Knowledge extraction from archives of natural history collections

Stork, L.

CHAPTER 2

Background

"Whether in literary criticism or scientific investigation, the academic mind is best at taking things apart. The complementary arts of integration are far less well developed. As with any interdisciplinary pursuit, it is the bridging across disparate ways of knowing that is the constant challenge."

– Richard J. Borden, in: *Ecology and Experience*

In this chapter we present the background of the thesis. Owing to its interdisciplinary nature, we cover a number of diverse topics, ranging from *natural history*, which is the application context of this thesis (Section 2.1), to methods from two sub-fields of computer science and artificial intelligence (AI): *symbolic*, and *subsymbolic* artificial intelligence, each with their own academic legacy. We discuss both under the umbrella of the task *knowledge extraction* (Section 2.2). We close the chapter by detailing the use cases and datasets that formed the basis for analyses described in subsequent chapters (Section 2.3).

2.1 Natural History

Biodiversity research aims to understand the whole of life on earth, its evolution and the various factors that generate its diversity. The field is usually subdivided into three levels, in which diversity is measured and researched: species, genetics and ecology. In this thesis, we focus on research into the diversity of *species*. Inherent to species research is the comparison and classification of the various plants and animals that inhabit our world. In order to realise this, naturalists in the field are challenged to develop methods that moderate systematic descriptions. Expeditions to biodiverse areas allow naturalists to record organism observations and identifications, enabling them to extend, improve or challenge existing classifications.

2.1.1 Taxonomy and Nomenclature

In the first part of the 18th century, Carl Linnaeus published his *Systema Naturae*, a system that formed the basis for biological taxonomy and nomenclature. From then on, naturalists and taxonomists started to use *taxonomy* and *binomial nomenclature* for the hierarchical classification and systematic naming of organisms. Therefore, most historical records found today in museums and other institutions (38), as well as contemporary biodiversity datasets, use biological taxonomy and binomial nomenclature to classify and describe their specimens and observation records.

Taxonomy. In the *Systema Naturae*, Linnaeus presented ideas for the *hierarchical classification* of species. By his system of classification, the natural world was organised into three *kingdoms*: the *animal kingdom*, the *plant kingdom*, and the *mineral kingdom*, although his system for the classification of minerals was never widely adopted by the scientific community. Species were grouped based on shared traits into units called *taxa*, which were in turn organised hierarchically into six nomenclatural *ranks* that increasingly share more traits: *kingdom*, *class*, *order*, *genus*, *species* and *variety*, according to a subsumption relationship. For example: a common octopus is an octopod (of the order octopoda), as well as a cephalopod (of the class cephalopoda), as well as an animal (of the kingdom animalia), see Figure 2.1. More recent subdivisions that have been added over the years are *phyla*, *families* and *tribes*, and subranks such as *subspecies*, or *subtribes*.

```
Animalia [kingdom]
└── Mollusca [phylum]
    └── Cephalopoda [order]
        └── Octopoda [class]
            └── Octopodidae [family]
                └── Octopus Cuvier, 1797 [genus]
                    └── Octopus vulgaris Cuvier, 1797 [species]
```

Figure 2.1: Classification of the species *Octopus vulgaris Cuvier, 1797*, a common octopus.[1] Edges represent the "part–of" relationship.

Binomial nomenclature. *Binomial nomenclature* translates into *two-term naming system*, and was introduced to formally name species according to one system. The idea of a two-term naming system was first put forth by Linnaeus in 1753 in his work *Species Plantarum*. A systematic name in binomial nomenclature is called a *binomial name*, also known as a *scientific name*. Octopus vulgaris (Figure 2.1) is an example of a binomial name. The first of the two terms identifies the genus to which the organism belongs, and

[1] https://www.gbif.org/species/2289671

the second is called the *specific epithet*, and points to the specific species within that genus. Commonly, the binomial is followed by the name of the author who published the name, and the date when the name was published in literature. It is also common for a name to have more than one author. Figure 2.2 shows an example of a scientific species name from a field note; it dates back to 1821.

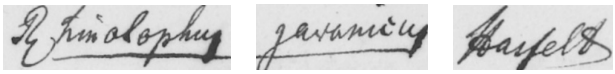

Figure 2.2: A scientific name in binomial nomenclature: (a) *Rhinolophus* (genus) (b) *javanicus* (specific epithet) (c) *Hasselt* (author of the name: *Johan Coenraad van Hasselt*)

Taxonomic Debates and Name Ambiguity. During the development of biodiversity research, methods of biological classification were continuously subject to intense discussion (39). Multiple theories emerged regarding collection practices and classification. In particular in the early nineteenth century and before, naturalists were struggling to find and agree upon one 'true' natural system. NHCs embody this search for a terminological structure which could be used to order, describe and classify nature.

The lack of consensus on biological classifications, as well as the challenges that came with the publication of scientific names—the very act of bringing home the actual observation records as well as tensions that arose through top-down policy-making (16)—resulted in species descriptions that are challenging to analyse within the present scientific paradigm, but also within collections themselves: (i) biological classification systems implied in field books cannot be directly mapped to present taxonomies (ii) taxa have various types of synonyms and homonyms within collections, see Figure 2.3, and Figure 2.4, and (iii) scientific names shift between genera and species (39; 29; 40).

```
Scotophilus kuhlii temminckii (Horsfield, 1824) [current name]
├── Vespertilio temminckii Horsfield, 1824 [synonym]
└── Vespertilio fulvus Kuhl & Van Hasselt [synonym]
```

Figure 2.3: Synonyms of the current taxon `Scotophilus kuhlii temminckii`. Courtesy: E. Gassó Miracle (2016)

Due to this, *taxonomic referencing* (resolving historic scientific names to current scientific names) of historical observation records, as well as establishing links between scientific names in general, are important processes in species research. Thomer et al. (2012) (35) describe taxonomic referencing as the process of linking a legacy name to its valid scientific name. They mention the process is analogous to that of georeferencing for localities.

Similarly to georeferencing, the process helps to integrate data related to the same entities, as well as separate data from unrelated ones.

```
Orestias elegans [hemihomonym]
├── Orestias elegans Garman, 1895 [accepted name]
└── Orestias elegans Ridley, 1887 [accepted name]
```

Figure 2.4: Hemihomonymy, an accepted form of homonymy where two species have the same name, but come from distinct kingdoms: the first referring to an animal (a pupfish), and the second to a plant (an orchid).[1]

2.1.2 Multimodal Field Observations

Early field observations exist in natural history museums as physical **specimens**, accompanied by archival material such as handwritten **field books**, and **illustrations**. Museums keep historical records for comparison with contemporary records. Many collections date back 100 years or more (3). Through more recent next-generation techniques such as photogrammetry, laser scanning, and computed tomography, rich digital representations of specimens as well as manuscripts (e.g., the Dutch Metamorfoze programme[2]) can be created (7). Below we discuss each modality and its characteristics.

Specimens. Specimens that are commonly kept in natural history museums are: fluid-preserved whole organisms and organism parts, frozen tissues, pinned dried insects, pressings and seeds or spores of plants, dried skins, skeletons, nests of birds and eggs of birds and insects (3). Figure 2.5 shows a skeleton of a *Pteropus vampyrus* from the Naturalis Biodiversity Center (NBC).

It is common for such specimens to be accompanied by labels containing metadata regarding the specimen, such as the name of the collector, the scientific name, location and date of collection, although metadata are often limited to a location or naturalist that performed the identification or collected the specimen. In most cases, labels include scientific names, but do not record any scientific context (29), for instance regarding the literature used for classification. As alternative views on taxonomy exist, mentioned earlier in Subsection 2.1.1, one name can point to two very distinct species. Linking the physical specimens to observation records becomes crucial. When a specimen is accompanied by a record of the organism's latent, faded or internal traits and attributes (e.g. behaviour, coat colour, or intestines), identification of the preserved specimen can be revisited and its value for use in long-term scientific studies therefore increases.

[1] https://species.wikimedia.org/wiki/Orestias_elegans
[2] https://www.metamorfoze.nl/kennis-onderzoek/lexicon/preservation-imaging

Figure 2.5: A specimen from the *Naturalis - Zoology and Geology catalogues.*[1]

Field Books. Since the onset of field work in biodiversity expeditions, naturalist have been manually recording species observation data. The containers that preserve these observation records are fittingly named *field books* (41), see Figure 2.6. They provide rich descriptions of species-specific traits such as measurements of specific organs or other body parts, the environmental conditions in which organisms are discovered and information about how organisms were collected, classified and described. Because of this, field books provide rich insight into the daily practices, methods, and results of the research field (33).

The interpretation of historical field records is an intricate and complex task. We demonstrate this complexity with the use of an example. The field note shown in Figure 2.6 describes an occurrence of an organism identified as the *Titthaecheilos javanicus Nobis* (right page, upper left corner).

Nobis is latin for *by us*. The space behind the binomial name is reserved for the author of the species. Therefore, the term *by us* refers to the authors of the field book: according to them, they were the first ones to have identified, described and named the organism. The name *Titthaecheilos javanicus* has, however, never been published in any classification system. Most likely, the name served as a basionym[2] for the published name *Pteropus titthaecheilus Tem.* (upper right corner) believed to have been added to the field note in Leiden, years later, by *Jacob Coenraad Temminck*, a dutch zoologist and museum director. The name can be found in older classification systems as the name *Pteropus*

[1] `https://data.biodiversitydata.nl/naturalis/specimen/RMNH.MAM.33245.a` Images free of known restrictions under copyright law (Public Domain Mark 1.0).

[2] A basionym is a synonym on which a later scientific name is based.

titthaecheilus (Temminck 1825). In turn, that name served as a basionym for the accepted name *Cynopterus titthaecheilus (Temminck, 1825)*.

Moreover, below the scientific name we find another name type: *Buitenzorg*, a place name. Historically, Buitenzorg was the name for the large city of *Bogor*, close to the capital of *Java*, *Jakarta*. The city houses the largest botanical garden in the world, the botanical garden of *Bogor*, which served as the headquarters of the NC. Last, the field note is written in a distinct, historical style, and mixes three languages: the note starts in Dutch, continues in German, and ends in Latin.

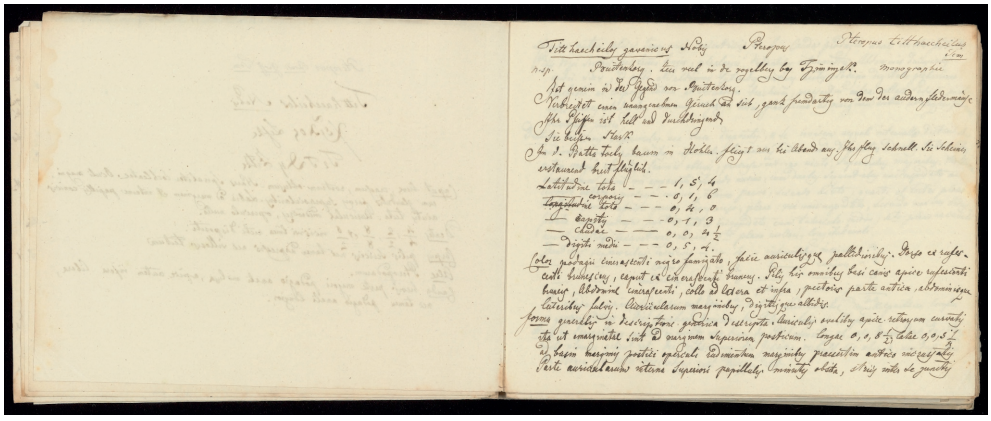

Figure 2.6: A page from the annotated field book describing the species *Titthaecheilos javanicus Nobis*. *Pteropus titthaecheilus Tem* (upper right corner) is believed to have been added later in Leiden by *Jacob Coenraad Temminck*, `http://viaf.org/viaf/69703180`, a dutch zoologist and museum director. The written annotation is thus an additional identification of the observed organism. Collection Naturalis Biodiversity Center, MMNAT01_AF_NNM001001033_013.[1] Image free of known restrictions under copyright law (Public Domain Mark 1.0).

Illustrations. Historically, collectors were accompanied by professional illustrators, who produced detailed drawings of organisms, as shown in Figure 2.7. The habitus illustration—a scientific illustration of a species' physical appearance—was the most important medium to convey a species' characterising traits to other scientists. In illustrations, scientists are capable of delineating and highlighting minuscule details, often more so than photographs. Habitus illustrations were routinely and abundantly created and commonly served as examples for the description of newly discovered species, so-called holotypes. Additionally, they sometimes recorded the habitat or behaviour of an organism.

[1] `https://dh.brill.com/nco/view/nco_NNM001001033_013/makingsense`

In illustrations, the background (natural habitat) is often omitted and species are depicted in the form of collages of multiple (smaller) depictions of their external and internal anatomy (e.g., bones, organs, limbs). These appear in a combination of various views (e.g., frontal, dorsal, lateral). Moreover, illustrations exist as rough pencil sketches and/or detailed colour drawings and commonly contain handwritten captions. Often, they are published in digital archives with limited or no identifications. When illustrations contain captions with handwritten *historical names*, these are mostly unpublished or obsolete within today's taxonomy. The left illustration in Figure 2.7 says *Asterias tesselatus (Lamarck, 1816)*, an unaccepted name, and *Asterias granularis Kuhl*, an unknown name. The current accepted name of the starfish is the *Goniaster tessellatus (Lamarck, 1816)*. The middle photograph has some unreadable text in the upper right corner, and a pencil annotation that most likely reads *Noae Lam.*, appearing to refer to a genus published by Jean-Baptiste de Lamarck.[1] The current accepted name of the species is *Arca noae (Linnaeus, 1758)*.

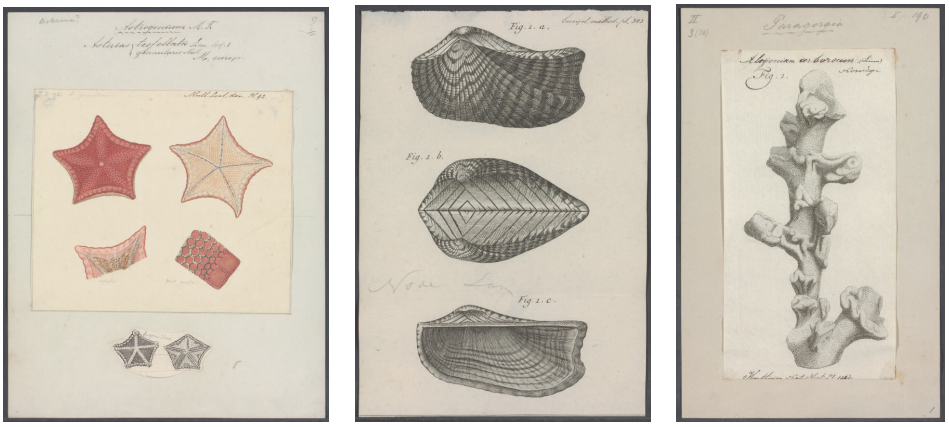

Figure 2.7: Zoological illustrations from Iconographia Zoologica online[2] (best viewed in colour). Images free of known restrictions under copyright law (Public Domain Mark 1.0).

The identification of an organism from a photograph or illustration without reference to a scientific name, is a complex and delicate task, even for domain experts (42).

2.2 Knowledge Extraction

In this section, we introduce the preliminaries used throughout this thesis. Our approach employs techniques from subsymbolic AI (e.g., *computer vision*) well as symbolic AI (e.g., *Semantic Web*) , for the purpose of knowledge extraction.

[1] Jean-Babtiste de Lamarck, a French naturalist. URI: `https://viaf.org/viaf/41849820/`
[2] `https://bijzonderecollecties.uva.nl/gedeelde-content/beeldbanken/iconographia.html`

We take the definition of the term *information extraction* from (43), and use this as a red thread that weaves through the thesis:

Definition 2.1. *"Information extraction is the process of extracting information and turning it into structured data. This may include populating a structured knowledge base with information from an unstructured knowledge source. The information contained in the structured knowledge base can then be used as a resource for other tasks, such as answering natural language queries, or improving on standard search engines with deeper or more implicit forms of knowledge than that expressed in the text".*

Knowledge extraction is a form of information extraction. It uses similar methods, but the main criteria is that results of the extraction process are structured according to formalised semantics such as taxonomies or ontologies (which we will discuss in Subsection 2.2.2).

Examples of information and knowledge extraction tasks are *semantic annotation* and named entity recognition and classification (NERC), both described earlier in Subsection 1.1.3, in which *ontologies*—formal specifications of concepts and their relationships—play a large role in the information extraction process. A similarity between these tasks and our work is that we leverage domain specific ontologies and taxonomies for knowledge extraction. One major distinction between these and our work is that we extract knowledge from *digital images*, whereas commonly, an intermediate step transforms the content of images to digital text, to which then information extraction is applied.

In the following sections, we will detail the most important notations, concepts and techniques used throughout this thesis, which fall under the umbrella of **machine learning**, used to automatically extract patterns (subsymbolic AI, Subsection 2.2.1) and **knowledge representation and reasoning (KRR)**, used to organise the patterns semantically (symbolic AI, Subsection 2.2.2).

2.2.1 Machine Learning

In machine learning, *learning algorithms* learn from data to perform a certain task: e.g., hypothesise to which category y or target value t a sample belongs. The type of learning algorithm or **model** that is used depends on a number of things, such as (i) the **structure** of the **data**, (ii) the **task**, (iii) the kind of **experience** the models are allowed to have during the learning process (simulating certain real-world situations) (44).

Data Structures. Examples of data types often used in machine learning tasks—and that we will use in this thesis—are digital images, sequences such as sentences.

- **Digital images** are digital captions of scenes or pictorial materials. They represent a coherent collection of focus points of light rays coming from an object. A digital image divides the real image into a grid of real numbers, called pixels, which discretise properties of the underlying areas such as *brightness* and *hue*. The process of digitisation of the spatial domain is called *sampling*. Discretising the range in which these real numbers fall is called *quantisation* (45). A gray-scale image is a 2-dimensional (2D) digital image, of which each value represents a pixel that samples the brightness of that pixel. Commonly, the brightness range is encoded in 256 (2^8) levels (values from 0 to 255), corresponding with an 8-bit discretisation. RGB images sample three values per pixel, also called *channels*: the brightness of the red, green and blue values. Similarly to gray-scale images, these three channels are commonly encoded in 256 levels. Both sampling and quantisation depend on the imaging device that is used. The resulting multidimensional array of real numbers can be stored and handled by a digital computer.

  We define a digital image (either gray-scale or RGB), with m rows and n columns, as follows:

  Definition 2.2. *A gray-scale digital image* $\mathbf{X}$ *is a 2D numerical array, or* matrix*, with* $x_{ij} \in \mathbb{R}$ *being the gray value of the pixel in the* i*-th row of and the* j*-th column, see the matrix representation in Equation* (2.1) *below.*

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,n} \end{bmatrix} \tag{2.1}$$

  An RGB image is a multi-channel digital image in which each channel represents a colour layer. This can be considered a 3D numerical array, or tensor*, with* $x_{ijk} \in \mathbb{R}$ *being the value of the pixel in the* i*-th row, the* j*-th column, and* k*-th colour channel. A tensor representation of an RGB image is shown in Equation* (2.2) *below.*

$$\mathrm{X} = \begin{array}{l} \begin{bmatrix} x_{1,1,1} & x_{1,2,1} & \dots & x_{1,n,1} \\ x_{2,1,1} & x_{2,2,1} & \dots & x_{2,n,1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1,1} & x_{m,2,1} & \dots & x_{m,n,1} \end{bmatrix} \\ \begin{bmatrix} x_{1,1,2} & x_{1,2,2} & \dots & x_{1,n,2} \\ x_{2,1,2} & x_{2,2,2} & \dots & x_{2,n,2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1,2} & x_{m,2,2} & \dots & x_{m,n,2} \end{bmatrix} \\ \begin{bmatrix} x_{1,1,3} & x_{1,2,3} & \dots & x_{1,n,3} \\ x_{2,1,3} & x_{2,2,3} & \dots & x_{2,n,3} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1,3} & x_{m,2,3} & \dots & x_{m,n,3} \end{bmatrix} \end{array} \tag{2.2}$$

- **Sequences** are digital representations of values that are meaningful in a certain arrangement, such as sentences, digital texts or even sequences of images that represent

words in a sentence.

We define a sequence of values as:

Definition 2.3. *A sequence* ***s*** *is a finite set of values* $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)})$ *that are tightly interrelated, where* t *indicates the length of the sequence, and* $\mathbf{x}^{(i)}$ *the* $i-$*th value of the sequence.*

In classical machine learning, "raw" data are first transformed into *features* (numerical representations of raw data) that represent the variations in the data well. This process is called *feature engineering*.

Machine Learning Tasks. Below, we discuss two machine learning tasks used in this thesis, but many more exist (e.g., regression or clustering).

- **Classification:** In a classification task, a machine is asked to infer to which of k categories some input belongs. The learning algorithm is asked to produce a function (*a model*) $f : \mathbb{R}^n \rightarrow \{1, \ldots, k\}$ (44). An example of this is image classification, in which an image $\mathbf{X}$ is mapped to the category $y \in \{1, \ldots, k\}$ to which the machine thinks the image belongs, each number representing a class. For example, an image $\mathbf{X}$ of a *bear* gets mapped to the label $y^{(i)}$ that encodes the *bear* class, or an image of the word *Pteropus* gets mapped to the label $y^{(i)}$, representing the class *Pteropus*. A machine learning model trained for classification generally produces a decision boundary, see Figure 2.8, that separates data from distinct classes (in Figure 2.8, the decision boundary separates instances from the red class from that of the black class). The model classifies a new data point as belonging to a certain class by calculating on which side of the boundary it lies.
- **Classification with structured output:** In *classification with structured output*, a machine is asked to produce, given some input, several values that are all *tightly interrelated*: a sequence **s**. Examples are (i) image captioning, where a machine receives an image as an input and outputs a sentence that describes the image, or (ii) NERC, where a machine receives a sentence, and returns the same sentence with its named entities annotated with terms from a structured knowledge base.

Learning from Experience. Below, we discuss two types of learning strategies that vary in the amount of experience they are allowed to have during the learning process.

- **Supervised learning:** Most types of learning algorithms get to "see" all training examples $x \in \mathbb{R}^n$ and their labels $y \in Y^{tr}$ (classification) or targets $t \in \mathbb{R}^n$ (regression), and are therefore supervised in the sense that they are instructed as to what the output

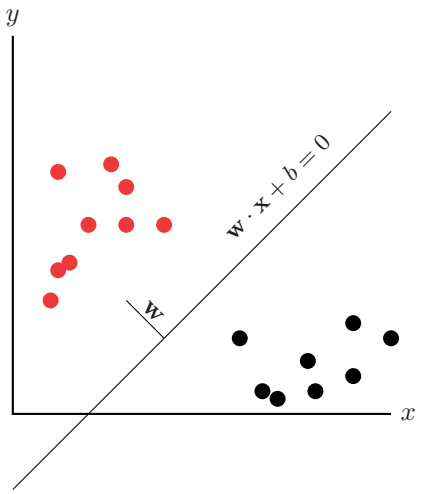

Figure 2.8: Example decision boundary of a binary classifier.

for a certain input should be. From the training data, the machine learning algorithm is asked to learn the mapping between input and output, and use it for extrapolation to new data from a test-set.

- **Zero-shot learning (ZSL):** ZSL is an extension of supervised learning in which the test-set represents a distinct set of classes $y \in Y^{ts}$, for which $Y^{tr} \cap Y^{ts} = \emptyset$. The goal of ZSL is that a classifier learns representations for data from a set of *seen* classes Y^{tr} (seen by the algorithm) with medium to large samples, which are then transferred to classify classes from another set of *unseen* classes Y^{ts}, for which no or small samples are available for training.

  An appealing characteristic of ZSL techniques is that it is possible to exploit data from auxiliary data sources to share representations between classes, and hereby push the boundaries of automated recognition for a specific problem. As with regular supervised learning, it can be difficult in some cases to control which features are shared.

Other popular types of learning are unsupervised learning and semi-supervised learning, but these are out of the scope of this thesis.

Deep Learning Models. Deep learning is a subfield of machine learning that brings forth a specific type of machine learning models, called *deep (artificial) neural networks (DNNs)*. DNNs are able to learn representations of data from data (46), replacing part of the feature engineering pipeline. These learned representations then allow computers to perform a machine learning task. Artificial neural networks (ANNs) are models that are

inspired by the human brain, as they are trained to strengthen and weaken connections between input variables, much like the brain's networks of neurons. DNNs are types of ANNs that are trained to learn *deep*, *hierarchical* representations of data—with multiple levels of abstraction (46). For the purpose of classification, they define a mapping between an input array $\mathbf{x} = (x_1, \ldots, x_n)$ and an output y, $y = f(\mathbf{x}; \boldsymbol{\theta})$, and learn the value of parameters $\boldsymbol{\theta}$ that defines the best function approximation.

Below we discuss types of DNNs that are used in this thesis.

- **Multi-layer perceptrons (MLPs):** An MLP is an example of the simplest type of DNN, see Figure 2.9. It is a type of feed-forward neural network, meaning that the multiplications flow 'forward' in one direction through the network. Although the network represented here has one level of abstraction (one *hidden layer*), DNNs usually have many. By adding multiple hidden layers, we increase the network's *depth*. In Figure 2.9, each connection represents a multiplication with a weight. In this figure, there are two weight matrices $\mathbf{W}^{(i)}$, the superscript denoting the i-th weight matrix, one between the input and the hidden layer, and one between the hidden layer and the output. We call these layers *fully connected*, as each node in one layer is connected with every other node in the next layer, i.e., has its own weight.

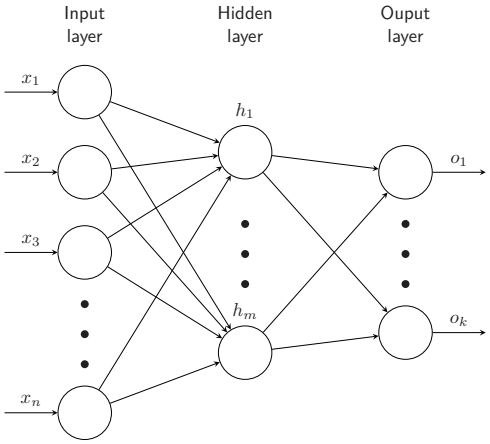

Figure 2.9: An example of an MLP with one layer of abstraction. The network has an n-dimensional input, m hidden nodes and k outputs (classes).

See Equations (2.3) to (2.5) for the network's mapping between $\mathbf{x}$ and y. Equation (2.3) and (2.4) are called node *activations*. They show an additional parameter $b_j^{(i)}$, which stands for *bias* and serves to shift the activation function and thereby the classification boundary by adding a constant.

$$h_i = \sum_j w_{ij}^{(1)} x_j + b_j^{(1)} \tag{2.3}$$

$$o_i = \sum_j w_{ij}^{(2)} x_j + b_j^{(2)} \tag{2.4}$$

$$y = argmax(\mathbf{o}) \tag{2.5}$$

While an MLP is a linear model, most ANNs employ at least one layer in which a non-linearity function g, called *activation function*, is applied to the activation of each of the neurons in a hidden or output layer, as in Equation (2.6). Common activation functions are the *ReLU* or the *softmax* functions.

$$h_i = g\left(\sum_j w_{ij}^{(1)} x_j + b_i\right) \tag{2.6}$$

The activation of the last layer (often a softmax function applied to the activation of the last layer) produces a distribution over output classes, also called *confidence values*, which correspond to the distance of an instance to the decision boundary, and thereby how confident the classifier is about that class being the one represented in the data. This intuitively makes sense, as when our instance lies very close to the decision boundary (or boundaries for multi-class classification), it is more likely to actually belong to the other class than when it's further away.

- **Convolutional neural networkss (CNNs):** Whereas the MLP is used for processing 1D arrays, CNNs are used for processing grid data. They are often used in *computer vision* for classification of images. Characteristically, they use a mathematical operation called a *convolution*. In image classification, a convolution is a function that extracts features from regions of pixels in an image (47). The kernel $\mathbf{K}$, a small matrix of integers, acts as a sliding window that is moved over the digital image $\mathbf{X}$ and produces a weighted average with the underlying pixels, see Equation (2.7) (44).

$$S(i,j) = (\mathbf{X} * \mathbf{K})(i,j) = \sum_m \sum_n \mathbf{X}(i-m, j-n)\mathbf{K}(m,n). \tag{2.7}$$

Hence, instead of acting on the full input, kernels act on subregions of an image. Digital images exist of large 2D or 3D arrays, so employing kernels makes networks easier to train: parameters are shared over multiple regions of the input, so a much smaller number of parameters need to be optimised then when the layer would be fully connected.

Convolving regions in an image with kernel $\mathbf{K}$ results in an output matrix called a *feature map*, which gives an indication of whether or not the features in the kernel are present in certain regions of the image: similar regions produce similar output values. CNNs are types of DNNs as they stack layers of convolutional operations to extract image features on various levels of granularity, from fine-grained features such as *corners* and *edges* to coarser, class-specific features such as *eyes*, *feathers*, a *beak*—even though these coarser features are never that clear-cut. Similarly to an MLP, classification happens in the last fully connected layer, see Figure 2.9. The array of 2D feature maps is re-arranged to a 1D array and act as input to a fully connected layer. Often, engineers employ some fully connected layers before the final classification layer.

- **Recurrent neural networks (RNNs):** RNNs (48) are other types of DNNs for processing sequences of values $\mathbf{t} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)})$. Like CNNs, they share weights and perform a single computation multiple times over a different part of the input, here called $\mathbf{x}^{(t)}$ with time step t. Different from CNNs, they have a *recurrence* property, which means they use output at time t to serve as auxiliary input to a hidden layer at the next time step $t+1$. Effectively, the recurrence property serves as a *memory* that uses past computations $\mathbf{h}^{t-1}$ to influence present computations $\mathbf{h}^t$. Their basic operations are detailed below in Equation (2.8) and (2.9):

$$\mathbf{o}^t = f(\mathbf{h}^t; \boldsymbol{\theta}) \tag{2.8}$$

$$\mathbf{h}^t = g(\mathbf{h}^{t-1}, \mathbf{x}^t; \boldsymbol{\theta}) \tag{2.9}$$

where $\mathbf{o}^t$ is the output at time t, $\mathbf{h}^t$ the state of the hidden layer at time t, and $\mathbf{x}^t$ the input array at time t. f and g serve as activation functions.

Long short-term memory networks (LSTMs) are types of RNNs that overcome some of the issues that occur with regular RNNs. They have a *bilateral* variety, the bilateral long short-term memory network (BLSTM), that can additionally use *future* computations $\mathbf{h}^{t+1}$ to influence present computations $\mathbf{h}^t$.

- **Prototypical neural networks:** Prototypical networks are networks developed for low-shot learning strategies such as few-shot learning (FSL) or ZSL (49). They compute M-dimensional class representations $\mathbf{c}_k \in \mathbb{R}^M$ called *class prototypes*. In contrast to the other DNNs that we discussed, classification does not happen based on a distribution (softmax activation) over the last fully connected layer. Instead, the last fully connected layer maps instances to a metric space, and a distribution over distances

from an instance to class prototypes is produced. Example distance functions are *euclidean distance*, see Equation (2.10), and *cosine similarity*, see Equation (2.11).

$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{2.10}$$

$$\cos(\boldsymbol{p}, \boldsymbol{q}) = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} (p_i)^2} \sqrt{\sum_{i=1}^{n} (q_i)^2}} \tag{2.11}$$

To calculate prototypes for FSL, *support points* (example datapoints) are mapped to a metric space, and per-class averages of the resulting embedded support points are calculated, see Equation 2.12. In Equation 2.12, $\mathcal{S}_k$ refers to the set of support points for class k, and $\mathbf{c}_k$ refers to its calculated prototype. We further refer to this metric space by the term *prototype space*.

$$\mathbf{c}_k = \frac{1}{|\mathcal{S}_k|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_k} f_\phi(\mathbf{x}_i) \tag{2.12}$$

For ZSL, Snell et al. (49) mention that rather than embedding support points in prototype space, prototypes can be constructed by embedding auxiliary information, such as class embeddings in the form of attribute annotations (50; 51), in prototype space. Attribute embeddings encode whether a certain attribute—from a set of predefined attributes—is present for a specific class. Attribute embeddings can be either binary or continuous, e.g., *{wing: 0.1, red: 0.4, tail: 0.7}*.

Training and Evaluation. After model selection, the model's parameters $\boldsymbol{\theta}$ are learned through iterative minimisation of the *training error*. One iteration commonly consists of minimally three basic steps.

1. The model is applied to a batch of training data (data from the *training-set* $\mathcal{T}^{tr}$).
2. A *loss function* is applied to the output of the model, which calculates the training error—a function over the difference between the output y and the desired output $\hat{y}$.
3. The training error is propagated backwards through the model using the *backpropagation algorithm*, and parameters $\boldsymbol{\theta}$ are adjusted via an optimisation algorithm (such as *gradient descent*).

How well the model performs on real-world data should be evaluated on a dataset that is separate from the training-set, the *test-set*, $\mathcal{T}^{ts}$. It often happens that the representations learned by the model too closely fit the variation in the training-set, and will therefore generalise poorly to new data. This phenomenon is called *overfitting*, and can for instance happen when samples in the training-set are too small to obtain a good representation, or when there are too many parameters in the model, causing it to learn too much of the variance in the data. A metric that is most used for classification is the *average accuracy*, see Equation (2.13) (in percentages).

$$\textit{Accuracy} = \frac{n \text{ correct predictions}}{n \text{ total predictions}} * 100 \tag{2.13}$$

The average accuracy metric is not always the best choice, as it does not correctly portray the predictive power of a model, especially when data are imbalanced (52). Let us first consider a binary classification problem, and a dataset with a uniform distribution over its classes. We produce a naive model, let us call it model g, that for every input predicts the class c^m, the class in which the majority of the data resides, i.e., the *majority guess*: $g : \mathbb{R}^d \rightarrow c^m$. The estimated average accuracy on the test-set would already be as high as 50%. Imagine now a classifier h, that is trained on a 5-class classification problem with a similar uniform distribution over its classes. If such a classifier h produces an average accuracy of 50%, it will have learned a much better data representation than our naive binary classifier, even though the average accuracy produced would be equal.

The accuracy metric is especially vulnerable to bias that skewed data introduces. If we would apply our naive classifier g to a dataset where 90% of the data are of class 0 and the rest of class 1, we would obtain an average accuracy of 90%.

Further details of specific models, learning strategies, evaluation metrics and other, can be found in the respective chapters.

2.2.2 Knowledge Representation and Reasoning

The field of knowledge representation and reasoning (KRR) is quite extensive, so we limit ourselves to techniques for structuring data and data about data (metadata), with a focus on the representation of data in the form of **knowledge graphs**. Through the use of **schemas** and **ontologies**, which impose constraints or assign attributes to data, new knowledge can be inferred (*reasoning*). We furthermore discuss the principles of Linked Data (LD), which allow knowledge graphs served on the Web to link together, forming a Web of semantic data, called the **Semantic Web**.

Structured data, in contrast to *unstructured data*, are data that are structured according to some data model, and can therefore be interpreted by machines. Unstructured data, such as free text, can be made machine understandable by adding structure to capture the implicit semantics. Below we define what it means to make implicit semantics of data accessible to machines (53), turning data into machine-understandable *knowledge*.

We distinguish three levels for structuring data, that vary based on their capability to express implicit semantics (54):

- **Controlled vocabularies:** Controlled vocabularies include shared terminologies and nomenclatures and define terms to formally describe concepts within a domain.

  In the biodiversity domain, for instance, community collaboration is used to create shared knowledge representations, in order to make effective use of existing data (36). For broad-scale analyses, biodiversity information must be readily available in digital form, published as FAIR data. Below, in listing 2.1, we show a piece of Extensible Markup Language (XML), taken from the Simple Darwin Core documentation[1] that structures biodiversity data according to the Darwin Core (DwC)[2] standard, a glossary of terms (properties) for the description of biodiversity records. Such a basic form of structuring data allows intelligent machine search over data, for example the aggregation of scientific names across distributed collections using the term `dwc:scientificName`.

```
<?xml version="1.0" encoding="UTF-8"?>
<SimpleDarwinRecordSet
    xmlns="http://rs.tdwg.org/dwc/xsd/simpledarwincore/"
    xmlns:dc="http://purl.org/dc/terms/"
    xmlns:dwc="http://rs.tdwg.org/dwc/terms/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <SimpleDarwinRecord>
        <dwc:basisOfRecord>Taxon</dwc:basisOfRecord>
        <dwc:scientificName>Centropyge flavicauda Fraser-Brunner 1933
        </dwc:scientificName>
        ...
        <dwc:kingdom>Animalia</dwc:kingdom>
        <dwc:phylum>Chordata</dwc:phylum>
        <dwc:class>Osteichthyes</dwc:class>
        <dwc:order>Perciformes</dwc:order>
        <dwc:family>Pomacanthidae</dwc:family>
        <dwc:genus>Centropyge</dwc:genus>
        <dwc:specificEpithet>flavicauda</dwc:specificEpithet>
        <dwc:taxonRank>species</dwc:taxonRank>
        ...
    </SimpleDarwinRecord>
</SimpleDarwinRecordSet>
```

Listing 2.1: A piece of simple darwin core

[1] `https://dwc.tdwg.org/simple/`
[2] `https://dwc.tdwg.org/`

Other than an unique definition of terms within a domain, the XML document does not capture a lot of meaning that can be interpreted by machines. A machine does not know how the term Centropyge relates to the term flauvicauda, or would not know the difference between the specific epithet flauvicauda, part of the genus Centropyge, and the epithet flauvicauda belonging to a different genus (should such a scientific name exist).

- **Taxonomies** extend controlled vocabularies with "is–a" (subsumption) relationships between terms and thus add hierarchy to 'flat' controlled vocabularies.

  A biological taxonomy, used to structure the scientific name in Figure 2.10, shows an example of how pieces of data can be related through the subsumption relationship, where a class lower in the hierarchy is connected to the one above it using the *is–a* relationship.

```
Animalia
└─ Chordata
   └─ Osteichthyes
      └─ Perciformes
         └─ Pomacanthidae
            └─ Centropyge
               └─ Centropyge flavicauda (Fraser-Brunner 1933)
```

Figure 2.10: A hierarchy of the species Centropyge flavicauda (Fraser-Brunner 1933), where the edges (from top to bottom) refer to the "is–a" subsumption relationship

- **Schemas and Ontologies** further extend taxonomies by distinguishing (hierarchically organised) types and properties (relationships), and allow the modelling of constraints, axioms and rules.

  Ontology in philosophy is the study of existence. More specifically, the study concerns itself with questions that relate to what types of entities exist, and how they relate to one another. In computer science, an *ontology* refers to a data structure that can be processed by machines (11; 55):

  Definition 2.4. *An ontology is a formal, explicit specification of a shared conceptualisation (56).*

  - formal: *an ontology has well-defined syntax and semantics,*
  - explicit: *an ontology can be represented and processed algorithmically*
  - shared: *an ontology is agreed upon in a community and facilitates communication between its member agents, and*
  - conceptualisation: *an ontology presents a model of the real world*

Similarly to philosophy, an ontology in computer science consists of a formally defined set of terms, and relationships (properties) that define how the terms are related (53). We will denote these consistently in the same script throughout this thesis: e.g., `class` for classes, *property* for properties, `instance` for instances of classes, and "*literal*" for literals (a value of some datatype).

Knowledge graphs use ontologies and database schemas to structure data according to a directed graph data model. We use the following definition:

Definition 2.5. *A knowledge graph is a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities (57).*

The Semantic Web. The *Semantic Web* is a network of shared, structured data and metadata. It is based on directed labelled graphs as data models for resources and their relationships, and IRIs such as HTTP URLs[1] for the description of these resources and relationships. The Resource Description Framework (RDF) is the World Wide Web Consortium (W3C)[2] recommended data format for graph data on the Semantic Web. The RDF data model uses triples for the description of resources, in the form:

$$\langle subject, predicate, object \rangle \tag{2.14}$$

An example being: $\langle$`:img.jpg`,*:creator*,"*Heinrich Kuhl*"$\rangle$. Each such triple forms a single arc in a directed labelled graph.

RDF Schema[3] and the Web Ontology Language (OWL)[4] are formalisms that provide vocabularies for structuring knowledge for various levels of expressiveness. Properties of terms can be used for *reasoning* over data, i.e., inferring new facts from a set of asserted axioms.

A simple example term that can be used to infer new facts from assertions is the term `owl:TransitiveProperty`, and it is defined as follows:

Definition 2.6. *A transitive relation $\xrightarrow{x}$ is a relation specifying that if A $\xrightarrow{x}$ B and B $\xrightarrow{x}$ C, then A $\xrightarrow{x}$ C.*

[1] Web-resolvable URIs

[2] The World Wide Web Consortium (W3C) is an international community for the development of standards on the Web. `https://www.w3.org/Consortium/`

[3] `https://www.w3.org/TR/rdf-schema/`

[4] `https://www.w3.org/OWL/`

Instances of the owl class `owl:TransitiveProperty` inherit this transitive property. *rdfs:subclassOf*, for example, is an instance of `owl:TransitiveProperty`. If we define each term from Figure 2.10 as a class that is connected to the class above it with the property *rdfs:subclassOf*, machines can infer from this statement that the class `Centropyge` is also a subclass of every class above it, such as the class `Chordata`.

The SPARQL Protocol and RDF Query Language (SPARQL)[1] is one of the query languages with which RDF graphs can be queried, making use of their graphical structure.

Through shared formalisms, distributed directed labeled graphs on the Web are linked together. The collection of directed labeled graphs on the Web are referred to as Linked data. Tim Berners-Lee, one of the inventors of the Web, suggests a 5-star scheme with which to deploy Linked Data, in which each step assumes employment of the previous step(s):[2]

★☆☆☆☆ *publish data on the Web in any format (e.g., PDF, JPEG) accompanied by an explicit Open License (expression of rights).*

★★☆☆☆ *publish structured data on the Web in a machine-readable format (e.g., XML).*

★★★☆☆ *publish structured data on the Web in a documented, non-proprietary data format (e.g., CSV, KML).*

★★★★☆ *publish structured data on the Web as RDF (eg Turtle, RDFa, JSON-LD, SPARQL)*

★★★★★ *In your RDF, have the identifiers be links (HTTP URLs) to useful data sources.*

These also relate to the Linked Data principles, mentioned earlier in Subsection 1.1.3. Serving data to the Web of Linked Data has many benefits: (i) the adoption of HTTP URIs for the representation of entities (such as named entities) ensures more accurate content descriptions and thereby allows for the computational disambiguation of terms, (ii) through the use of shared conceptualisations, data become interoperable, and (iii) through use of Semantic Web services, data can be federated and integrated between distributed collections (11).

2.3 Data Sources and Use-Cases

In this section, we discuss contemporary online biodiversity data sources relevant for this thesis (Subsection 2.3.1), and detail two NHC use-cases (Subsection 2.3.2) used for

[1] `https://www.w3.org/TR/sparql11-query/`

[2] `https://dvcs.w3.org/hg/gld/raw-file/default/glossary/index.html#x5-star-linked-open-data`

analyses in further chapters.

2.3.1 Data Sources

- **The Global Biodiversity Data Facility (GBIF)** is an international network and data infrastructure for biodiversity data on the Web (58). Through the use of data standards, the organisation has pulled together hundreds of millions of species occurrence records. These include record types from multiple sources, such as, museum specimens, scientific expedition data and photos taken by amateur naturalists. GBIF provides an API, allowing data to be downloaded as *Darwin Core archives*,[1] a Biodiversity Information Standards (TDWG) standard. Additionally, they provide other resources such as the GBIF backbone taxonomy (59), a single taxonomy that organises all names included in GBIF according to one taxonomic system, which integrates information from external resources such as the Encyclopedia of Life (EOL),[2] the International Union for Conservation of Nature and Natural Resources (IUCN)[3] and GenBank.[4] It includes knowledge such as whether a name is accepted, what synonyms it has, what its higher classifications are.
- **The Biodiversity Heritage Library (BHL)**[5] is an online library that provides open access to biodiversity heritage literature from all over the world (13). The library has employed OCR and automatically recognises scientific names in order to improve access to the printed literature through full-text search.
- **iNaturalist** (60), is a citizen science project,[6] that allows amateur naturalists to upload photographs of organism encounters in the wild together with an identification and geo-location. For their mobile application, they employ image recognition to help naturalists with the identification of observed organisms.

2.3.2 Use Cases

- **Committee for Natural History of the Netherlands Indies ("Natuurkundige Commissie voor Nederlands-Indië") (NC):** The NC was founded by King William I of the United Kingdom of the Netherlands. Consisting of a group of naturalist, draftsmen and preparators[7] from the Netherlands as well as German-speaking countries

[1] http://rs.tdwg.org/dwc/
[2] https://eol.org/docs/what-is-eol
[3] https://www.iucn.nl/over-iucn-nl
[4] https://www.ncbi.nlm.nih.gov/genbank/
[5] https://www.biodiversitylibrary.org/
[6] https://www.inaturalist.org/
[7] In the field of natural history preparators are those responsible for preparing plants and dead animals so that they could be used for research. Those preparing only animals are also known as taxidermists

and France (16), the committee was sent to the Indonesian archipelago. Their primary task was the collection of information on natural resources in the Dutch Indies. In addition, they were deployed to observe and describe the local flora and fauna. As a result, many specimens, biological illustrations and observation descriptions were brought back to the Netherlands for closer investigation, with the aim to publish results on the natural diversity of the Dutch Indies.

Currently, the physical collection is stored at the Naturalis Biodiversity Center in Leiden. In 2008 the archival part of the collection was digitised (scanned through the Metamorfoze programme[1]), leading to a digitised collection of roughly 8,000 field book pages, and 2,000 illustrations.

- **The Iconografia Zoologica collection (IZ):** The Iconographia Zoologica[2] (short: IZ) is a 19th century collection of biological illustrations from the Artis Library of the University of Amsterdam. The collection was formed by three collectors: the well-known collector and naturalist Th. G. van Lidth de Jeude, the zoologist R.T. Maitland and the curator of the shell collection at the Amsterdam Zoo, Abraham Oltman, together with the Amsterdam society *Natura Artis Magistra*. In the 21st century, the collection was digitised and labelled with either complete binomial species names (genus and specific epithet) or corresponding genera. The full online collection contains over 26,500 sketches and drawings.

[1] https://www.metamorfoze.nl/kennis-onderzoek/lexicon/preservation-imaging
[2] https://bijzonderecollecties.uva.nl/gedeelde-content/beeldbanken/iconographia.html