 48–48.

Pujara, J., Augustine, E., and Getoor, L. (Sept. 2017). “Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1751–1756. DOI: 10.18653/v1/D17-1184. URL: https://www.aclweb.org/anthology/D17-1184.

Radford, A. et al. (2019). “Language Models are Unsupervised Multitask Learners”. In:

Randić, M. (1997). “On Characterization of Chemical Structure”. In: *Journal of Chemical Information and Computer Sciences* vol. 37, no. 4, pp. 672–687. DOI: 10.1021/ci960174t. eprint: https://doi.org/10.1021/ci960174t. URL: https://doi.org/10.1021/ci960174t.

Reimers, N. and Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. DOI: 10.48550/ARXIV.1908.10084. URL: https://arxiv.org/abs/1908.10084.

Riedel, S., Yao, L., and McCallum, A. (2010). “Modeling Relations and Their Mentions without Labeled Text”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Balcázar, J. L. et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 148–163. ISBN: 978-3-642-15939-8.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (June 2013). “Relation Extraction with Matrix Factorization and Universal Schemas”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 74–84. URL: https://aclanthology.org/N13-1008.

Ristoski, P. and Paulheim, H. (2016). “RDF2Vec: RDF Graph Embeddings for Data Mining”. In: *International Semantic Web Conference*.

Ristoski, P., Rosati, J., et al. (2019). “RDF2Vec: RDF graph embeddings and their applications”. In: *Semantic Web* vol. 10, no. 4, pp. 721–752. DOI: 10.3233/SW-180317. URL: https://doi.org/10.3233/SW-180317.

Rogers, F. B. (Jan. 1963). “Medical subject headings”. en. In: *Bull Med Libr Assoc* vol. 51, no. 1, pp. 114–116.

Rohr, J. R., Salice, C. J., and Nisbet, R. M. (2016). “The pros and cons of ecological risk assessment based on data from different levels of biological organization”. In: *Critical Reviews in Toxicology* vol. 46, no. 9. PMID: 27340745, pp. 756–784. DOI: 10.1080/10408444.2016.1190685. eprint: https://doi.org/10.1080/10408444.2016.1190685. URL: https://doi.org/10.1080/10408444.2016.1190685.

Rossi, A., Barbosa, D., et al. (2021). “Knowledge Graph Embedding for Link Prediction: A Comparative Analysis”. In: *ACM Trans. Knowl. Discov. Data* vol. 15, no. 2, 14:1–14:49. DOI: 10.1145/3424672. URL: https://doi.org/10.1145/3424672.

Rossi, A., Firmani, D., et al. (2022). “Explaining Link Prediction Systems Based on Knowledge Graph Embeddings”. In: *Proceedings of the 2022 International Conference on Management of Data*. SIGMOD ’22. Philadelphia, PA, USA: Association for Computing Machinery, pp. 2062–2075. ISBN: 9781450392495. DOI: 10.1145/3514221.3517887. URL: https://doi.org/10.1145/3514221.3517887.

Rumelhart, D. E. and McClelland, J. L. (1987). “Learning Internal Representations by Error Propagation”. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pp. 318–362.

Russo, D. P. et al. (Feb. 2017). “CIIPro: a new read-across portal to fill data gaps using public large-scale chemical and biological data”. In: *Bioinformatics* vol. 33, no. 3, pp. 464–466.

Sakai, M. et al. (Jan. 2021). “Prediction of pharmacological activities from chemical structures with graph convolutional neural networks”. In: *Scientific Reports* vol. 11, no. 1, p. 525. ISSN: 2045-2322. DOI: 10.1038/s41598-020-80113-7. URL: https://doi.org/10.1038/s41598-020-80113-7.

Sani, M. (2020). “Knowledge Graph on Cybersecurity: A Survey”. In:

Sarker, M. K. et al. (2017). *Explaining Trained Neural Networks with Semantic Web Technologies: First Steps*. arXiv: 1710.04324 [cs.AI].

Sayers, E. W. et al. (Oct. 2008). “Database resources of the National Center for Biotechnology Information”. In: *Nucleic Acids Research* vol. 37, no. suppl_1, pp. D5–D15. ISSN: 0305-1048.

Schlichtkrull, M. et al. (2017). *Modeling Relational Data with Graph Convolutional Networks*. arXiv: 1703.06103 [stat.ML].

Scholkopf, B., Smola, A., and Müller, K.-R. (1999). “Kernel principal component analysis”. In: *Advances in Kernel Methods - Support Vector Learning*. MIT Press, pp. 327–352.

Seth Carbon, and et al. (Dec. 2020). “The Gene Ontology resource: enriching a GOld mine”. In: *Nucleic Acids Research* vol. 49, no. D1, pp. D325–D334. DOI: 10.1093/nar/gkaa1113. URL: https://doi.org/10.1093/nar/gkaa1113.

Sharma, A. K. et al. (Nov. 2017). “ToxiM: A Toxicity Prediction Tool for Small Molecules Developed Using Machine Learning and Chemoinformatics Approaches”. eng. In: *Frontiers in pharmacology* vol. 8, pp. 880–880. ISSN: 1663-9812. DOI: 10.3389/fphar.2017.00880. URL: https://doi.org/10.3389/fphar.2017.00880.

Shvaiko, P. and Euzenat, J. (2013). “Ontology Matching: State of the Art and Future Challenges”. In: *IEEE Trans. Knowl. Data Eng.* vol. 25, no. 1, pp. 158–176.

Skrindebakke, N. P. O. (2020). *Understanding the Role of Background Knowledge in Predictions*. Master’s thesis.

Smaili, F. Z., Gao, X., and Hoehndorf, R. (June 2018). “Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations”. In: *Bioinformatics* vol. 34, no. 13, pp. i52–i60. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty259. eprint: https://academic.oup.com/bioinformatics/article-pdf/34/13/i52/25098468/bty259.pdf. URL: https://doi.org/10.1093/bioinformatics/bty259.

Smaili, F. Z., Gao, X., and Hoehndorf, R. (2019). "Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction". In: *Bioinformatics* vol. 35, no. 12, pp. 2133–2140.

Socher, R. et al. (2013). "Reasoning With Neural Tensor Networks for Knowledge Base Completion". In: *Advances in Neural Information Processing Systems*. Ed. by Burges, C. J. C. et al. Vol. 26. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). "Yago: A Core of Semantic Knowledge". In: *16th International Conference on the World Wide Web*, pp. 697–706.

Sun, Z. et al. (2019a). "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HkgEQnRqYQ.

— (2019b). "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HkgEQnRqYQ.

Sushko, I. (2011). "Applicability Domain of QSAR models". In:

Swain, M. et al. (2014). *PubChemPy: Python wrapper for the PubChem PUG REST API.* [Online; accessed 15.08.2019]. URL: https://pubchempy.readthedocs.io/.

Tanimoto, T. (1958). *An Elementary Mathematical Theory of Classification and Prediction*. International Business Machines Corporation. URL: https://books.google.no/books?id=yp34HAAACAAJ.

Tiddi, I., Lécué, F., and Hitzler, P., eds. (2020). *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges*. Vol. 47. Studies on the Semantic Web. IOS Press. ISBN: 978-1-64368-080-4.

Tipping, M. E. and Bishop, C. M. (1999). "Probabilistic Principal Component Analysis". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* vol. 61, no. 3, pp. 611–622. ISSN: 13697412, 14679868. URL: http://www.jstor.org/stable/2680726.

Tollefsen, K. E. (2018). *NIVA Risk Assessment Database (RAdb)*. URL: www.niva.no/radb.

Trouillon, T. et al. (2016). "Complex Embeddings for Simple Link Prediction". In: *CoRR* vol. abs/1606.06357. arXiv: 1606.06357.

Tsakovska, I., Diukendjieva, A., and Worth, A. P. (2022). "In Silico Models for Predicting Acute Systemic Toxicity". en. In: *Methods Mol Biol* vol. 2425, pp. 259–289.

U.S. Environmental Protection Agency. (2020). *ToxCast & Tox21 Summary Files from invitrodb_v3*. URL: https://www.epa.gov/chemical-research/toxicity-forecaster-toxcasttm-data.

Universitetet i Bergen)", " (i realfag ved (n.d.). *Smådyr i ferskvann.* Accessed 11.06.2020. URL: https://www.miljolare.no/aktiviteter/vann/dammer/artslister/smadyr.

Van Leeuwen, C. J. (1995). "Ecotoxicological Effects". In: *Risk Assessment of Chemicals: An Introduction*. Ed. by Leeuwen, C. J. van and Hermens, J. L. M.

Dordrecht: Springer Netherlands, pp. 175–237. ISBN: 978-94-015-8520-0. DOI: 10.1007/978-94-015-8520-0_6. URL: https://doi.org/10.1007/978-94-015-8520-0_6.

Vilone, G. and Longo, L. (2020). *Explainable Artificial Intelligence: a Systematic Review.* arXiv: 2006.00093 [cs.AI].

— (2021). "Notions of explainability and evaluation approaches for explainable artificial intelligence". In: *Inf. Fusion* vol. 76, pp. 89–106. DOI: 10.1016/j.inffus.2021.05.009. URL: https://doi.org/10.1016/j.inffus.2021.05.009.

Vrandecic, D. and Krötzsch, M. (2014). "Wikidata: a free collaborative knowledgebase". In: *Commun. ACM* vol. 57, no. 10, pp. 78–85.

Waagmeester, A. et al. (Mar. 2020). "Wikidata as a knowledge graph for the life sciences". In: *eLife* vol. 9, e52614.

Wang, P. et al. (2015). *Explicit Knowledge-based Reasoning for Visual Question Answering.* arXiv: 1511.02570 [cs.CV].

Wang, Q. et al. (2017). "Knowledge Graph Embedding: A Survey of Approaches and Applications". In: *IEEE Trans. Knowl. Data Eng.* vol. 29, no. 12, pp. 2724–2743.

Wang, R.-L., Edwards, S., and Ives, C. (2019). "Ontology-based semantic mapping of chemical toxicities". In: *Toxicology* vol. 412, pp. 89–100. ISSN: 0300-483X. DOI: https://doi.org/10.1016/j.tox.2018.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0300483X18302920.

Wang, Z. et al. (June 2014). "Knowledge Graph Embedding by Translating on Hyperplanes". In: *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 28, no. 1. URL: https://ojs.aaai.org/index.php/AAAI/article/view/8870.

Watford, S. et al. (2019). "ToxRefDB version 2.0: Improved utility for predictive and retrospective toxicology analyses". In: *Reproductive Toxicology* vol. 89, pp. 145–158. ISSN: 0890-6238. DOI: https://doi.org/10.1016/j.reprotox.2019.07.012. URL: https://www.sciencedirect.com/science/article/pii/S0890623819300863.

Weaver, S. and Gleeson, M. P. (Jan. 2008). "The importance of the domain of applicability in QSAR modeling". en. In: *J Mol Graph Model* vol. 26, no. 8, pp. 1315–1326.

Wiener, H. (1947). "Structural Determination of Paraffin Boiling Points". In: *Journal of the American Chemical Society* vol. 69, no. 1. PMID: 20291038, pp. 17–20. DOI: 10.1021/ja01193a005. eprint: https://doi.org/10.1021/ja01193a005. URL: https://doi.org/10.1021/ja01193a005.

Willighagen, E. (2011). *InChIKey collision: the DIY copy/pastables.* URL: https://chem-bla-ics.blogspot.com/2011/09/inchikey-collision-diy-copypastables.html.

Wittwehr, C. et al. (2019). "Artificial Intelligence for Chemical Risk Assessment". In: *Computational Toxicology*, p. 100114. ISSN: 2468-1113. DOI: https://doi.org/10.1016/j.comtox.2019.100114. URL: http://www.sciencedirect.com/science/article/pii/S2468111319300349.

Wu, Y. and Wang, G. (Aug. 2018). “Machine Learning Based Toxicity Prediction: From Chemical Structural Description to Transcriptome Analysis”. In: *International journal of molecular sciences* vol. 19, no. 8, p. 2358.

Wu, Z. et al. (2016). “In silico prediction of chemical mechanism of action via an improved network-based inference method”. In: *British Journal of Pharmacology* vol. 173, no. 23, pp. 3372–3385. DOI: 10.1111/bph.13629. eprint: https://bpspubs.onlinelibrary.wiley.com/doi/pdf/10.1111/bph.13629. URL: https://bpspubs.onlinelibrary.wiley.com/doi/abs/10.1111/bph.13629.

Xian, Y. et al. (2017). “Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly”. In: *CoRR* vol. abs/1707.00600. arXiv: 1707.00600. URL: http://arxiv.org/abs/1707.00600.

Xu, C. et al. (2021). *Multiple Run Ensemble Learning with Low-Dimensional Knowledge Graph Embeddings*. arXiv: 2104.05003 [cs.AI].

Yang, B. et al. (2015). “Embedding Entities and Relations for Learning and Inference in Knowledge Bases”. In: *CoRR* vol. abs/1412.6575.

Yang, H. et al. (2018). “In silico prediction of chemical toxicity for drug design using machine learning methods and structural alerts”. In: *Frontiers in chemistry* vol. 6, p. 30.

Youden, W. J. (1950). “Index for rating diagnostic tests”. In: *Cancer* vol. 3, no. 1, pp. 32–35.

Yuan, J., Jin, Z., et al. (Jan. 2020). “Constructing biomedical domain-specific knowledge graph with minimum supervision”. In: *Knowledge and Information Systems* vol. 62, pp. 1–20. DOI: 10.1007/s10115-019-01351-4.

Yuan, J., Gao, N., and Xiang, J. (July 2019). “TransGate: Knowledge Graph Embedding with Shared Gate Structure”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 33, no. 01, pp. 3100–3107. DOI: 10.1609/aaai.v33i01.33013100. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4169.

Zhan, Q. and Yin, H. (2018). “A Loan Application Fraud Detection Method Based on Knowledge Graph and Neural Network”. In: *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence*. ICIAI ’18. Shanghai, China: Association for Computing Machinery, pp. 111–115. ISBN: 9781450363457. DOI: 10.1145/3194206.3194208. URL: https://doi.org/10.1145/3194206.3194208.

Zhang, F. et al. (2016). “Collaborative Knowledge Base Embedding for Recommender Systems”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, pp. 353–362. ISBN: 9781450342322. DOI: 10.1145/2939672.2939673. URL: https://doi.org/10.1145/2939672.2939673.

Zhang, W. et al. (Jan. 2019). “Interaction Embeddings for Prediction and Explanation in Knowledge Graphs”. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM. DOI: 10.1145/3289600.3291014. URL: https://doi.org/10.1145%5C%2F3289600.3291014.

Zhang, Z. et al. (2019a). *Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction.* arXiv: 1911.09419 [cs.LG].

— (2019b). *Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction.* arXiv: 1911.09419 [cs.LG].

Zhu, H. (2016). “Supporting read-across using biological data”. In: *ALTEX*, pp. 167–182. DOI: 10.14573/altex.1601252. URL: https://doi.org/10.14573/altex.1601252.

Zhu, Y. et al. (2021). *Faithfully Explainable Recommendation via Neural Logic Reasoning.* DOI: 10.48550/ARXIV.2104.07869. URL: https://arxiv.org/abs/2104.07869.

Zoph, B. et al. (2016). *Transfer Learning for Low-Resource Neural Machine Translation.* arXiv: 1604.02201 [cs.CL].

Papers

Paper I

Knowledge Graph Embedding for Ecotoxicological Effect Prediction

Erik B. Myklebust, Ernesto Jimenez-Ruiz, Jiaoyan Chen, Raoul Wolf, Knut Erik Tollefsen

Reprinted by permission from Springer Nature: Springer Nature, Lecture Notes in Computer Science book series vol. 11779, Knowledge Graph Embedding for Ecotoxicological Effect Prediction, Erik B. Myklebust; Ernesto Jimenez-Ruiz; Jiaoyan Chen; Raoul Wolf; Knut Erik Tollefsen; 2019.

Abstract

Exploring the effects a chemical compound has on a species takes a considerable experimental effort. Appropriate methods for estimating and suggesting new effects can dramatically reduce the work needed to be done by a laboratory. In this paper we explore the suitability of using a knowledge graph embedding approach for ecotoxicological effect prediction. A knowledge graph has been constructed from publicly available data sets, including a species taxonomy and chemical classification and similarity. The publicly available effect data is integrated to the knowledge graph using ontology alignment techniques. Our experimental results show that the knowledge graph based approach improves the selected baselines.

Contents

I.1 Introduction

Extending the scope of risk assessment models is a long-term goal in ecotoxicological research. However, biological effect data is only available for a few combinations of chemical-species pairs.[1] Thus, one of the main efforts in ecotoxicological research is the design of tools and methods to extrapolate from known to unknown combinations in order to facilitate risk assessment predictions on a population basis.

The Norwegian Institute for Water Research (NIVA) is a leading Norwegian institute for fundamental and applied research on marine and freshwaters.[2] The Ecotoxicology and Risk Assessment programme at NIVA has through the last years developed a risk assessment system called RAdb.[3] This system has been applied to several case studies based on agricultural/industrial runoff into lakes or fjords. However, the underlying relational database structure of RAdb has its limitations when dealing with the integration of diverse data and knowledge sources. This limitation is exacerbated when these resources do not share a common vocabulary, as it is the case in our ecotoxicology risk assessment setting.

In this paper we present a preliminary study of the benefits of using Semantic Web tools to integrate different data sources and knowledge graph embedding approaches to improve the ecotoxicological effect prediction. Hence, our contribution to the NIVA institute is twofold:

1. We have created a knowledge graph by gathering and integrating the relevant biological effect data and knowledge. Note that the format of the source data varies from tabular data, to SPARQL endpoints and ontologies. In order to discover equivalent entities we exploit internal resources, external resources (*e.g.*, Wikidata Vrandecic and Krötzsch 2014) and ontology alignment (*e.g.*, LogMap Jiménez-Ruiz, Cuenca Grau, Zhou, et al. 2012).

2. We have evaluated three knowledge graph embedding models (TransE (Bordes, Usunier, et al. 2013b), DistMult (B. Yang et al. 2015) and HolE (Nickel, Rosasco, and Poggio 2015)) together with the (baseline) prediction model currently used at NIVA. Our evaluation shows a considerable improvement with respect to the baseline and the benefits of using the knowledge graph models in terms of recall and $F_{\beta=2}$ score. Note that, in the NIVA use case, *false positives* are preferred over *false negatives* (*i.e.*, missing the hazard of a chemical over a species).

The rest of the paper is organised as follows. Section 2 provides some preliminaries to facilitate the understanding of the subsequent sections. In Section 3 we describe the use case where the knowledge graph and prediction models are applied. The creation of the knowledge graph is described in Section 4. Section 5 introduces the effect prediction models, while Section 6 presents the evaluation of these models. Finally, Section 7 elaborates on the contributions and discusses future directions of research.

[1] Chemical and compound are used interchangeably.

[2] NIVA Institute: https://www.niva.no/en

[3] NIVA Risk Assessment Database: https://www.niva.no/en/projectweb/radb

I.2 Preliminaries

Knowledge graphs. We follow the RDF-based notion of knowledge graphs (Arnaout and Elbassuoni 2018a) which are composed by RDF triples $\langle s, p, o\rangle$, where s represents a subject (a class or an instance), p represents a predicate (a property) and o represents an object (a class, an instance or a data value *e.g.*, text, date and number). RDF entities (*i.e.*, classes, properties and instances) are represented by an URI (Uniform Resource Identifier). A knowledge graph can be split into a TBox (terminology), often composed by RDF Schema constructors like class subsumption (*e.g.*, `ncbi:taxon/6668 rdfs:subClassOf ncbi:taxon/6657`) and property domain and range (`ecotox:affects rdfs:domain ecotox:Chemical`),[4] and an ABox (assertions), which contain relationships among instances (*e.g.*, `ecotox:chemical/330541 ecotox:affects ecotox:effect/202`) and semantic type definitions (*e.g.*, `ecotox:taxon/28868 rdf:type ecotox:Taxon`). RDF-based knowledge graphs can be accessed with SPARQL queries, the standard language to query RDF graphs.

Ontology alignment. Ontology alignment is the process of finding mappings or correspondences between a source and a target ontology or knowledge graph (Euzenat and Shvaiko 2013). These mappings are typically represented as equivalences among the entities of the input resources (*e.g.*, `ncbi:taxon/13402 owl:sameAs ecotox:taxon/Carya`).

Embedding models. Knowledge graph embedding (Q. Wang et al. 2017) plays a key role in link prediction problems where the goal is to learn a scoring function $S : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$. $S(s, p, o)$ is proportional to the probability that a triple $\langle s, p, o\rangle$ is encoded as true. Several models have been proposed, *e.g.*, Translating embeddings model (TransE) (Bordes, Usunier, et al. 2013b). These models are applied to knowledge graphs to resolve missing facts in largely connected knowledge graphs, such as DBpedia (Lehmann et al. 2015). Embedding models have also been successfully applied in biomedical link prediction tasks (*e.g.*, Agibetov and Samwald 2018; Alshahrani et al. 2017).

Evaluation metrics. We use (A)ccuracy, (P)recision, (R)ecall, (F_β) score to evaluate the models. They are defined as

$$A = \frac{tp + tn}{tp + tn + fp + fn} \tag{I.1}$$

$$P = \frac{tp}{tp + fp} \tag{I.2}$$

$$R = \frac{tp}{tp + fn} \tag{I.3}$$

$$F_\beta = (1 + \beta^2)\frac{PR}{\beta^2 P + R} \tag{I.4}$$

[4] The OWL 2 ontology language provides more expressive constructors. Note that the graph projection of an OWL 2 ontology can be seen as a knowledge graph (*e.g.*, Agibetov, Jiménez-Ruiz, et al. 2018).

where tp, tn, fp, and fn stand for *true positive*, *true negative*, *false positive*, and *false negative*, respectively. Essentially, accuracy is the proportion of correct classifications. Recall is a measure of how many expected positive predictions were found by our model, and precision is the proportion of predictions that were correctly classified. F_β is a combined measure of precision and recall. $\beta = 1$ gives equal weight, while $\beta < 1$ favours precision and $\beta > 1$ favours recall. Here we use $F_{\beta=1}$ (F_1 in short) and $F_{\beta=2}$.

As the above metrics all depend on a selected threshold, we also use area under the receiver operating characteristic (ROC) curve (AUC) to measure and compare the overall pattern recognition capability of the prediction models. ROC is the curve of true positive rate ($\frac{tp}{(tp+fn)}$, i.e., recall) and false positive rate ($\frac{fp}{(fp+tn)}$), with the threshold ranging from 0 to 1 using a small step. AUC is the area under this curve, its values range between 0 and 1. Larger AUC indicates higher performance.

I.3 NIVA use case: ecotoxicology and risk assessment

Ecotoxicology is a multidisciplinary field that studies the ecological and toxicological effects of chemical pollutants on populations, communities and ecosystems. Risk assessment is the result of the intrinsic hazards of a substance combined with an estimate of the environmental exposure (*i.e.*, Hazard + Exposure = Risk).

The Computational Toxicology Program within NIVA's Ecotoxicology and Risk Assessment section aims at designing and developing prediction models to assess the effect of chemical mixtures over a population where traditional laboratory data cannot be easily acquired.

Figure I.1 shows the risk assessment pipeline followed at NIVA. *Exposure* is data gathered from the environment, while *effects* are hypothesis that are tested in a laboratory. These two data sources are used to calculate risk, which is used to find (further) susceptible species and the mode of action (MoA) or type of impact a compound would have over those species. Results from the MoA analysis are used as new effect hypothesis.

The effect data is gathered during experiments in a laboratory, where the population of a single species is exposed to a concentration of a toxic compound. Most commonly, the mortality rate of the population is measured at each time interval until it becomes a constant. Although the mortality at each time interval is referred to as *endpoint* in the ecotoxicology literature, we use *outcome* of the experiment to avoid confusion. Table I.1 shows the typical outcomes and their proportion within the effects data. To give a good indication of the toxicity to a species, these experiments need to be repeated with increasing concentrations until the mortality reaches 100%. However, this is time consuming and is generally not done (*sola dosis facit venenum*). Hence, some compounds may appear more toxic than others due to limited experiments. Thus, when evaluating prediction models, (higher values of) recall are preferred over precision.

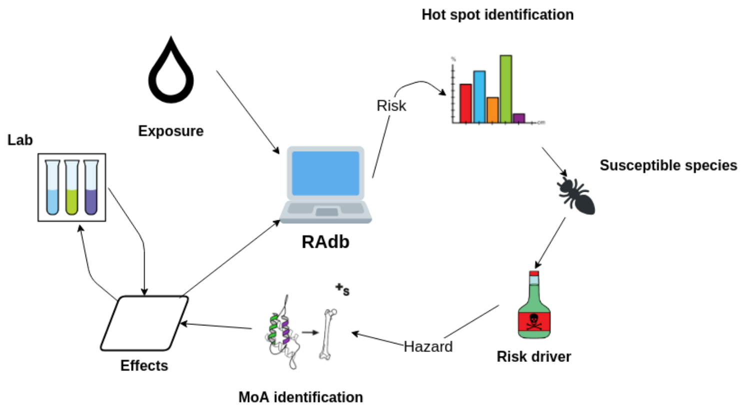


Figure I.1: NIVA risk assessment pipeline.

Proportion	Abbreviation	Description
0.21	NR	Not reported
0.17	NOEL	No-observable-effect-level
0.16	LC50	Lethal concentration for 50% of test population
0.14	LOEL	Lowest-observable-effect-level
0.05	NOEC	No-observable-effect-concentration
0.05	EC50	Effective concentration for 50% of test population
0.04	LOEC	Lowest observable effect concentration
0.03	BCF	Bioconcentration factor
0.02	NR-LETH	Lethal to 100% of test population
0.02	LD50	Lethal dose for 50% of test population
0.11	Other	

Table I.1: The 10 most frequent outcomes in ECOTOX effect data.

Risk assessment methods require large amounts of effect data to efficiently predict long term risk for the ecosystems. The data must cover a minimum of the chemicals found when analysing water samples from the ecosystem, along with covering species present in the ecosystem. This leads to a immense search space that is close to impossible to encompass in its entirety. Thus, it is essential to extrapolate from known to unknown combinations of chemical-species and suggest to the lab (ranked) effect hypothesis. The state-of-the-art within effect prediction are quantitative structure–activity relationship models (QSARs). These models have shown promising results for use in risk assessment, *e.g.*, Pradeep et al. 2016. However, QSARs have limitations with regard the coverage of compounds and species. These models use some chemical properties, but they usually only consider one or few species at a time. In this work we contribute with an alternative approach based on knowledge graph embeddings where the knowledge graph provides a global and integrated view of the domain.

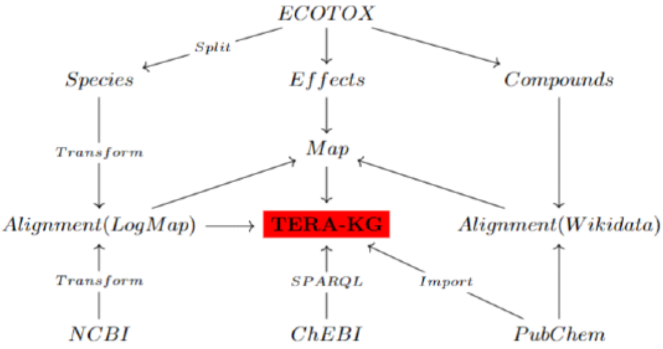


Figure I.2: Data sources in the TERA knowledge graph. Compound classification is available from PubChem. Chemical class hierarchy comes from the ChEMBL SPARQL endpoint. Compound literals are gathered from PubChem REST API and transformed into triples. ECOTOX and PubChem identifiers are aligned using the Wikidata SPARQL endpoint. ECOTOX and NCBI taxonomies are aligned using LogMap.

test_id	reference_number	test_cas	species_number	result_id	test_id	endpoint	conc1_mean	conc1_unit
1068553	5390	877430 (2,6-Dimethylquinoline)	5156 (Danio rerio)	98004	1068553	$LC50$	400	mg/kg diet
2037887	848	79061 (2-Propenamide)	14 (Rasbora heteromorpha)	2063723	2037887	$LC10$	220	mg/L

Table I.2: ECOTOX database entry examples.

Currently, the NIVA RAdb is under redevelopment, giving opportunities to include sophisticated effect prediction approaches, like the one presented in this paper, as a novel module for improving domain wide regulatory risk assessment.

I.4 A knowledge graph for toxicological effect data

Risk assessment involves different data sources and laboratory experiments as shown in Figure I.1. In this section we describe the relevant datasets and their integration to create the *Toxicological Effects and Risk Assessment* (TERA) knowledge graph (see Figure I.2).

I.4.1 The ECOTOX database

We rely on the ECOTOXicology database (ECOTOX) (Olker et al. 2022). ECOTOX consists of $\sim 930k$ tests (or experiments) derived from the literature. Currently, an ECOTOX test considers the effect of one of $\sim 12k$ chemicals on one of $\sim 13k$ species. Which implies that less than 1% of compound-species pairs have been tested. The effect is categorised in one of a plethora of predefined outcomes. For example, the $LC50$ outcome implies lethal concentration for 50% of the test population. Table I.1 shows the most frequent outcomes in ECOTOX.

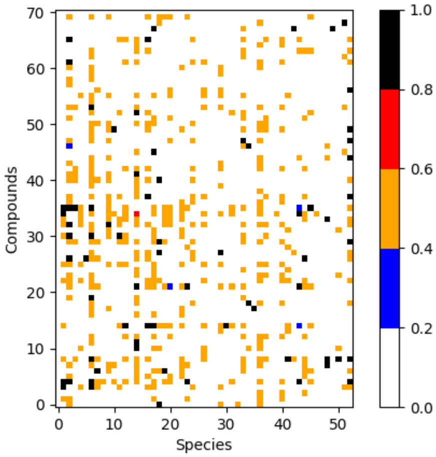


Figure I.3: ECOTOX effects data. x and y-axis represent individual species and chemicals sorted by similarity. Similarities are given by Equations (I.6) and (I.7) in Section I.5.1. *i.e.*, chemicals $c_i \in C$ are indexed such that $S_{0,1} > S_{1,2} > \cdots > S_{n-1,n}$. Showing only chemicals and species that are involved in 25 or more experiments. Values relate to mortality rate of the test population, *i.e.*, LC50 corresponds to 0.5.

Table I.2 contains an excerpt of the ECOTOX database. ECOTOX includes information about the compounds and species used in the tests. This information, however, is limited and additional (external) resources are required to complement ECOTOX.

The number of outcomes per compound and species varies substantially. For example, there are 1,881 experiments where the compound used is *sulfuric acid*, and 9,436 experiments where *Pimephales promelas* (fathead minnow) is the test species. The median number of experiments per chemical and species are 3 and 6, respectively. Figure I.3 visualises a subset of the outcomes, here the zero values are either no effect or missing. This figure shows certain features of the data, *e.g.*, that compounds are more diversely used than species and that compound similarity is closely correlated to effects with regards to a species.

Currently, the ECOTOX database in used in risk assessment as reference data when calculating risk for a ecosystem. Essentially, comparing the reference and the observed chemical concentrations (per species). Since most compounds have multiple experiments per species, the mean and standard deviation of risk to a species can be calculated. However, if there is only one experiment for a compound-species pair we cannot calculate a standard deviation, such that the risk assessment is featureless. Therefore, estimating new effects is important to represent the natural variability of the effect data.

#	subject	predicate	object
(i)	ecotox:group/Worms	owl:disjointWith	ecotox:group/Fish
(ii)	ncbi:division/2	owl:disjointWith	ncbi:division/4
(iii)	ecotox:taxon/34010	rdfs:subClassOf	ecotox:taxon/hirta
(iv)	ncbi:taxon/687295	rdfs:subClassOf	ncbi:taxon/513583
(v)	compound:CID10198308	rdf:type	obo:CHEBI_134899
(vi)	compound:CID10198308	pubchem:formula	“$C_7H_6O_6S$”
(vii)	ecotox:chemical/115866	ecotox:affects	ecotox:effect/001
(viii)	ecotox:effect/001	ecotox:species	ecotox:taxon/26812
(ix)	ecotox:effect/001	ecotox:endpoint	LC50
(x)	ecotox:taxon/33155	owl:sameAs	ncbi:taxon/311871

Table I.3: Example triples from the TERA knowledge graph

I.4.2 Dataset integration into the TERA knowledge graph

Figure I.2 shows the different datasets and their transformation that contribute in the creation of the TERA knowledge graph. For example Triples *(vii)*-*(ix)* in Table I.3 have been created from the ECOTOX effect data.

Each compound in the ECOTOX effect data has a identifier called CAS Registry Number assigned by the Chemical Abstracts Service. The CAS numbers are proprietary, however, Wikidata (Vrandecic and Krötzsch 2014) (indirectly) encodes mappings between CAS numbers and open identifiers like *InChIKey*, a 27 character hash of the International Chemical Identifier (InChI) that encodes the chemical information in a unique manner. Hence, other datasets, such as PubChem (Kim, J. Chen, et al. 2018), can be used to gather chemical features and classification of compounds. PubChem is already available as a knowledge graph and can be imported directly. However, the PubChem hierarchy only contains permutations of compounds. To create a full taxonomy for the chemical data, we use the ChEMBL SPARQL endpoint to extract the classification (provided by the ChEBI ontology (Hastings, Owen, et al. 2016)) for the relevant PubChem compounds. For example Triples *(v)* and *(vi)* in Table I.3 come from the integration with PubChem and ChEMBL.

Aligning ECOTOX and NCBI. The species lineage in ECOTOX is not complete and therefore this (missing) information has been complemented with the NCBI taxonomy (Sayers et al. 2008), a curated classification of all of the organisms in the public sequence databases (around 10% of the species on Earth). The tabular data provided for the ECOTOX species and the NCBI taxonomies has been transformed into subsumptions and disjointness triples (see first four triples in Table I.3). Leaf nodes are treated as instance entities.

Since there does not exist a complete and public alignment between ECOTOX species and the NCBI Taxonomy, we have used the LogMap (Jiménez-Ruiz and Cuenca Grau 2011; Jiménez-Ruiz, Cuenca Grau, Zhou, et al. 2012) ontology alignment systems to index and align the ECOTOX and NCBI vocabularies. ECOTOX currently only provides a subset of the mappings via its web search interface. We have gathered a total of 929 ground truth mappings for validation purposes. The lexical indexation provided by LogMap left us with 5,472 possible NCBI entities to map to ECOTOX (we focus only on instances, *i.e.*, leaf nodes).

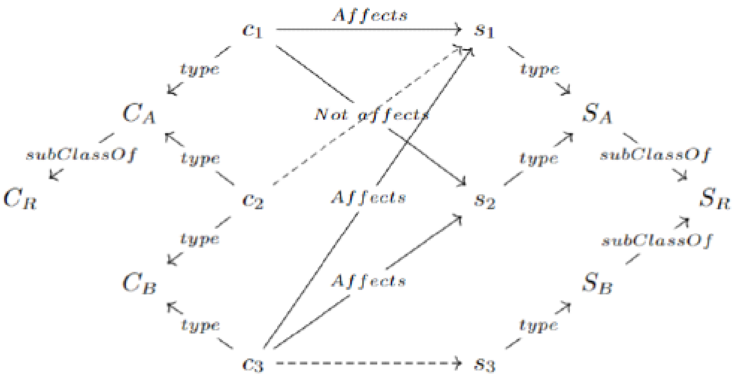


Figure I.4: The effect prediction problem. Lowercase s_j and c_i are instances of species and compounds, while uppercase denote classes in the hierarchy. Solid lines are observations and dashed lines are to be predicted. *i.e.*, does c_2 affect s_1?

LogMap identified 4,681 (instance) mappings to ECOTOX ($\sim$ 40% of its entities) covering all 929 mappings from the (incomplete) ground truth, thus, an estimated recall of 100%. The mappings computed by LogMap have been included to the TERA knowledge graph as additional equivalence triples (see Triple *(x)* in Table I.3 as example).

I.5 Effect prediction models

In this section we introduce the selected machine learning models to solve the effect prediction problem shown in Figure I.4. We use the known effects, denoted as *Affects* and *Not affects* in the figure, to predict whether or not new proposed chemical-species pairs are *true* (Affects) or *false* (Not affects).[5]

Effect data sampling. A balance between positive and negative effect data samples is desired, therefore, we choose outcomes in categories (refer to Table I.1): NOEL, LCp, LDp, NR-LETH, and NR-ZERO (p ranges from 0 to 100). We are only concerned about the mortality rate in experiments, consequently, we treat LC* and LD* identically. In addition, NR-LETH is treated as LC100. For simplicity, we treat the effects as binary entities. Hence, the outcome for a compound-species pair c, s is defined as

$$f(c,s) = \begin{cases} 1 & \text{if } (c,s) \in \text{LCp} \cup \text{LDp} \cup \text{NR-LETH} \\ 0 & \text{if } (c,s) \in \text{NOEL} \cup \text{NR-ZERO}. \end{cases} \tag{I.5}$$

[5] The models are implemented with Keras (Chollet et al. 2015). Data and codes available from: https://github.com/Erik-BM/NIVAUC

For example, according to Figure I.4, $f(c_1, s_1) = 1$ (*i.e.*, c_1 affects s_1) and $f(c_1, s_2) = 0$ (*i.e.*, c_1 does not affects s_1), while $f(c_2, s_1)$ is unknown and thus a prediction is required for this chemical-species pair.

Knowledge graphs. We rely on the TERA knowledge graph (see excerpts in Table I.3 and Figure I.4) to feed the knowledge graph embedding algorithms. For simplicity we discard the ECOTOX species entities that have not a correspondence to NCBI. Note that we currently do not consider literals.

I.5.1 Baseline model (M_1)

This (baseline) prediction model is based on the current prediction method used at NIVA. The basic idea of this method is to find the nearest-neighbour from the observed samples. In this context, the nearest neighbours are defined by hierarchy distance for species and similarity for compounds. Therefore, we first define a adjacency matrix for the taxonomy and a similarity matrix for compounds.

$$A_{i,j} = \frac{1}{|P(s_i, r)| + |P(s_j, r)| - 2|P(s_i, r) \cap P(s_j, r)| + 1} \tag{I.6}$$

where r is the taxonomy root, $P(x, r)$ is the classes in the path from x to r, and $|\cdot|$ denotes the cardinality. One basic approach to calculate the chemical similarity is using the Jaccard index of the binary fingerprints of the compounds (Nikolova and Jaworska 2003). Hence, the similarity matrix is defined as

$$S_{i,j} = J(c_i, c_j) = \frac{|(F_i)_2 \cap (F_j)_2|}{|(F_i)_2 \cup (F_j)_2|} \tag{I.7}$$

We define a matrix $E \in \mathbb{R}^{|C| \times |T|}$, where C and T denote the set of compounds and species respectively. E contains all the observed effects (training set):

$$E_{i,j} = \begin{cases} 1 & \text{if } (c_i,\ \text{affects},\ s_j) \\ 0 & \text{else} \end{cases} \tag{I.8}$$

We can then make the prediction with A, S, and E , as shown in Algorithm I.5. The algorithm terminates when t_{max} neighbours are visited or $p > 0$.

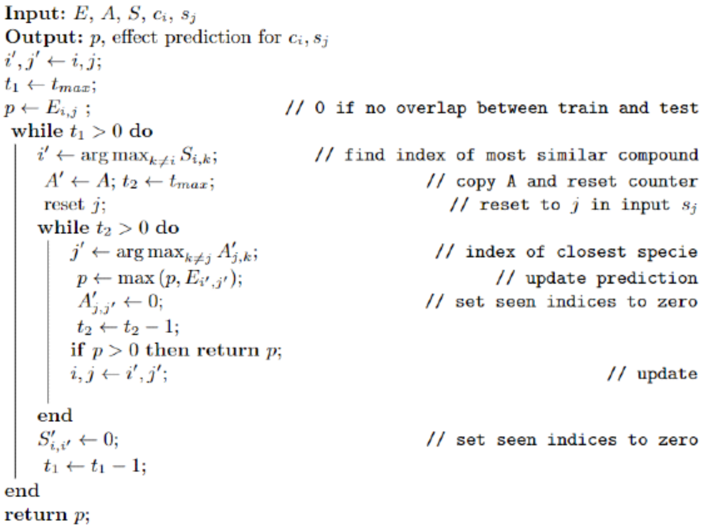
I.5.2 Multilayer perceptron (M_2)

Our second prediction model is a Multilayer perceptron (MLP) network with n hidden layers. The model can be expressed as:

$$\mathbf{y}^0 = [\mathbf{e}_c, \mathbf{e}_s] \tag{I.9}$$

$$\mathbf{y}^t = \mathit{ReLu}(\mathbf{y}^{t-1} W_t + \mathbf{b}_t) \tag{I.10}$$

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



```
Input: $E$, $A$, $S$, $c_i$, $s_j$
Output: $p$, effect prediction for $c_i, s_j$
$i', j' \leftarrow i, j$;
$t_1 \leftarrow t_{max}$;
$p \leftarrow E_{i,j}$ ;                          // 0 if no overlap between train and test
while $t_1 > 0$ do
    $i' \leftarrow \arg\max_{k \neq i} S_{i,k}$;        // find index of most similar compound
    $A' \leftarrow A$; $t_2 \leftarrow t_{max}$;                  // copy A and reset counter
    reset $j$;                                    // reset to $j$ in input $s_j$
    while $t_2 > 0$ do
        $j' \leftarrow \arg\max_{k \neq j} A'_{j,k}$;             // index of closest specie
        $p \leftarrow \max(p, E_{i',j'})$;                   // update prediction
        $A'_{j,j'} \leftarrow 0$;                         // set seen indices to zero
        $t_2 \leftarrow t_2 - 1$;
        if $p > 0$ then return $p$;
        $i, j \leftarrow i', j'$;                                    // update
    end
    $S'_{i,i'} \leftarrow 0$;                             // set seen indices to zero
    $t_1 \leftarrow t_1 - 1$;
end
return $p$;
```

Figure I.5: Baseline prediction model algorithm (M_1).

where $t = 1, 2, ..., n$. $[\cdot, \cdot]$ denotes vector concatenation. $ReLu$ is the rectifier function and σ is the logistic sigmoid function. W_t are the weight matrices and b_t are the biasses for each layer. $\mathbf{e}_c, \mathbf{e}_s \in \mathbb{R}^k$ are the embedded vectors of c and s. For example $\mathbf{e}_c$ is defined as

$$\mathbf{e}_c = \delta_c W_C \tag{I.12}$$

where δ_c is the one-hot encoded vector for entity c, $W_C \in \mathbb{R}^{|C| \times k}$ is an embedding transformation matrix to learn.

A dropout layer is stacked after each hidden layer to prevent the network from overfitting. The model is optimised using ADAGRAD (Duchi, Hazan, and Singer 2011) with the following log loss function:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)] \tag{I.13}$$

I.5.3 Knowledge graph (KG) embedding and MLP ($M_2^\star$)

We have extended the MLP model (M_2) by feeding it with the TERA KG-based embeddings of c (*i.e.*, the chemical) and s (*i.e.*, the species), which encode the information of the taxonomy and compound hierarchies, among other semantic relationships. Note that the TERA knowledge graph also includes similarity triples about compounds. These triples represent pairs of compounds c_i and c_j where their similarity $S_{i,j}$ (as in Equation I.7) is above a threshold ϕ.

The embeddings are learned by applying the scoring function from one of DistMult (B. Yang et al. 2015), HolE (Nickel, Rosasco, and Poggio 2015), and

TransE (Bordes, Usunier, et al. 2013b). TransE was selected as it provides a very intuitive model. DistMult was included as it has shown state-of-the-art performance (*e.g.*, Kadlec, Bajgar, and Kleindienst 2017), while HolE was considered as it also encodes directional relations. The score function for DistMult is defined as

$$S_D(s,p,o) = \sigma(\mathbf{e}_s^T W_p \mathbf{e}_o),\ W_p = diag(\mathbf{e}_p) \tag{I.14}$$

HolE uses a circular correlation score function, defined by

$$S_H(s,p,o) = \sigma(\mathbf{e}_r^T[\mathbf{e}_s \star \mathbf{e}_o]),\ \mathbf{e}_s \star \mathbf{e}_o = \mathcal{F}^{-1}[\overline{\mathcal{F}(\mathbf{e}_s)} \odot \mathcal{F}(\mathbf{e}_o)] \tag{I.15}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier transform and its inverse, $\overline{x}$ is the elementwise complex conjugate, $\odot$ denotes the Hadamard product. The final method is TransE, which has the score function

$$S_T(s,p,o) = ||\mathbf{e}_s + \mathbf{e}_p - \mathbf{e}_o|| \tag{I.16}$$

where $||\mathbf{x}||$ is the norm of $\mathbf{x}$. $\mathbf{e}_s$, $\mathbf{e}_p$ and $\mathbf{e}_o$ are the vector representation for the subject, predicate and object of a triple, respectively.

DistMult and HolE optimises for a score of 1 for positive samples and 0 for negative samples. Moreover, TransE scores positive samples as 0 and with no upper bound for negative samples. We modify the TransE score function to $S'_T = \tanh(1/S_T)$, such that $\lim_{S_T \to 0} S'_T = 1$ and $\lim_{S_T \to \infty} S'_T = 0$, to avoid modifying the labels.

The embeddings are used in the same network as the M_2 model. We train the embeddings and the classifier simultaneously using log loss and `ADAGRAD`. Training simultaneously will optimise the embeddings with regards to both the knowledge graph triples and the classifier loss.

I.6 Effect prediction evaluation

Sampling. We split the effect data 50%/50% for train/test. To prevent test set leakage, those training inputs that appear in the test set are removed, resulting in a 70%/30% split. $M_2^\star$ can be trained with the entirety of the knowledge graph, which is ignored under effect prediction. The negative knowledge graph samples are generated by randomly re-sampling subject and object of a true sample, while maintaining the distribution of predicates. We generate four negative samples per positive sample.

$\mathbf{M_1}$ model settings. We tested the performance of M_1 with 6 choices of nearest neighbour (5, 10, 20, 30, 40, 50). In addition to Algorithm I.5, we tested an alternative technique for iterating over the data. However, Algorithm I.5 yielded better results. The most balanced results were found when using 30 neighbours. When using more than 30 neighbours recall increases, but accuracy and precision suffer from a considerable decrease since the use of more neighbours increases the false positive rate.

	M_1 ($t_{max} = 30$)	M_2	$M_2^\star$ (S'_T)	$M_2^\star$ (S_D)	$M_2^\star$ (S_H)
Accuracy	0.58	0.82	**0.83**	**0.83**	**0.83**
Precision	0.47	**0.76**	0.75	**0.76**	0.73
Recall	0.80	0.78	0.84	0.82	**0.87**
F_1 score	0.59	0.77	**0.79**	**0.79**	**0.79**
$F_{\beta=2}$ score	0.70	0.78	0.82	0.81	**0.84**
AUC	–	0.90	**0.91**	**0.91**	**0.91**
Accuracy	0.56 ± 0.01	$\mathbf{0.81 \pm 0.02}$	$\mathbf{0.81 \pm 0.02}$	$\mathbf{0.81 \pm 0.01}$	$\mathbf{0.81 \pm 0.02}$
Precision	0.55 ± 0.01	0.79 ± 0.04	$\mathbf{0.80 \pm 0.04}$	0.78 ± 0.03	0.79 ± 0.03
Recall	0.76 ± 0.03	0.84 ± 0.08	0.83 ± 0.08	$\mathbf{0.87 \pm 0.05}$	0.86 ± 0.02
F_1 score	0.65 ± 0.01	0.81 ± 0.03	0.81 ± 0.03	$\mathbf{0.82 \pm 0.01}$	0.82 ± 0.01
$F_{\beta=2}$ score	0.72 ± 0.02	0.83 ± 0.06	0.82 ± 0.06	$\mathbf{0.85 \pm 0.03}$	0.84 ± 0.01
AUC	–	$\mathbf{0.89 \pm 0.01}$	0.88 ± 0.01	$\mathbf{0.89 \pm 0.01}$	$\mathbf{0.89 \pm 0.02}$

Table I.4: Performance of the prediction models. $M_2^\star$ (S'_T), $M_2^\star$ (S_D) and $M_2^\star$ (S_H) stand for the MLP prediction models using TransE, DistMult, and HolE embedding models, respectively. *Above line*: ensemble averages of 10 clean tests. *Below line*: 10 fold cross validation on training set with standard deviation.

$\mathbf{M_2/M_2^\star}$ model settings. The embedding dimension used in M_2 and $M_2^\star$ was based on a search among sizes 16, 64, 128 and 256. We found no difference between these parameters for M_2, therefore, 16 is chosen to aid faster training. $M_2^\star$ used a larger amount of entities and needs a larger embedding space to capture the features of the data. The performance plateaued at 128, hence, this was chosen. The models (M_2,$M_2^\star$) were trained until the loss stops improving for 5 iterations. For $M_2^\star$ we used different loss weights for the embeddings and the effect predictor. These weights were chosen such that the embeddings and effects are learned at similar rates. DistMult and HolE used 0.5 and 1.0 as loss weights for embeddings and effects models, respectively, while TransE used equal weights. We used a dropout rate of 0.2 and a similarity threshold of 0.5. Note that in $M_2^\star$ we simultaneously train the embedding models and the effect predictor. We perform

1. 10 fold cross validation on the training set, and
2. a clean test on the unseen test set. This test consist of a ensemble of 10 models trained on the training set, each with a new set of random negative knowledge graph samples. We used an ensemble to limit the impact the random negative samples has on the results.

Evaluation. Figures I.6a and I.6b and Table I.4 show the results of the conducted evaluation for the five effect prediction models. Figures I.6a and I.6b visualise the impact on accuracy and recall with different thresholds on the M_2-$M_2^\star$ prediction scores, while Table I.4 presents the relevant evaluation metrics with a threshold of 0.5 for M_2-$M_2^\star$ and 30 neighbours for M_1. The results can be summarised as follows:

1. M_1 is only slightly better than random choice, as the prior binary output distribution is 0.59 and 0.41. Thus it would not be appropriate for predicting effects. The false positive rate is also high, hence, M_1 would not be practical to use as a recommendation system.

2. M_2 is considerably better than M_1 and has balance between precision and recall. We suspect that this balance is due to random choice when the model has not previously seen a chemical or species. *i.e.*, a prediction close to the decision boundary when an input is unseen will maintain the false negative/positive proportion, hence good for accuracy, not necessary for giving (interesting) recommendations to the laboratory.

3. Introducing the background knowledge to M_2, in the form of KG embeddings gives higher recall, without loosing accuracy. In contrast to M_2, $M_2^\star$ is more uncertain when unseen combinations are presented to the model (*in dubio pro reo*). Therefore, $M_2^\star$ is better suited to giving recommendations for cases where there is limited information about the chemical and the species in the effect data.

4. The best results in terms of recall, when using a threshold of 0.5 (see Table I.4), are obtained by $M_2^\star$ with the embeddings provided by HolE (9 points higher than the M_2).

5. As shown in Figures I.6a and I.6b, lowering the decision threshold (0.30) would yield a higher recall (0.90) for the DistMult-based model, while maintaining the accuracy. TransE and HolE-based models have higher recall (0.97 and 0.94) at decision threshold 0.30, however, this comes at a cost of reduction in accuracy (0.74 and 0.79).

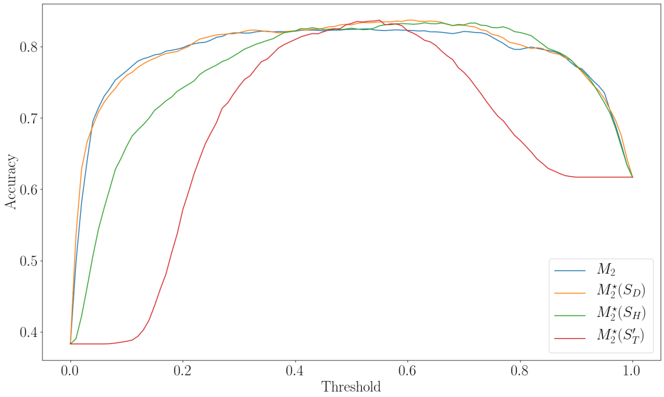
6. The highest overall $F_{\beta=2}$ score is 0.87, and is shared by all $M_2^\star$ models, albeit, at different decision boundaries, 0.34, 0.14 and 0.31 for models with TransE, DistMult, and HolE embeddings, respectively.

I.7 Discussion and future work

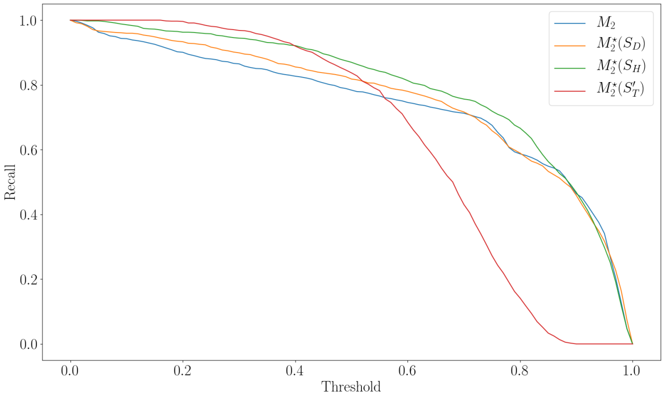
We have created a knowledge graph called TERA that aims at covering the knowledge and data relevant to the ecotoxicological domain. We have also implemented a proof-of-concept prototype for ecotoxicological effect prediction based on knowledge graph embeddings. The obtained results are encouraging, showing the positive impact of using knowledge graph embedding models and the benefits of having an integrated view of the different knowledge and data sources.

Knowledge graph. The TERA knowledge graph is by itself an important contribution to NIVA. TERA integrates different knowledge and data sources and aims at providing an unified view of the information relevant to the ecotoxicology and risk assessment domain. At the same time the adoption of a RDF-based knowledge graph enables the use of

1. an extensive range of Semantic Web infrastructure that is currently available (*e.g.*, reasoning engines, ontology alignment systems, SPARQL query engines), and



(a) Accuracy for the M_2 and $M_2^\star$ prediction models.



(b) Recall for the M_2 and $M_2^\star$ prediction models.

Figure I.6: Accuracy and Recall for the M_2 and $M_2^\star$ models with various thresholds.

2. state of the art knowledge graph embedding strategies.

Prediction models. The obtained predictions are promising and show the validity of the selected models in our setting and the benefits of using the TERA knowledge graph. As mentioned before, we favour recall with respect to precision. One the one hand, false positives are not necessarily harmful, while overlooking the hazard of a chemical may have important consequences. On the other hand, due to the limited experiments in terms of concentration (*i.e.*, effect data may not be complete), some chemicals may look less toxic than others while they may still be hazardous.

Value for NIVA. The conducted work falls into one of the main research lines of NIVA's Computational Toxicology Program (NCTP) to enhance the generation of hypothesis to be tested in the laboratory (Myklebust, Jiménez-Ruiz, et al. 2019a). Furthermore, the data integration efforts and the construction of the TERA knowledge graph also goes in line with the vision of NIVA's section for Environmental Data Science. The availability and accessibility of the best knowledge and data will enable optimal decision making.

Novelty. Knowledge graph embedding models have been applied in general purpose link discovery and knowledge graph completion tasks (Q. Wang et al. 2017). They have also attracted the attention in the biomedical domain to find, for example, candidate genes for a disease, protein-protein interactions or drug-target interactions (*e.g.*, Agibetov and Samwald 2018; Alshahrani et al. 2017). However, we are not aware of the application of knowledge graph embedding models in the context of toxicological effect prediction.

Future work. The main goal in the mid-term future is to integrate the TERA knowledge graph and the machine learning based prediction models within NIVA's risk assessment pipeline. In the near future, we intend to improve the current ecotoxicological effect prediction prototype and evaluate the suitability of more sophisticated models like Graph Convolutional Networks. The TERA knowledge graph will also be extended with additional information about species (*e.g.*, interactions) and compounds (*e.g.*, target proteins) which is expected to enhance the computed embeddings and the effect predictions.

Resources. The datasets, evaluation results, documentation and source codes are available from the following GitHub repository: https://github.com/NIVA-Knowledge-Graph/NIVAUC

Acknowledgements. This work is supported by the grant 272414 from the Research Council of Norway (RCN), the MixRisk project (RCN 268294), the AIDA project, The Alan Turing Institute under the EPSRC grant EP/N510129/1, the SIRIUS Centre for Scalable Data Access (RCN 237889), the Royal Society, EPSRC projects DBOnto, MaSI^3 and ED^3. We would also like to thank Martin Giese and Zofia C. Rudjord for their contribution in early stages of this project.

Paper II

Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings

Erik B. Myklebust, Ernesto Jimenez-Ruiz, Jiaoyan Chen, Raoul Wolf, Knut Erik Tollefsen

Reprinted from Semantic Web, 13, Erik B. Myklebust; Ernesto Jimenez-Ruiz; Jiaoyan Chen ; Raoul Wolf; Knut Erik Tollefsen, Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings, 299-338, 2022, with permission from IOS Press. The publication is available at IOS Press through https://doi.org/10.3233/SW-222804.

Abstract

We have created a knowledge graph based on major data sources used in ecotoxicological risk assessment. 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.

Contents

II.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 disparate data sources, formats and vocabularies. The use of Semantic Web technologies and (RDF-based) knowledge graphs (Arnaout and Elbassuoni 2018b) 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.

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:

1. 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 endpoints over public linked data. We have exploited external resources (*e.g.*, Wikidata (Vrandecic and Krötzsch 2014)) and ontology alignment methods (*e.g.*, LogMap (Jiménez-Ruiz, Cuenca Grau, Zhou, et al. 2012)) to discover equivalences between the data sources.

2. 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. This is used to tailor the embeddings to this specific task.

3. 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 (Myklebust, Jimenez-Ruiz, et al. 2020) (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 (Myklebust, Jiménez-Ruiz, et al. 2019b). We have

1. extended TERA with new sources (Encyclopedia of Life (EOL), MeSH, and a larger part of ChEMBL) and provided detailed steps about its creation;
2. created a more robust prediction model with nine (up from three) embedding algorithms supported and a task-specific embedding fine-tuning strategy; and
3. 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 II.2 introduces essential concepts to the subsequent sections. Section II.3 introduces the use case where the knowledge graph and prediction models are applied. Section II.4 introduces related work. The creation of the knowledge graph is described in Section II.5. Section II.6 introduces the prediction models, while Section II.7 presents the evaluation of these models. Section II.8 elaborates on the contributions and discusses future directions of research. Finally, Appendix II.A gives an overview of the knowledge graph embedding models used in this work.

II.2 Preliminaries

In this section we introduce important background concepts that will be used throughout the paper. Table II.1 contain the most important symbols.

Symbol	Definition
RDF	Resource Description Framework
OWL	Web Ontology Language
SPARQL	SPARQL Protocol and RDF Query Language
KG	Knowledge graph
KGE	Knowledge graph embedding
t	A triple
sb	The subject of a triple
ob	The object of a triple
p, r	The predicate/relation of a triple
e	A KG entity
$\mathcal{T}$	The set of KG triples
$\mathcal{E}$	The set of KG entities
$\mathcal{R}$	The set of KG relations
$\mathcal{L}$	The set of literal values
$\mathbf{e}$	The vector representation of an entity or relation
k	The dimension of a vector
SF	The scoring function of a KGE model
PT	Pre-trained KGE-based model
FT	Fine-tuning KGE-based model
s	A species
c	A chemical
S	Refers to species
C	Refers to chemicals
κ	Chemical concentration

Table II.1: Key symbols and acronyms used throughout the paper.

II.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 several endpoints, *e.g.*, LC50 at 48 hours and LC100 at 96 hours (lethal concentration for all test organisms). See Table II.2 for the most common endpoints.

[1] Not to be confused with SPARQL endpoint.

II.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) (Hogan et al. 2020). 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 $\langle sb, p, ob \rangle \in \mathcal{E} \times \mathcal{R} \times \mathcal{E} \cup \mathcal{L}$,[3] where sb represents a subject (a class or an instance), p represents a predicate (a property) and ob represents an object (a class, an instance or a literal). KG entities (*i.e.*, $\mathcal{E} \cup \mathcal{R}$: 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.[4] The ABox contains assertions among instances, including OWL equality and inequality, and semantic type definitions. Table II.5 shows several examples of TBox and ABox triples.

II.2.3 Ontology alignment

Ontology alignment is the process of finding mappings or correspondences between a source and a target ontology or knowledge graph (Euzenat and Shvaiko 2013; Shvaiko and Euzenat 2013). These mappings typically represent equivalences or broader/narrower relationships among the entities of the input ontologies. In the ontology matching community (Abd Nikooie Pour et al. 2020), mappings are exchanged using the RDF Alignment format (David et al. 2011); but they can also be interpreted as standard OWL axioms (*e.g.*, Faria, Jiménez-Ruiz, et al. 2014; Jiménez-Ruiz, Cuenca Grau, Horrocks, et al. 2011). In this work we treat ontology alignments as OWL axioms (*e.g.*, Triple t_{13} in Table II.5). An ontology matching system (*e.g.*, LogMap (Jiménez-Ruiz and Cuenca Grau 2011)) is a program that, given as input two ontologies or knowledge graphs, generates as output a set of mappings (*i.e.*, an alignment) M.

II.2.4 Embedding models

Knowledge graph embedding (KGE) (Rossi, Barbosa, et al. 2021; Q. Wang et al. 2017) 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 (Lehmann et al. 2015). Biomedical link prediction is another

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

[3] $\mathcal{E}$ is the set of all classes and instances, $\mathcal{R}$ is the set of all properties, while $\mathcal{L}$ represents the set of all literal values.

[4] Note that the Web Ontology Language (OWL) (B. Cuenca Grau et al. 2008) 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/

area where embedding models have been applied successfully (*e.g.*, Agibetov and Samwald 2020; Alshahrani et al. 2017).

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,[5] *i.e.*, $SF : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$, $SF \propto P(\langle sb, p, ob \rangle \in KG)$; 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 $\langle sb, p, ob \rangle$ into a vector representation $\mathbf{e}_{sb}, \mathbf{e}_p, \mathbf{e}_{ob} \in \mathbb{R}^k$ or $\mathbb{C}^k$, where k is the dimension of the vector; *(ii)* apply a scoring function to $(\mathbf{e}_{sb}, \mathbf{e}_p, \mathbf{e}_{ob})$; 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.

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

Figure II.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

[5] For the embedding process, we focus on triples where $o \in \mathcal{E}$ is a class or an instance.

[6] The interested reader please refer to (Rossi, Barbosa, et al. 2021) for a comprehensive survey.

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

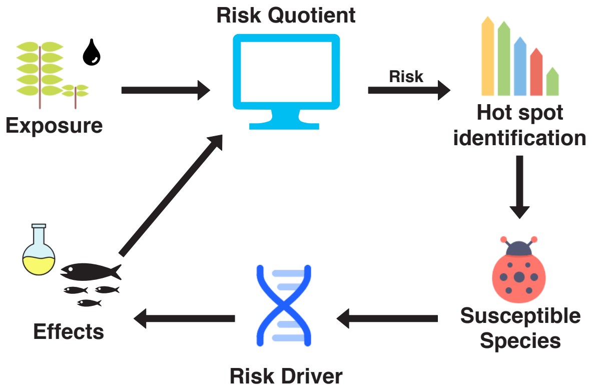


Figure II.1: Simplified ecological risk assessment pipeline.

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 II.2: The most frequent endpoints in ECOTOX (Olker et al. 2022) chemical effect data.

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 II.2).

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

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

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 (Universitetet i Bergen)" n.d.). 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:

1. highly specialized (restricted in chemical and species domains) models to predict chemical concentrations that will have an effect on a test species, and

2. 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 II.4.1.

II.4 Related Work

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

II.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 (*c.f.*, reviews Dudek, Arodz, and Gálvez 2006; Fukuchi et al. 2019). 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 2018 take this further by using the logarithm of chemical concentration to achieve a polynomial relationship: $\log(1/\kappa) = f(\pi) + g(\sigma)$, $f \in P_2$ and $g \in P_1$ (P_n is a polynomial of nth degree), where κ is the chemical concentration while π and σ denote the derived chemical features hydrophobicity[9] and electronic effects in the molecule, respectively. 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.

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

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 an example of such an approach (Doering et al. 2018; LaLone et al. 2014). SeqAPASS uses a large amount of data available for humans, mice, rats, and zebrafish to extrapolate to areas with lower coverage.

II.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 II.A. The interested reader please refer to Rossi, Barbosa, et al. 2021 for a comprehensive survey.

The three categories of models are decomposition, geometric, and convolutional (Rossi, Barbosa, et al. 2021). 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 (B. Yang et al. 2015). ComplEx uses the same scoring function as DistMult, however, in a complex vector space, such that it can handle inverse relations (Trouillon et al. 2016). HolE is based on holographic embeddings (Nickel, Rosasco, and Poggio 2015), however, it has been shown that HolE is equivalent to ComplEx (Hayashi and Shimbo 2017).

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 (Bordes, Usunier, et al. 2013b). RotatE is similar to TransE, however, the translation using the predicate is done by rotating it (via Euler's identity) (Sun et al. 2019b). Furthermore, pRotatE is a baseline for RotatE where the modulus in Euler's identity is ignored (Sun et al. 2019b). 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 (Z. Zhang et al. 2019b).

The convolutional models take a deep learning approach to the task of KGE. We use ConvKB (D. Q. Nguyen et al. 2018) and ConvE (Dettmers et al. 2018), 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 (Rossi, Barbosa, et al. 2021). 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.

II.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.*, Agibetov and Samwald 2020; Alshahrani et al. 2017; Liang et al. 2019. X. Chen, M.-X. Liu, and Yan 2012 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 (Z. Wu et al. 2016).

Opa2vec (Smaili, X. Gao, and Hoehndorf 2019) and Blagec et al. 2019 have developed embedding models to improve similarity-based prediction in the biomedical domain, while OpenBioLink (Breit et al. 2020) has created a framework for evaluating models in the biomedical domain.

EL Embeddings (Kulmanov et al. 2019) and Opa2vec (Smaili, X. Gao, and Hoehndorf 2019) 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* (J. Chen, Hu, Jiménez-Ruiz, et al. 2020) 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 (J. Chen, Jiménez-Ruiz, et al. 2021).

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 KGE models in this new domain.

II.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), an ontology-enhanced RDF-based 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 II.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 II.5.4).

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

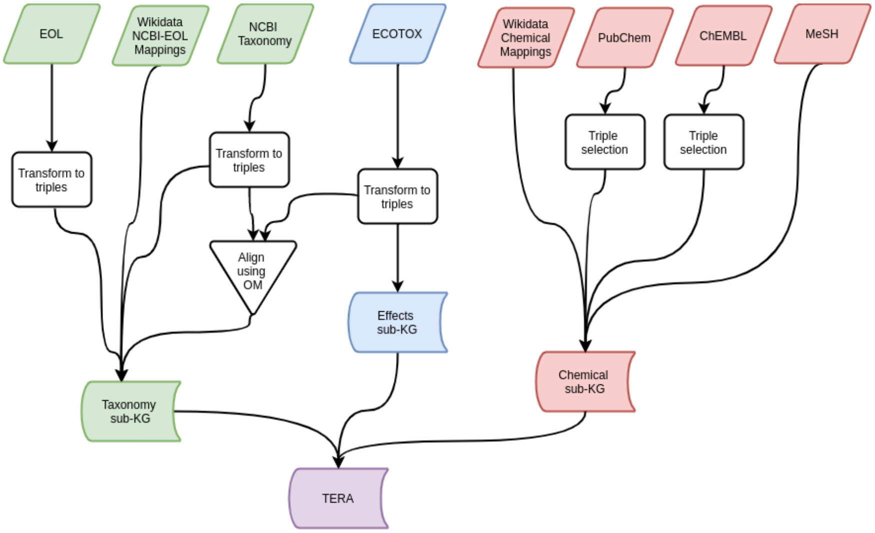


Figure II.2: Data sources and processes to create the TERA knowledge graph.

The data sources integrated into TERA vary from tabular and RDF files to SPARQL endpoints over public linked data. The sources currently integrated into TERA are:

1. biological: NCBI Taxonomy, Encyclopedia of Life, and Wikidata mappings ($\sim 500k$ species);
2. chemical: PubChem, ChEMBL, MeSH, and Wikidata mappings ($\sim 110M$ compounds); and
3. biological effects: ECOTOXicology Knowledgebase ($\sim 1M$ results, $\sim 12k$ compounds, $\sim 13k$ species), and system-generated mappings.

These three distinct parts make up the sub-KGs of TERA, *i.e.*,

1. the Taxonomy sub-KG (KG_S),
2. the Chemical sub-KG (KG_C), and
3. the Effects sub-KG (KG_E).

The different processes to transform and integrate these sources into TERA are shown in Figure II.2.

A snapshot of TERA is available on Zenodo (Myklebust, Jimenez-Ruiz, et al. 2020), where licenses permit.[11] PubChem and ChEMBL are not included in the

[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.

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

Table II.3: ECOTOX database tests example.

result_id	test_id	endpoint	effect	conc1_mean	conc1_unit
102570	1147366	$LC50$	MOR	110000	$\mu g/L$

Table II.4: ECOTOX database results example.

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 in our GitHub repository.[14] Table II.5 shows several examples of RDF triples from TERA.[15]

II.5.1 Dataset overview

TERA, as mentioned above, 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 (Olker et al. 2022). 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 $\sim 0.6\%$. The resulting endpoint from an experiment is categorised in one of a plethora of predefined endpoints (see Table II.2 above).

Tables II.3 and II.4 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.

[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.

[16] Version dated Sep. 15, 2020.

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 (Vrandecic and Krötzsch 2014) (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 (Heller et al. 2015).[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 (Sayers et al. 2008). This taxonomy data source consists of a number of database dump files, which contains a hierarchy for all sequenced species, 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) (Parr et al. 2014a), 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.

II.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 II.5 shows an excerpt of the triples in TERA.

[17] While InChI is unique, InChiKey is not, and collisions have greater than zero probability (Willighagen 2011).

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

#	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:endpoint/LC50
t_5	et:result/102570	et:effect	et:effect/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 II.5: 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.

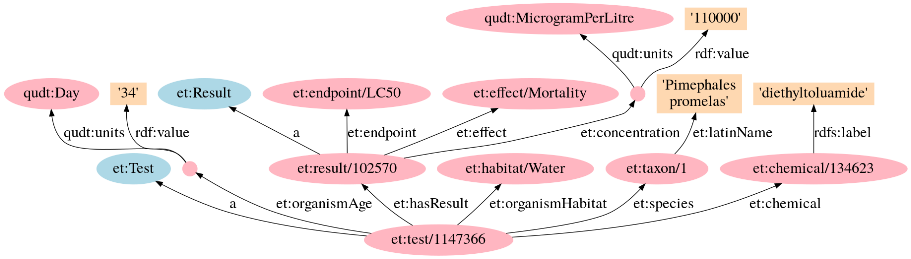


Figure II.3: Example of an ECOTOX test and related triples.

II.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 II.3 and II.4, 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 II.5 (see Triples t_1 - t_{12}). A graphical representation for an effect test and its result is also shown in Figure II.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 II.5.2.2).

1. 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 I.3). 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.

2. 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 levels in the taxonomy is consistent (see triples t_6 and t_7 in Table I.3). Note that when adding triples such as t_{11} in Table I.3, we also add a taxonomic rank to facilitate the querying for a specific taxonomic level.

3. 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 I.3) and a (chemical) ECOTOX group.

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

```
@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.

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.

II.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:

1. LogMap (Jiménez-Ruiz and Cuenca Grau 2011; Jiménez-Ruiz, Cuenca Grau, Zhou, et al. 2012),
2. AgreementMakerLight (AML) (Faria, Pesquita, et al. 2013), and
3. a string matching algorithm based on Levenshtein distance (Levenshtein 1966).

LogMap and AML were chosen since they have performed well across many datasets in the Ontology Alignment Evaluation Initiative (*e.g.*, Abd Nikooie Pour et al. 2020; Algergawy, Cheatham, et al. 2018; Algergawy, Faria, et al. 2019). Most mappings in our setting are expected to be lexical, therefore, we

[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.

Method	1-to-1 mappings		
	# M	R	$P^{\approx}$
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	$\mathbf{21,145}$	$\mathbf{0.83}$	$\mathbf{0.86}$

Table II.6: Alignment results for ECOTOX-NCBI. #M: number of mappings (at instance level), R: Recall, $P^{\approx}$: estimated precision.

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 II.6 shows the alignment results over the ground truth samples for the 1-to-1 (filtered) system mappings. We report number of mappings (#M), Recall (R) and estimated precision ($P^{\approx}$) with respect to the known entities in the incomplete ground truth, assuming only 1-to-1 mappings are valid. $P^{\approx}$ is calculated as

$$P^{\approx} = |M^{\approx} \cap M_{ref}|/|M^{\approx}|, \tag{II.1}$$

$$M^{\approx} = \{\langle e_e, \texttt{owl:sameAs}, e_n\rangle \in M \mid e_e \in \mathcal{E}_e^{ref} \vee e_n \in \mathcal{E}_n^{ref}\}, \tag{II.2}$$

where M_{ref} is the (incomplete) reference mapping set and M is the set of generated mappings between entities $e_e \in \mathcal{E}_e$ from ECOTOX and entities $e_n \in \mathcal{E}_n$ from the NCBI Taxonomy, $\mathcal{E}_e^{ref} \subseteq \mathcal{E}_e$ and $\mathcal{E}_n^{ref} \subseteq \mathcal{E}_n$ are the sets of entities that appear in the reference mappings. Thus, $M^{\approx}$ 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

$$R = |M \cap M_{ref}|/|M_{ref}|. \tag{II.3}$$

Note that, the recall will be the same for M and $M^{\approx}$.

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

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

[23] There is no need for more complex mappings in this use case.

```
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.

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 a larger coverage of the domain. See Triple t_{13} in Table I.3 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 (Waagmeester et al. 2020) 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

[24] Wikidata endpoint: https://query.wikidata.org/sparql

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 II.5. 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 II.5.

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 chemicals has 4% ChEMBL identifiers, 0.5% MeSH identifiers, 55% PubChem identifiers, and 95% InChiKey identifiers.

II.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.

1. 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 II.5.

2. 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 II.5. 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.

3. Finally, we add the labels of the divisions found in *divisions.dmp* (see triples t_{26} and t_{28}). We also add disjointness axioms among unrelated divisions, *e.g.*, triple t_{27} in Table II.5.

We use the TraitBank from EOL (Parr et al. 2014b) 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

```
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 compounds. `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.

shown as triple t_{29} in Table II.5. The EOL TraitBank also includes subsumption definitions (*i.e.*, via `rdfs:subClassOf`) for a large portion of traits. These subsumptions can be downloaded separately and are added to TERA in a similar way as mentioned above.

II.5.2.4 Chemical sub-KG construction

The Chemical sub-KG (KG_C) is created from PubChem (Kim, J. Chen, et al. 2018), ChEMBL (Hastings, Owen, et al. 2016), and MeSH (NLM 2020). 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:

1. component information, *i.e.*, what are the building blocks of the chemical or parts of a mixture;

2. type assertions, which either link to ChEBI or describe the type of molecule, *e.g.*, small or large;

3. role assertions, which describe additional attributes or relationships of the chemical, *e.g.*, `FDAApprovedDrug`; and

4. drug products, which link to the clinical data in SNOMED CT (Benson 2012).

Examples of these can be seen in triples t_{35}, t_{36} and t_{37} in Table II.5.

Parent chemical data in PubChem is limited to permutations *e.g.*, bonds, polarity, and part of mixtures axioms (triple t_{34} in Table II.5). 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 (Swain et al. 2014)) is ≥ 0.9,[25] see triple t_{38} in Table II.5.

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 II.5). 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 II.5 show examples of chemical group and functional classifications.

[25] Default value used in PubChem (Kim, Bolton, and Bryant 2016).

```
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, concentrations and units, where the species is endemic to the *Oslofjord*.

II.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.

[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$'_C$	6.9×10^3	8.6	2.3	17	7.7×10^{-5}
TERA KG$'_S$	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 II.7: Densities and entropies of benchmark datasets. TERA KG_C and KG_S are the chemical and species parts of TERA, while KG'_C and KG'_S denote the parts of TERA used in prediction in Section II.7.

II.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 II.6, where the effect prediction model is built upon state-of-the-art knowledge graph embedding models.

Table II.7 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, Augustine, and Getoor 2017 and calculate the relational density, $RD = |\mathcal{T}|/|\mathcal{R}|$, and entity density, $ED = 2|\mathcal{T}|/|\mathcal{E}|$, where $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{E}$ 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 (Pujara, Augustine, and Getoor 2017), and are calculated as

$$P(r) = \frac{|t.p = r|}{|T|}, \tag{II.4}$$

$$P(e) = \frac{|t.sb = e| + |t.ob = e|}{|T|}, \tag{II.5}$$

$$RE = \sum_{r \in \mathcal{R}} -P(r)log(P(r)), \tag{II.6}$$

$$EE = \sum_{e \in \mathcal{E}} -P(e)log(P(e)), \tag{II.7}$$

where $|t.p = r|$ is the number of triples with r as predicate, and $|t.sb = e| + |t.ob = e|$ is the number triples with e as subject or object.

[27] YAGO3-10 (Suchanek, Kasneci, and Weikum 2007), FB15k-237 (Bollacker et al. 2008b), WN18 (Miller 1995) and WN18RR (Dettmers et al. 2018).

In addition, we calculate the absolute density of the graph, which is $AD = |\mathcal{T}|/(|\mathcal{E}|(|\mathcal{E}|-1))$. This is the ratio of edges to the maximum number of edges possible in a simple directed graph (Coleman and Moré 1983).

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.*, Dettmers et al. 2018). In Table II.7 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 (Dettmers et al. 2018). 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 and KG'_S) for prediction. Full details of the construction of KG'_C and KG'_S are given in Section II.7.1.1.

II.6 Adverse Biological Effect Prediction

The aim of chemical effect prediction is to extrapolate exiting data to new combinations of (possibly unknown) chemicals and species. In this section we present three classification models used to predict the adverse biological effect of chemicals on species:

1. a multilayer perceptron (MLP) model (our baseline),
2. the baseline model fed with pre-trained KG embeddings,
3. a model that simultaneously trains the baseline model and the KGE models (*i.e.*, it fine-tunes the KG embeddings).

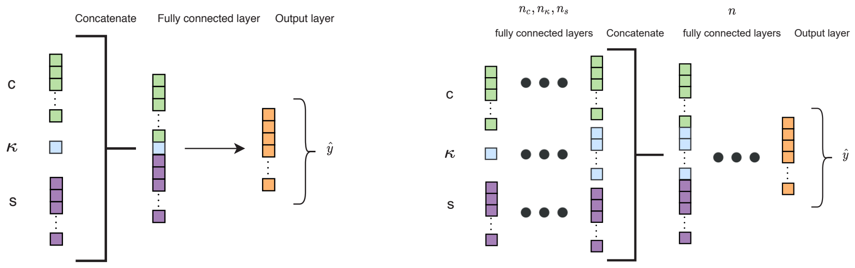
A MLP was chosen as baseline as it is a basic model where additional components and penalties can be easily added and assessed as we do in our third model (see Section II.6.3).

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

$$y_{c,s,\kappa} = \begin{cases} 1 & c \text{ is lethal to } s \text{ at } \kappa, \\ 0 & \text{otherwise.} \end{cases} \tag{II.8}$$

Note that the effect can have a more fine-grained categorization (endpoints LCx, LDx, ECx[28], and NR-LETH in Table II.2). Without losing the generality in introducing and evaluating our effect prediction methods, we simplify the effect into two cases: “lethal” and “non-lethal”.

[28] If effect is mortality (*e.g.*, see Table II.4).



(a) Simple setting. Without transformation layers: $n_c = 0, n_s = 0, n_\kappa = 0$ and $n = 1$.

(b) Complex setting. Model with branches/transformation layers. In contrast to the simple setting, here $n_c \geq 1, n_s \geq 1, n_\kappa \geq 1$ and $n \geq 1$.

Figure II.4: Baseline model. Inputs: c, s, κ as in Equation (II.9); Outputs: $\hat{y}$ as in Equation (II.15).

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.

II.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_κ hidden layers appended to the real valued chemical concentration κ. 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 $x_{c,s,\kappa}$ as input):

$$\mathbf{y}_c^0 = \mathbf{e}_c,\ \mathbf{y}_s^0 = \mathbf{e}_s,\ y_\kappa^0 = \kappa \tag{II.9}$$

$$\mathbf{y}_c^h = ReLu(\mathbf{y}_c^{h-1}\mathbf{W}_c^h + \mathbf{b}_c^h),\ h \in \{0, \dots, n_c\} \tag{II.10}$$

$$\mathbf{y}_s^h = ReLu(\mathbf{y}_s^{h-1}\mathbf{W}_s^h + \mathbf{b}_s^h),\ h \in \{0, \dots, n_s\} \tag{II.11}$$

$$\mathbf{y}_\kappa^h = ReLu(\mathbf{y}_\kappa^{h-1}\mathbf{W}_\kappa^h + \mathbf{b}_\kappa^h),\ h \in \{0, \dots, n_\kappa\} \tag{II.12}$$

$$\mathbf{y}^0 = [\mathbf{y}_c^{n_c}, \mathbf{y}_s^{n_s}, \mathbf{y}_\kappa^{n_\kappa}] \tag{II.13}$$

$$\mathbf{y}^h = ReLu(\mathbf{y}^{h-1}\mathbf{W}^h + \mathbf{b}^h),\ h \in \{1, \dots, n\} \tag{II.14}$$

$$\hat{y} = \sigma(\mathbf{y}^n\mathbf{W}^n + \mathbf{b}^n) \tag{II.15}$$

$\mathbf{e}_c, \mathbf{e}_s \in \mathbb{R}^k$ in (II.9) denote the embeddings of c and s respectively, and are calculated as

$$\mathbf{e}_c = \delta_c\mathbf{W}_c, \quad \mathbf{e}_s = \delta_s\mathbf{W}_s \tag{II.16}$$

where δ_c and δ_s denote the one-hot encoding vectors of the chemical entity c (w.r.t. all the entities in $\mathcal{E}_C$ from KG_C) and the species entity s (w.r.t. all the entities in $\mathcal{E}_S$ from KG_S), respectively;[29] $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{E}_C| \times k}$ and $\mathbf{W}_s \in \mathbb{R}^{|\mathcal{E}_S| \times k}$ are embedding transformation matrices to learn. (II.10), (II.11) and (II.14) 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. $[\cdot, \cdot]$ in (II.13) denotes vector concatenation. σ in (II.15) 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 II.4):

1. *Simple setting.* Figure II.4a shows the model without embedding transformation layers, *i.e.*, $n_s = n_c = n_\kappa = 0$, and $n = 1$.

2. *Complex setting.* The complex model shown in Figure II.4b introduces transformation layers on the embeddings and chemical concentration input. These transformations 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.

II.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 II.4.2 and Appendix II.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 (II.16) by the matrices composed of embeddings by the respective KGE models. Namely $\mathbf{W}_c$ is set to $[\mathbf{e}_{c,1}; \mathbf{e}_{c,2}; \ldots; \mathbf{e}_{c,|\mathcal{E}_C|}]$, $\mathbf{W}_s$ is set to $[\mathbf{e}_{s,1}; \mathbf{e}_{s,2}; \ldots; \mathbf{e}_{s,|\mathcal{E}_S|}]$, where $[\cdot; \cdot]$ denotes stacking vectors, $\mathbf{e}_{c,i}$ denotes the embedding of the i^{th} chemical in the chemicals KG_C, $\mathbf{e}_{s,i}$ denotes the embedding of the 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 species KG, respectively (*e.g.*, *Complex PT DistMult-HAKE*). For simplicity, we also refer to these models as PT-based models.

II.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

[29] $\delta_c \in \mathbb{R}^{|\mathcal{E}_C|}$, where $\delta_c^i = 1$ if c is the i^{th} chemical in $\mathcal{E}_C$, else 0. δ_s is defined similarly.

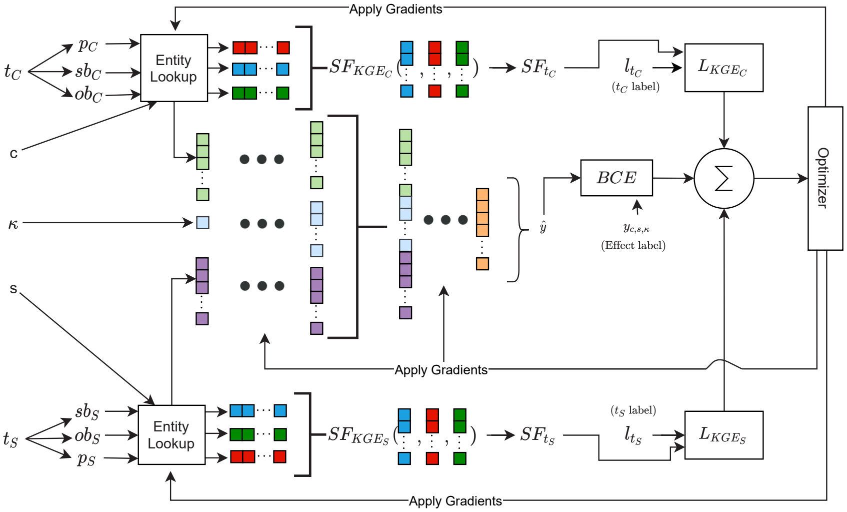


Figure II.5: Fine-tuning optimization model. In addition to variables described in Figures II.4a and II.4b, $t_C = (sb_C, p_C, ob_C) \in KG_C \cup \overline{KG}_C$, $t_S = (sb_S, p_S, ob_S) \in KG_S \cup \overline{KG}_S$. Entity lookups transform an entity into a vector (see Equation (II.16)). SF_{KGE_C} and SF_{KGE_S} are the triple scoring functions implemented by the selected KGE model (see Appendix II.A). SF_{t_C} and SF_{t_S} are the scores for a chemicals and species triple, respectively. $x_{c,s,\kappa}$ is the prediction input and $y_{c,s,\kappa}$ is described in Equation (II.8). l_{t_C} and l_{t_S} are the triple labels (*i.e.*, True or False). BCE is the binary cross-entropy loss function (from Equation (II.18)). The summation of the losses is described in Equation (II.17), that is the loss used by the optimizer to apply changes to model weights.

the (selected) KGE models and the MLP-based baseline model. Such that the $\mathbf{W}_C$ and $\mathbf{W}_S$, and the MLP weights ($\mathbf{W}_x$ and $\mathbf{b}_x$ in Equations (II.10), (II.11), (II.14) and (II.15)) are optimized simultaneously. Note that we initialize the KGE models with the previously pre-trained embeddings.

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

$$L = \alpha_C L_{KGE_C} + \alpha_S L_{KGE_S} + \alpha_{MLP} L_{MLP} \tag{II.17}$$

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,[30] α_C and α_S denote their weights respectively, L_{MLP} and α_{MLP} denote the loss of the MLP and its weight. Specifically, we use binary cross-entropy (BCE) as the loss for the

[30] Appendix II.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.

classification. L_{MLP} is calculated as

$$L_{MLP} = -\frac{1}{N}\sum_{i}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{II.18}$$

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 (II.8)). With the overall loss, gradient-based learning algorithms such as Adam optimizer (Kingma and Ba 2014) can be adopted to jointly training the embeddings of both KGEs and the MLP.

Figure II.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 summarised as follows:

1. Select N triples from KG_C and KG_S, where N is the length of the effects training set.[31]

2. Generate negative knowledge graph triples (see Appendix II.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 (only L_{MLP}) over the validation set stops improving.

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.

II.7 Results

II.7.1 Experimental setup

All models are implemented using Keras (Chollet et al. 2015) and the model codes are available in our GitHub repository, alongside all data preparation and analysis scripts.[32]

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

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

II.7.1.1 Preparation of TERA for prediction

As shown earlier, TERA consists of three sub-KGs. These are the basis for the chemical effect prediction.[33] 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:

1. 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 (κ, 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

2. 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 (Kim, Bolton, and Bryant 2016).

3. 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 and KG'_S, are given in Table II.7 (Section II.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 which corresponds to the index of a column vector in matrices $\mathbf{W}_c$ or $\mathbf{W}_s$ in Equation (II.16), depending on which sub-KG is transformed.[34] Relations are treated similarly.

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

[34] $i \in [0, |\mathcal{E}_C| - 1]$ for KG_C and $i \in [0, |\mathcal{E}_S| - 1]$ for KG_S

II.7.1.2 Sampling

We use four sampling strategies of the effect data to analyze 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, *c.f.*, Section II.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:

Strategy *(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).

Strategy *(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 77%/14%/9% split.

Strategy *(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 77%/14%/9% split.

Strategy *(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. For example, some chemicals are used on (close to) all species, therefore, these chemicals are discarded in the sampling strategy *(iv)*, affecting the final number of samples.

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.*, $\approx$ 16% of samples are labelled as non-lethal and $\approx$ 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 (Branco, Torgo, and Ribeiro 2016).

II.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 II.8 and II.9 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, $RL_{ep} = L_{ep}/L_0$, where L_{ep} is the training loss at epoch ep, L_0 is the loss with the initial weights.

We use validation loss to select the best hyper-parameter setting for the classification models presented in Section II.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 II.7.2.

The hyper-parameter ranges for the KGE models are shown in Table II.8 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 II.9 we show the best hyper-parameters.

We can see in Table II.9 that the decomposition models have similar hyper-parameters for KG_C and KG_S. As shown in Section II.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-$|\mathcal{E}|$ scoring, where $|\mathcal{E}|$ is the number of entities in the KG (Dettmers et al. 2018).

The *fine-tuning optimization model* (Section II.6.3), in order to save on intensive computation, reuses the same hyper-parameters found for the KGE

KGE hyper-parameters	**Search space**
Loss function	$\{L_{H_1}, L_{H_2}, L_{L_1}, L_{L_2}\}$
Margin (only hinge loss)	$\{1, 2, \dots, 10\}$
Bias (only geometric models)	$\{0, 1, \dots, 20\}$
Embedding dimension	$\{100, 101, \dots, 400\}$
Negative samples	$\{10, 11, \dots, 100\}$
Prediction hyper-parameters	**Search space**
n_c (II.10), n_s (II.11), n_κ (II.12), n (II.14)	$\{0, 1, 2, 3\}$
# units (II.10), (II.11), (II.14)	$\{2^u \text{ with } u \in \{4, 5, \dots, 10\}\}$
# units (II.12)	$\{2^u \text{ with } u \in \{2, 3, 4, 5\}\}$

Table II.8: Hyper-parameter choices for the models. Please refer to the Equations (II.9)-(II.15) in Section II.6.1 for the prediction hyper-parameters.

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 II.9: 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 II.10: 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 fine-tuning models. Organized as follows: $(|\mathbf{b}_c^1|,...,|\mathbf{b}_c^{n_c}|)/(|\mathbf{b}_s^1|,...,|\mathbf{b}_s^{n_s}|)/(|\mathbf{b}_\kappa^1|,...,|\mathbf{b}_\kappa^{n_\kappa}|)/(|\mathbf{b}^1|,...,|\mathbf{b}^n|)$ as in Equations (II.10), (II.11), (II.12), and (II.14)). $-$ denotes no hidden layers. *e.g.*, $(128)/(256)/(8,8)/-$ denotes $n_c = 1, n_s = 1, n_\kappa = 2, n = 0$ and $|\mathbf{b}|_c^1 = 128$, $|\mathbf{b}_s^1| = 256$, $|\mathbf{b}_\kappa^1| = 8$ and $|\mathbf{b}_\kappa^2| = 8$.

models. Depending on the optimizer choice, the choice of loss weights, α_C, α_S, and α_{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 $\alpha_C = \alpha_S = \alpha_{MLP} =$ 1. Had we used, *e.g.*, stochastic gradient descent, these variables would needed to be tuned.

II.7.1.4 Initialization of the fine-tuning optimization models

As presented in Section II.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.

II.7.1.5 Simple and complex settings

As presented in Section II.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 MLP model. The simple setting uses no branching layers, *i.e.*, $n_C = n_S = n_\kappa = 0$ and $n = 1$ as in Equations (II.10), (II.11), (II.12) and (II.14) 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 II.8. The optimal choices for the top performing models (using one-hot and pre-trained embeddings) are shown in Table II.10.

Looking at the increasing complexity of the layer configuration of the one-hot models in Table II.10 we can see a correlation from the simplest sampling strategy (*i.e.*, *(i)*) through the most challenging one (*i.e.*, *(iv)*). The same can be seen for PT HAKE-DisMult from strategy *(iii)* 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 KG_S than KG_C.

II.7.2 Prediction results

In this section we present a summary of the conducted chemical effect prediction evaluation. Complete results are available at the project repository.[35] The default decision threshold is set to 0.5. That is, if a model predicts $\hat{y} > 0.5$ for an input $x_{c,s,\kappa}$ then the chemical c is considered lethal to s at a concentration κ.[36]

We use several metrics to compare the different prediction models. These are Sensitivity (*i.e.*, recall), Specificity, and Youden's index (YI) (Youden 1950). Precision and F-score were also considered as metrics. However, they were not representative for the performance with respect to non-harmful chemicals. This is attributed to the larger number of positive samples (*i.e.*, harmful chemicals) than negative samples (*i.e.*, non-harmful chemicals) in the test data.

Sensitivity and Specificity are defined as

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \tag{II.19}$$

$$\text{Specificity} = \frac{TN}{FP + TN}, \tag{II.20}$$

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

$$YI = \text{Sensitivity} + \text{Specificity} - 1. \tag{II.21}$$

We also present the maximized Youden's index (YI_{max}), this is defined as

$$YI_{max} = \max_t \ \text{Sensitivity} + \text{Specificity} - 1, \tag{II.22}$$

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

[36] We set the decision threshold $\hat{y} > 0.5$ since the model output bias (*c.f.*, Equation (II.15)) will be (close to) 0.5 after training. Recall that we have oversampled the classes to reach a 0.5/0.5 prior probability during training (*c.f.*, Section II.7.1.2).

Model	**Sensitivity**	**Specificity**	**YI**	$\mathbf{YI}_{max}$	τ_{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	$\mathbf{0.997 \pm 0.003}$	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	$\mathbf{0.734 \pm 0.006}$	0.743 ± 0.007	0.547 ± 0.074
Simple FT pRotatE-HAKE	0.92 ± 0.005	0.808 ± 0.013	$\underline{0.728 \pm 0.011}$	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	$\mathbf{0.765 \pm 0.006}$	0.842 ± 0.064
Simple FT RotatE-ConvE	0.928 ± 0.015	0.797 ± 0.036	$\underline{0.726 \pm 0.022}$	$\underline{0.761 \pm 0.01}$	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	$\underline{0.996 \pm 0.006}$	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	$\underline{0.726 \pm 0.018}$	0.754 ± 0.008	0.776 ± 0.109
Complex FT HolE-ConvKB	0.895 ± 0.01	$\mathbf{0.835 \pm 0.016}$	$\underline{0.73 \pm 0.01}$	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 II.11: Prediction results (mean and standard deviation over 10 runs) for sampling strategy *(i)*. **Bold** denotes *best mean result* and underline denotes *within one standard deviation of best result.* PT prefix denotes pre-trained and FT denotes fine-tuning. Simple denotes $n_C = n_S = n_\kappa = 0$ and $n = 1$ while in complex, n_C, n_S, n_κ and n are hyper-parameters in Equations (II.10), (II.11), (II.12) and (II.14).

i.e., we maximize Youden's index based on the decision threshold (τ), we call this optimal threshold τ_{max}. This metric is equivalent to the maximum of the Receiver operating characteristic (ROC) curve over a random 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 than 0, but this is solved by swapping labeled classes. Similarly to how negative correlation is still useful.

Tables II.11-II.14 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

Model	**Sensitivity**	**Specificity**	**YI**	**YI**$_{max}$	τ_{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 II.12: Prediction results for sampling strategy *(ii)*. Same notation as Table II.11.

Model	**Sensitivity**	**Specificity**	**YI**	**YI**$_{max}$	τ_{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 II.13: Prediction results for sampling strategy *(iii)*. Same notation as Table II.11.

Model	**Sensitivity**	**Specificity**	**YI**	$\mathbf{YI}_{max}$	τ_{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	$\underline{0.971 \pm 0.011}$	0.361 ± 0.065	0.332 ± 0.056	$\underline{0.546 \pm 0.031}$	0.89 ± 0.042
Simple PT pRotatE-ComplEx	$\mathbf{0.972} \pm \mathbf{0.008}$	0.36 ± 0.079	0.332 ± 0.074	$\underline{0.527 \pm 0.045}$	0.852 ± 0.04
Simple PT HolE-ComplEx	$\underline{0.965 \pm 0.032}$	0.363 ± 0.068	0.328 ± 0.063	$\underline{0.549 \pm 0.075}$	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	$\underline{0.963 \pm 0.01}$	0.423 ± 0.102	$\underline{0.386 \pm 0.096}$	$\underline{0.57 \pm 0.03}$	0.875 ± 0.079
Simple FT pRotatE-ComplEx	0.954 ± 0.009	$\underline{0.5 \pm 0.058}$	$\underline{0.454 \pm 0.052}$	$\underline{0.569 \pm 0.024}$	0.854 ± 0.073
Simple FT HolE-ComplEx	$\underline{0.965 \pm 0.007}$	0.418 ± 0.058	0.383 ± 0.053	$\mathbf{0.571} \pm \mathbf{0.042}$	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	$\underline{0.513 \pm 0.064}$	$\underline{0.437 \pm 0.058}$	0.522 ± 0.034	0.83 ± 0.09
Complex FT HolE-DistMult	0.926 ± 0.015	$\mathbf{0.536} \pm \mathbf{0.03}$	$\mathbf{0.462} \pm \mathbf{0.03}$	$\underline{0.543 \pm 0.039}$	0.81 ± 0.084
Complex FT ConvKB-DistMult	0.933 ± 0.01	$\underline{0.525 \pm 0.065}$	$\underline{0.459 \pm 0.063}$	$\underline{0.55 \pm 0.04}$	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 II.14: Prediction results sampling strategy *(iv)*. Same notation as Table II.11.

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. $\underline{\text{Underlined}}$ results are where there is a $\geq 32\%$ chance that a single run outperforms the best mean (*i.e.*, one standard deviation contains 68% of results, assuming normally distribute 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 τ_{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.

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.

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

II.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 1.5% 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 about the entities, specially for unseen chemicals and species, that the KG embedding models aim at capturing.

II.7.2.2 Baseline with pre-trained KG embeddings

We can see that the PT-based models do not lead to an important improvement with respect to YI_{max} in 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 deviation in the same order of magnitude as the metric (*e.g.*, *Simple FT HAKE-ComplEx*).

II.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 *(iii)* and *(iv)*. For example, the FT-based models *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 fine-tuning of the

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 II.15: 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.

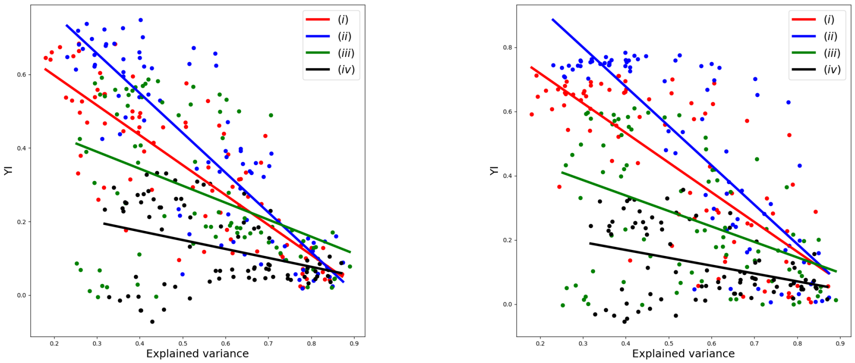
KG embeddings. The results are expected as the fine-tuned KG embeddings are tailored to the effect prediction task.

II.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 II.15 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 influence of the non-linear transforms on the embeddings.

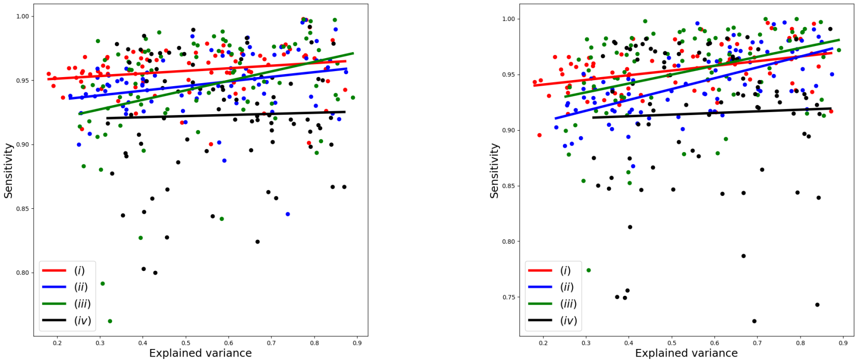
Looking at Table II.15 we can see that the KGE models used to embed the chemicals KG_C in the best performing models is distributed evenly across most models and settings. This indicates that the performance of the prediction models is not highly correlated with the use of a KGE model on KG_C. Referencing Table II.7, the high relational density in KG_C can contribute to worse performance (Pujara, Augustine, and Getoor 2017) and therefore equal distribution of models in Table II.15. This is different for KG_S. For sampling strategies *(i)* and *(ii)*, HAKE is extensively used in the top models to embed KG_S. HAKE is designed to embed hierarchies. Therefore, this indicates that in strategies *(i)* and *(ii)* the hierarchical structure of KG_S dwarfs the rest of the KG. KG_S has a higher entity density and lower entity entropy (Table II.7) than KG_C. This should lead to higher performance generally, but might also lead to larger discrepancies between models as seen in Table II.15.



(a) Simple PT models.

(b) Complex PT models.

Figure II.6: Relation between explained variance using 10 principal components and model performance represented as YI.

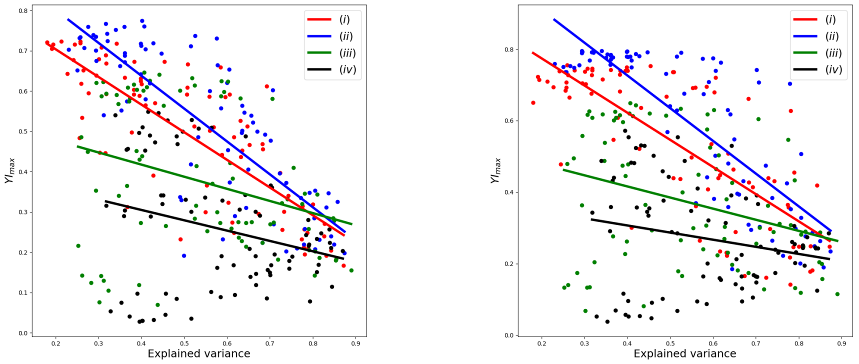


(a) Simple PT models.

(b) Complex PT models.

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

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)*.



(a) Simple PT models.

(b) Complex PT models.

Figure II.8: Relation between explained variance using 10 principal components and model performance represented as YI_{max}.

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 II.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 II.6a and complex setting in Figure II.6b. For example, in Figure II.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 II.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 II.6 are explained by specificity rather than sensitivity. By balancing sensitivity and specificity, *i.e.*, YI_{max} as seen in Figure II.8, the rate of change is reduced compared to YI in Figure II.6.

[38] We use the scikit-learn implementation (Pedregosa et al. 2011) based on Tipping and Bishop 1999.

Chemical	Species	$\log(\kappa)$	Predicted	Lethal	Classification
D001556 (hexachlorocyclohexane)	59899 (walking catfish)	−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 sunfish)	−0.9	0.86	0 (no)	FP

Table II.16: Example predictions by Complex FT HolE-DistMult (best model) for sampling strategy *(iv)*.

II.7.4 Example predictions

Table II.16 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 flower (*i.e.*, *eudicots*) could be due to the fact that flowers, and plants in general, are severely underrepresented in the available effect prediction data.

II.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 benefits of creating tailored (*i.e.*, fine-tuned) KG embeddings in the FT-based prediction models.

II.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 effects 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 identifiers. Biological data, both taxonomic and effects, might be harder to align to TERA as these mappings are not available in Wikidata. Here, ontology alignment systems play an important role to fill 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. This was 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.

II.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 II.15) 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.

II.9 Conclusions and Future Work

TERA is a novel knowledge graph which includes large amounts of data required by ecological risk assessment. We have conducted an extensive evaluation of KGE embedding models in a novel and very challenging application domain.

Moreover, 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.

II.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. Where the 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. These 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 (Wittwehr et al. 2019). We also believe the development of TERA contributes to a methodological change in the community, and encourages others to make their data interoperable.

II.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.*, Sharma et al. 2017). 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 (Y. Wu and G. Wang 2018; H. Yang et al. 2018). 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.*, Lécué and J. Wu 2018).

II.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 II.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.

II.9.4 Value to the ontology alignment community

As mentioned in Section II.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) (Myklebust, Jiménez-Ruiz, et al. 2020) 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 (*e.g.*, as done in the phenotype-disease domain (Harrow et al. 2017)) that could become the reference alignment to integrate ECOTOX and NCBI Taxonomy.

II.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 (U.S. Environmental Protection Agency. 2020) 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 Skrindebakke 2020. 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.*, Mumtaz and Giese 2021, or imposing restrictions on the embeddings, such as a minimum distance constraint.

II.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 II.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).

Appendix II.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 Rossi, Barbosa, et al. 2021 for a comprehensive survey.

II.A.1 Notation

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

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.

II.A.2 Decomposition models

DistMult. Developed by B. Yang et al. 2015 and shown to have state-of-the-art performance on link prediction tasks under optimal hyper-parameters (Kadlec, Bajgar, and Kleindienst 2017). 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.

$$SF_{\text{DistMult}}(sb, p, ob) = \langle \mathbf{e}_{sb}, \mathbf{e}_p, \mathbf{e}_{ob} \rangle \tag{II.23}$$

This model does not take the direction of the relation into account, that is, $SF_{\text{DistMult}}(sb, p, ob) = SF_{\text{DistMult}}(ob, p, sb)$.

ComplEx. This model use the same scoring function as DistMult (Trouillon et al. 2016). However, the entity vector representation are in the complex space ($\mathbf{e}_{sb}, \mathbf{e}_p, \mathbf{e}_{ob} \in \mathbb{C}^k$) and therefore, the drawback of lacking directionality in DistMult is solved.

$$\begin{aligned} SF_{\text{ComplEx}}(sb, p, ob) &= \langle \mathbf{e}_{sb}, \mathbf{e}_p, \mathbf{e}_{ob} \rangle \\ &= \langle \Re(\mathbf{e}_{sb}) + i\Im(\mathbf{e}_{sb}), \Re(\mathbf{e}_{sb}) \rangle \\ &+ \langle i\Im(\mathbf{e}_{sb}), \Re(\mathbf{e}_p) + i\Im(\mathbf{e}_{ob}) \rangle \\ &= \langle \Re(\mathbf{e}_{sb}), \Re(\mathbf{e}_p), \Re(\mathbf{e}_{ob}) \rangle \\ &+ \langle \Im(\mathbf{e}_{sb}), \Re(\mathbf{e}_p), \Im(\mathbf{e}_{ob}) \rangle \\ &+ \langle \Re(\mathbf{e}_{sb}), \Im(\mathbf{e}_p), \Im(\mathbf{e}_{ob}) \rangle \\ &+ \langle \Im(\mathbf{e}_{sb}), \Im(\mathbf{e}_p), \Re(\mathbf{e}_{ob}) \rangle \end{aligned} \tag{II.24}$$

where $i = \sqrt{-1}$ and, $\Re(x)$ and $\Im(x)$ are the real and complex parts of x, respectively. We can easily see that $SF_{\text{ComplEx}}(\mathbf{e}_{sb}, \mathbf{e}_p, \mathbf{e}_{ob}) = SF_{\text{DistMult}}(\mathbf{e}_{sb}, \mathbf{e}_p, \mathbf{e}_{ob})$ if $\Im(\mathbf{e}_{sb}) = \Im(\mathbf{e}_p) = \Im(\mathbf{e}_{ob}) = \mathbf{0}$.

HolE. The Holographic embedding model is described in Nickel, Rosasco, and Poggio 2015, and use a circular correlation scoring function

$$SF_{\text{HolE}}(sb, p, ob) = \mathbf{e}_p^T [\mathbf{e}_{sb} \star \mathbf{e}_{ob}], \tag{II.25}$$

$$\mathbf{e}_{sb} \star \mathbf{e}_{ob} = \mathcal{F}^{-1}[\overline{\mathcal{F}(\mathbf{e}_{sb})} \circ \mathcal{F}(\mathbf{e}_{ob})] \tag{II.26}$$

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, $\circ$ denotes the Hadamard product (element-wise). HolE has been show to be equivalent to ComplEx (Hayashi and Shimbo 2017), and therefore, we expect the performance to be similar.

II.A.3 Geometric models

TransE. The translational model has the scoring function (Bordes, Usunier, et al. 2013b)

$$SF_{\text{TransE}}(sb,p,ob) = ||\mathbf{e}_{sb} + \mathbf{e}_p - \mathbf{e}_{ob}||_n. \tag{II.27}$$

Such that if (sb,p,ob) 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 ($e^{i\theta} = \cos(\theta) + i\sin(\theta)$) and scores triples by rotating the relation embedding in complex space. RotatE has been shown to be efficient of modelling symmetric, inverse and composite relations (Sun et al. 2019b). The scoring function of RotatE is defined as

$$\begin{aligned} SF_{\text{RotatE}}(sb,p,ob) &= ||\mathbf{e}_{sb} \circ \mathbf{e}_p - \mathbf{e}_{ob}||_n \\ &= ||\mathbf{e}_{sb} \circ (\cos(\theta_p) + i\sin(\theta_p)) - \mathbf{e}_{ob}||_p \\ &= ||[\Re(\mathbf{e}_{sb})\cos(\theta_p) - \Im(\mathbf{e_{sb}})\sin(\theta_p) - \Re(\mathbf{e}_{ob}) \\ &\quad ;\Re(\mathbf{e}_{sb})\sin(\theta_p) + \Im(\mathbf{e_{sb}})\cos(\theta_p) - \Im(\mathbf{e}_{ob})]||_n. \end{aligned} \tag{II.28}$$

Here, we concatenate the real and complex parts of $\mathbf{e}_{sb} \circ \mathbf{e}_p - \mathbf{e}_{ob}$. The modulus constraint of $\mathbf{e}_p$ 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 (Sun et al. 2019b). pRotatE has the scoring function

$$SF_{\text{RotatE}}(sb,p,ob) = 2MC||\sin(\frac{\theta_{sb} + \theta_p - \theta_{ob}}{2})||_n \tag{II.29}$$

where $\theta_x = \angle \mathbf{e}_x$ (phase of $\mathbf{e}_x$) and MC is the modulus constraint on $\mathbf{e}_{sb}$ and $\mathbf{e}_{ob}$.

HAKE. The hierarchy-aware model use the modulus and the phase part of the embedding vectors (Z. Zhang et al. 2019b). 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

$$\begin{aligned} SF_{\text{pRotatE}}(sb,p,ob) = &\,|||\mathbf{e}_{sb}| \circ |\mathbf{e}_p| - |\mathbf{e}_{ob}|||_n \\ &+ ||\sin(\frac{\theta_{sb} + \theta_p - \theta_{ob}}{2})||_1 \end{aligned} \tag{II.30}$$

where $|\cdot|$ is the modulus of $\cdot$. The authors noted that a mixture bias can be added to $|||\mathbf{e}_{sb}| \circ |\mathbf{e}_p| - |\mathbf{e}_{ob}|||_n$ to improved performance (Z. Zhang et al. 2019b). We omit these details here.

II.A.4 Convolutional models

The final set of models used in this work are convolutional models. We denote convolutions between an *image* X and filters ω is denoted as $X * \omega$. The models also use dense layers, which is denoted by transform matrices, *e.g.*, $\mathbf{W}$, 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 (D. Q. Nguyen et al. 2018) use a single convolutional layer and a single dense layer

$$\begin{aligned} SF_{\text{ConvKB}}(sb, p, ob) = \\ & f(\text{vec}(f([\mathbf{e}_{sb}; \mathbf{e}_p; \mathbf{e}_{ob}] * \omega))\mathbf{W}), \end{aligned} \tag{II.31}$$

where $\text{vec}(x)$ reshapes x to a 1-dimensional vector. ω 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 (Dettmers et al. 2018) only perform convolution over the subject and predicate *image* (concatenated and reshaped) and multiples the output dense layer with the object vector as such

$$\begin{aligned} SF_{\text{ConvE}}(sb, p, ob) = \\ & f(\text{vec}(f([\overline{\mathbf{e}_{sb}}; \overline{\mathbf{e}_p}] * \omega))\mathbf{W})\mathbf{e}_{ob}^T \end{aligned} \tag{II.32}$$

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, (Dettmers et al. 2018) found that this did not yield improved results.

II.A.5 Loss functions

Work on KGE models usually define loss functions specific to the models. However, as show in (S. Mohamed et al. 2019; Nayyeri et al. 2019) 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 (Dong et al. 2014), *i.e.*, we sample from $\{sb, p, ob'\} \notin KG, ob' \in \mathcal{E}$. 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 Ali et al. 2020). The number of negative samples sampled per positive sample is controlled by a hyper-parameter. However, Kadlec, Bajgar, and Kleindienst 2017 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.

$$L_{H_1} = \sum_{\mathbf{t}\in X} [\gamma - y_{\mathbf{t}} S(\mathbf{t})]_+ \tag{II.33}$$

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 $\mathbf{t}$. γ 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.

$$L_{L_1} = \sum_{t\in X} \log(1 + \exp(-y_{\mathbf{t}} S(\mathbf{t})) \tag{II.34}$$

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

$$L_{H_2} = \sum_{\mathbf{t}^+\in X^+} \sum_{\mathbf{t}^-\in X^-} [\gamma + S(\mathbf{x}^-) - S(\mathbf{x}^+)]_+ \tag{II.35}$$

where X^+ and X^- are the sets of positive and negative triples, respectively. γ 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

$$L_{L_2} = \sum_{\mathbf{t}^+\in X^+} \sum_{\mathbf{t}^-\in X^-} \log(1 + \exp(S(\mathbf{x}^-) - S(\mathbf{x}^+)). \tag{II.36}$$

II.A.6 Implementation

We have implemented the KGE models in Keras (Chollet et al. 2015) 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 II.6.

Appendices

Appendix A

The TERA Knowledge Graph

This appendix will describe the construction of TERA and it's accompanying tools. This will not be exhaustive and further documentation can be found at https://niva-knowledge-graph.github.io/TERA.

A.1 Aggregation

Each data source component of the aggregation is based on a common data object. The data object has the methods apply, replace, and save. apply provides a wrapper to applying a function row-wise in tabular data. This function is unique to each tabular data source. The replace method simply replaces entities based on provided triples of *from-to* mappings. Finally, save saves the KG to provided path.

As mentioned, each data source requires a unique mapping function, however, this does not apply to data sources where RDF data is already available. Therefore, we will only describe the mapping functions where the source is tabular.

ECOTOX Chemical database. We are interested in three columns in this tabular data, CAS numbers (chemical identifier), chemical name, and chemical ECOTOX group. For each row, CAS numbers are concatenated with the namespace[1] and is added to the KG with a `type` assertion that it is a chemical. The chemical name column can contain multiple names, comma-separated. We add these to the KG by adding `rdfs:label` triples. A chemical can also be in multiple chemical groups. We construct classes by concatenating group names and the namespace. Thereafter, we add class assertions, adding the chemical to the groups in the KG.

ECOTOX Taxonomy database. Three tabular files in ECOTOX make up the taxonomy. Firstly, we load the taxa identifier and their common (English) name, Latin name, and taxa ECOTOX group. Same as above we concatenate the species identifier and the namespace[2] before adding it and the species class assertion to the KG. If a common and Latin name is avalible for the taxon these are added to the KG using custom predicates `ecotox:commonName` and `ecotox:latinName` which are sub-properties of `rdfs:label`. The second file contains Latin synonyms for the species, we add these in the same way as for the primary Latin names. The final file treated contain the in-compete ECOTOX species hierarchy. Each column contains the taxa at each level in the taxonomy. We add subsumption triples for each level in the taxonomy. However, in a

[1] https://cfpub.epa.gox/ecotox/cas/

[2] http://cfpub.epa.gov/ecotox/taxon/

few rows the taxa is missing, therefore, we construct artificial taxa made up by the taxa at the lower level and the hierarchical level (taxa rank). This ensures consistent depth in the hierarchy which help when aligning to the NCBI Taxonomy. To enable queries based on taxonomic rank, we add rank assertion to the KG. Finally, we add disjointness axioms to the KG. There are 16 classes and we add disjointness between the appropriate ones, *e.g.*, mammals are disjoint with birds, while vertebrates are not disjoint with either. This will help guide the ontology matching methods to not make illogical alignments.

ECOTOX Effect database. The effect data is held in two tabular file, tests and results. Note that, a single test can have multiple results. The columns of concern in the test file are: test identifier, CAS number, taxa identifier, study duration and unit, and organism habitat, life stage, age, and weight. There are overlap between some column, *e.g.*, life stage and age, therefore, some columns will be missing and we will simply not add this data to the KG. We create a new entity for each test and add triples for the test identifier, CAS number, and taxa identifier to the KG. For the study duration, age, and weight, and their units, we apply a unit parser to account for all different ways units can exist in the data, *e.g.*, mg/L or mg/l. Thereafter, we create a blank node and use the predicates `rdf:value` and `unit:units` to add these data to the KG. Thereafter, the test is associated with them using predicates `ecotox:studyDuration`, `ecotox:organismAge`, or `ecotox:organismWeight`. If the habitat or life stage exist we add these to the KG by using the predicates `ecotox:organismHabitat` and `ecotox:organismLifestage`.

The results files contains the columns: test identifier, endpoint, concentration and unit, and effect. We assert if the test exists in the KG and add endpoint and effect to a blank node representing a result with the predicates `ecotox:endpoint` and `ecotox:effect`. The concentration and unit is treated similarly to the other numerical values above, and added to the result using the `ecotox:concentration` predicate.

NCBI Taxonomy. We treat this similarly to the ECOTOX taxonomy. However, the files are organised a little differently. A file contains the hierarchy, another has the names of taxa, and one has the divisions metadata (analogous to groups in ECOTOX). From the hierarchy we gather the columns: child, parent, rank, and division. The child and parent are added to the KG by concatenating with the namespace[3]. We add subsumption axioms between the child and parent entities. We treat the rank and division (group) as above.

The names of taxa is loaded from a file with the columns: taxon identifier, name, unique name, and name type. We add these in the same way as for the ECOTOX taxonomy, however, using name type as a predicate, *e.g.*, `ncbi:latinName`.

The division file contains the division identifier, its acronym and name. This is added to KG as for the ECOTOX groups. The divisions are slightly different than ECOTOX groups, but and we add the disjointness axioms accordingly.

[3] https://www.ncbi.nlm.nih.gov/taxonomy/

Encyclopedia of Life Traits. The main files of interesting in the Encyclopedia of Life TraitBank contain traits and terms. The terms contain a URL and a description. We assume the URL is unique and treat is as a URI. We add this and its description to the KG using `rdfs:comment`. The traits have the columns: page identifier, predicate, a measurement, and a measurement unit. As the predicates are provided along with the measurement and its unit. The measurement is not necessarily numerical, but can be a page (URL/I) to EOL or external sources. The measurement unit is not used if the measurement is not numerical. We ensure that all page identifiers, predicates, and non-numeric measurements are valid URIs.

Finally, the TraitBank contain hierarchies of the page identifiers above. The two columns: child and parents are added as subsumption axioms to the KG.

A.2 Integration

The set of integration APIs and methods aid in the integration and alignment of the data sources in TERA. The top API for alignment defines one method and requires another to be defined. Firstly, the method convert is responsible for converting between two identifiers defined as URI-pairs in a set. Secondly, the load method is required to be implemented for each sub-API of the top alignment API. This top API is used directly in the set of data access APIs as described in the next section. The alignment sub-APIs vary in complexity and is described below with increasing level of complexity.

The top API also has private methods[4], such as add, to facilitate combinations of mappings.

Endpoint mapping queries a provided SPARQL endpoint, *e.g.*, EMBL-EBI[5], based on the `owl:sameAs` property.

Wikidata. Two APIs can be used for accessing Wikidata mappings, one for a pre-downloaded file, and one for querying Wikidata directly. The former might be necessary due to timeout error on large queries. The downloaded mappings are in a tabular file with two columns, *from* and `to`, and can easily be loaded. Querying the Wikidata SPARQL endpoint for mappings requires the input of a valid query. An example of this is shown in the API documentation. The API will handle the fetching and extraction of the mappings from the query results.

Chemical mappings. All chemical mappings are gathered from Wikidata and therefore, we sub-class the Wikidata API above to create these. These same is true for the mappings from NCBI taxonomy to EOL pages.

Ontology matching. We run OM tools which provide RDF file on the OAEI format. We load these and access the nodes which described the mappings. Note that, based on a provide threshold or a uniqueness requirement (`true/false` input to API), these mappings can be one-to-many.

[4] An internal method used by the API itself.

[5] https://www.ebi.ac.uk/rdf/services/sparql

String matchers. Finally, we define a string matching API, which either take two KGs or two dictionaries (entity to strings mapping) as input. If KGs are provided we load them and covert to dictionary form and continue with these. For each entity in the dictionaries, we perform fuzzy string matching between the potential lists of labels and extract the best match. This best match is added to the mapping if the similarity is gather than a predefined threshold.

A.3 Access

We define a top API for data access to TERA data. This API is instantiated with several parameters:

1. namespace, *i.e.*, the base namespace of the data;
2. endpoint, if TERA is hosted at an SPARQL endpoint;
3. data object, an instance of the data object API, note that, one of endpoint or data object need to be present;
4. mappings, a dictionary of instances of the mapping API (with identifier as key, *e.g.*, "ncbi");
5. base identifier, identifier in the data, *e.g.*, "ncbi" for a data object of the NCBI Taxonomy.

We perform some checks on the input, *e.g.*, to assert if the endpoint is reachable.

This top API has methods which are universal across all data access APIs:

- available_conversions returns the possible mappings.
- construct_subgraph constructs a sub-graph based on provided entity and the graph hierarchy. This is useful for creating the data for the prediction problem.
- convert_id coverts an entity from one identifier to another.

Here, we have omitted wrappers around SPARQL queries as these are designed for users without SPARQL knowledge.

The following describes the individual data access APIs: *Chemicals.* This sub-API contains methods for chemical specific tasks:

- class_hierarchy returns the triples in the hierarchy where an entity exists. This method will query both the ChEMBL and MeSH datasets, by first converting to the appropriate identifiers.
- compounds returns all compounds.
- get_features returns a compounds' features by using PubChemPy Swain et al. 2014.

- get_fingerprints returns a compounds' binary fingerprint.
- get_names returns synonyms of compounds.
- similarity returns the similarity between a compound and a list of compounds.
- which_features returns a list of available features for a compounds.

These methods enable the mixture of the KG and other data, *e.g.*, features from PubChem.[6]

NCBI and ECOTOX Taxonomy use the same methods to access the data:

- get_division returns all taxa in a provided division (or group).
- get_rank returns all taxa at a taxonomic rank.
- get_ranks returns all available taxonomic ranks.
- get_taxa returns all taxa.

EOL Traits provides methods for abstracting SPARQL queries to access traits data:

- get_conservation_status returns the status of a taxa, *e.g.*, endangered.
- get_ecoregion provides the ecoregion of a taxa, *e.g.*, Baltic.
- get_extinct_status returns a binary extinction status.
- get_habitat returns the habitat of a taxa, *e.g.*, fresh water.

ECOTOX Chemicals is the least used API as we prefer the larger set of chemical data described above. However, this API provides methods for gathering chemical synonyms, which can be useful in alignment between sources.

Effects is arguably the center of TERA and this API provides methods to extract and work with the effect data:

1. get_chemicals returns chemicals used in at least one experiment.
2. get_chemicals_from_species returns chemicals where provided species where used.
3. get_species_from_chemicals is the opposite of above.
4. get_endpoint returns all effect endpoint of provided chemical and species.
5. get_species returns species used in at least one experiment.

[6] Paper I mentions that we don't add this to TERA as the storage requirements are humongous.

A.4 Utilities

We have defined several utilities or helper functions:

- literals_to_dict converters a KG to a dictionary on the form {entity:literals}. Useful for creating feature sets from the KG.
- tanimoto calculates chemical similarity.
- unit_parser converts units to entities in the UNITS namespace, *e.g.*, mg/L to `unit:MilliGramPerLitre`. This method also tries to remove misprints.
- unit_conversion calculates the conversion factor from one unit to another while checking that they're compliant, *e.g.*, $mass/volume$ can only by scaled. An additional argument can be provided to include the molecular mass of a chemical in the conversion factor.
- strip_namespace simply removes namespace from URI. Useful when querying Wikidata.
- SPARQL query related methods:
  - test_endpoint tests the connection to a SPARQL endpoint.
  - query_graph and query_endpoint and methods that wrap the RDFLib logic and are used in the data access APIs depending if a (local) KG or and endpoints is the data source.

A.5 TERA Construction

We have described all tools to create and access TERA. In this section, we will describe the process of creating the RDF representation of TERA which can be used by a host of semantic web tools.

A.5.1 Constructing Full TERA

The steps to create full TERA are:[7]

1. Using the data aggregation APIs, load all data from the appropriate directories. This will convert all data to graph objects (this graph is a API property).
2. Run LogMap and AML on the two taxonomies to extract mappings.
3. Use the mapping API to load LogMap and AML RDF mappings, and add the intersection of these to the NCBI Taxonomy API object as `owl:SameAs` triples (the APIs has `add` methods and will simply concatenate the two graphs).

[7] A script for creating TERA is found at https://github.com/NIVA-Knowledge-Graph/\gls {tera} along with a script for downloading the raw data.

4. Use the NCBI to EOL mapping API to load from Wikidata and again add these as `owl:SameAs` triples to NCBI Taxonomy object.
5. Use the various chemical mapping APIs to load from Wikidata and add these as `owl:SameAs` triples to ChEMBL API object.
6. Use `add` methods to concatenate all API objects into one and use the `save` method to save the graph to a specified location as RDF.

A.5.2 Constructing Prediction TERA

In Paper II, we refer to the reduced version of TERA used for prediction as KG' with parts KG'_S, KG'_C and, KG'_E. We reduce the KG to be able use it in KGEMs on reasonable hardware, moreover, disconnected parts should not greatly influence the prediction tasks.

We start with the full KG_E and only keep effect data relevant to the particular task, *e.g.*, LC_{50} prediction, resulting in KG'_E. The transformation of KG_S and KG_C to their reduced versions is then

$$KG'_S = \{(s,p,o)|(s,p,o) \in KG_S \wedge ((s,\star,e) \vee (o,\star,e)) \wedge e \in KG'_E\}. \qquad \text{(A.1)}$$

where $\star$ is any property path. Equivalent transformation is performed on KG_C to create KG'_C.