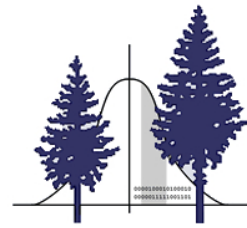




GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN



MASTERARBEIT

ENTWURF EINES FLEXIBLEN
ONLINE-BESTIMMUNGSSCHLÜSSELS FÜR
FORSTINSEKTEN (AUF DER GRUNDLAGE DER
FORMALEN BEGRIFFSANALYSE)

Daniel Jerome Bettin

Erstprüfer
Prof. Dr. Winfried Kurth

Zweitprüfer
Prof. Dr. Wolfgang Rohe

Göttingen, 14. Oktober 2019

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Stand der Technik	2
1.2.1	Webseiten	2
1.2.2	Android Applikationen	7
1.2.3	Tabellarisches Fazit	12
2	Grundlage	13
2.1	Formale Begriffsanalyse	13
2.1.1	Formaler Kontext	13
2.1.2	Formaler Begriff	13
2.1.3	Begriffsverband	14
2.1.4	Liniendiagramm eines Begriffsverbandes	15
2.1.5	Strukturanalyse in der Forstökologie als weiteres Anwendungsbeispiel	17
2.2	Technik	18
2.2.1	Python	18
2.2.2	Django	19
2.2.3	Aufbau eines django Projekts	20
2.2.4	Django Befehle	20
2.2.5	Python concepts	21
2.2.6	JavaScript und D3	22
2.2.7	css3	22
2.2.8	Spracheingabe	23
3	Implementierung	25

3.1	Die Implementierung der Bestimmung	25
3.1.1	Clientseite der Bestimmung	25
3.1.2	Server der Bestimmung	28
3.2	Das Datenbankmodell	30
3.3	Wie django eine HTML-Seite erstellt	32
3.4	Das Python-Modul view.py	34
3.4.1	Die Methode <code>create_context_from_db</code>	34
3.4.2	Die Methode <code>insect</code>	35
3.4.3	Die Methode <code>upload_image</code>	37
3.5	Django-Template-Filter	38
3.6	Infoseite mit d3 und SVG	40
4	Diskussion	43
4.1	Fazit	43
4.1.1	Hirschkäfer-Bestimmung	43
4.2	Ausblick	46
4.2.1	Veröffentlichung	47
4.2.2	Hinweise für Entwickler	47
4.2.3	Vorschläge zur Weiterentwicklung	50
	Verzeichnisse	I

Göttingen 14. Oktober 2019

D. J. Bettin

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Liste der Abkürzungen

APP	Applikation
CSS	Cascading Style Sheets
CSRF	Cross Site Request Forgery
D3	Data-Driven Documents
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JPG	Joint Photographic Experts Group
KML	Keyhole Markup Language
PIL	Python Imaging Library
PNG	Portable Network Graphics
SVG	Scalable Vector Graphics
SQL	Structured Query Language
URL	Uniform Resource Locator

„Insects are the most numerous and successful creatures on Earth“

[McG00]

1 Einleitung

In dem Kapitel 1.1 Motivation wird erörtert, wieso es eines Online-Bestimmungsschlüssel bedarf. In dem darauf folgenden Kapitel 1.2 werden Webseiten und Android Apps vorgestellt, welche ebenfalls zur Bestimmung von Insektenarten gedacht sind.

1.1 Motivation

Ein privates Interesse an der Entomologie war bestimmt ein Grund, weshalb ein Thema, welches mit Insekten zu tun hat, als Masterarbeit ausgewählt wurde. Aber hauptauschlaggebend waren die Themen Insektensterben und die Wichtigkeit der Insektenarten für alle Ökosysteme dieses Planeten. Die momentane Präsenz dieser Themen in den Medien ist durch ihre Wichtigkeit mehr als gerechtfertigt. Insekten bilden als Zersetzer, Bestäuber oder Verwerter meist die Grundlagen jedes Ökosystems. Mit einer riesigen Biomasse dienen sie auch als Futter für andere Arten, doch gerade diese Biomasse ist laut neusten Studien drastisch am sinken, und dies bedroht viele Arten[Sch18]. Wegen des Insektensterbens rufen einige Naturschutzorganisationen zum Zählen der Insektenarten auf[NAB19a]. Für eine genaue Zählung ist es immens wichtig, die Insektenarten unterscheiden zu können.

Die Erwärmung des Klimas hat ebenfalls eine Auswirkung auf die Insekten. Dabei erschließen Arten neue Lebensräume, die nun nicht mehr zu kalt sind, und verdrängen dort als Neozoen dort ansässige Arten. Will man diese ansässigen heimischen Arten schützen, ist eine genaue Unterscheidung zwischen den invasiven Arten und den heimischen Arten notwendig. Die Erwärmung des Klimas wird auch neben den Monokulturen als Ursache für das Waldsterben genannt. Beides begünstigt wohl die vermehrte Anzahl der Borkenkäfer wie sie beispielsweise im Nationalpark Harz gerade herrscht[Gru79]. Wenn von dem Borkenkäfer die Rede ist, geht der nicht entomologisch Erfahrene meist nur von einer Insektenart aus. Fachleute hingegen wissen, dass es sich bei dem Begriff Borkenkäfer oder auch *Scolytinae* um eine Unterfamilie der Rüsselkäfer oder auch *Curculionidae* handelt und es rund 6000 Borkenkäferarten weltweit gibt. Hier könnte ein Online-Bestimmungsschlüssel den entomologisch Unerfahrenen und auch dem Erfahrenen bei der Bestimmung und Unterscheidung der Arten helfen[Wik19a].

Die Insekten bilden die größte Gruppe der Tierarten, doch sind die modernen Bestimmungshilfen wie Webseiten oder Mobilfunktelefon Apps eher selten auf sie angewandt worden. Dazu kommt noch, dass nur ein Bruchteil der tatsächlich zu erwartenden Insektenarten laut dem Buch *Ökologie kompakt* von Wolfgang Nent-

wig, Roland Brandl und Sven Bacher [Nenge] wissenschaftlich beschrieben sind. Zur sicheren und schnellen Bestimmung der noch nicht entdeckten Arten benötigt es auch neue Technologien, die die Bestimmung verbessern und erleichtern, genau dies ist ein Ziel des Online-Bestimmungsschlüssels für Forstinsekten auf der Grundlage der formalen Begriffsanalyse.

1.2 Stand der Technik

In dem Abschnitt Stand der Technik werden die vorhandenen elektronischen Mittel zur Bestimmung von Insekten untersucht. Dabei liegt der Fokus auf den Webseiten, aber auch Android Applikationen werden berücksichtigt, da eine Anforderung an unseren Online-Bestimmungsschlüssel für Forstinsekten ist, dass er eine *responsive design Homepage* werden soll. Dies ist eine Homepage, die für mobile Endgeräte optimiert ist und deren Design an die Benutzung angepasst ist. Da es eine Vielzahl an Webseiten und Applikationen mit dem Thema Insekten gibt, wurden die ausgewählt, welche Vertreter für andere gleicher Art sind, oder besondere Funktionen haben, die die anderen nicht bieten.

1.2.1 Webseiten

Die meisten Webseiten zu dem Thema Insektenbestimmung sind so aufgebaut, dass die Insekten per Name und /oder Bild bestimmt werden. Als Beispiel für eine solche Bestimmung dient die Seite *digital-nature.de*[dig19]. Auf der Seite wird eine Mischung aus lateinischen und deutschen Namen bei der Bestimmung verwendet. Wenn das Insekt bestimmt ist, werden meist Fotos des Insekts zum Vergleich gezeigt. Die Webseite *insektenbox.de* bietet neben der Bestimmung über den deutschen Namen auch die Bestimmung über den wissenschaftlichen Namen[ins]. Sie bietet mit der Berücksichtigung einiger Ordnungen, Larven und Insekten-Gallen schon so etwas wie eine Sucherleichterung für den erfahrenen Nutzer. Unerfahrene Nutzer können die Insektenfibel der Seite nutzen um sich zu informieren. Im Folgenden werden Webseiten näher vorgestellt, die eine Funktion haben, die einer interaktiven Bestimmung gleicht oder nahe kommt. Sie wurden getestet mit dem Browser MOZILLA FIREFOX in der Version 63.0.1 (64-Bit) mit WINDOWS 8 als Betriebssystem.

offene-naturfuehrer.de Die Webseite *offene-naturfuehrer.de*[nat18] bietet neben der Bestimmungshilfe für Insekten eine Vielzahl anderer Bestimmungshilfen an. So ist es beispielsweise möglich, Amphibien, Pflanzen und Pilze zu bestimmen. Die Insektenbestimmung ist auf der Startseite über den Link Insekten und Glie-

dertiere erreichbar. Dort findet eine Unterteilung statt nach *Übersicht Insekten*, *Laufkäfer Deutschlands (vereinfacht)*, *Grashüpfer*, *Heuschrecken*, *Grillen (vereinfacht, Bayern)*, *Hummeln (vereinfacht)*, *Schmetterlinge*, *Spitzkopfizikaden - Fulgoromorpha*, *Pflanzenläuse – Sternorrhyncha*, *Vorrats- und Materialschädlinge* und *Häufige Pflanzengallen*.

Dabei ist der Unterpunkt *Übersicht Insekten* die allgemeine Bestimmung von Insekten. Die anderen Unterpunkte sind vorgesehen für die Bestimmung von Familien (Laufkäfer), Unterfamilien (Grashüpfer) und Gattungen (Hummel). Sie benötigen also alle schon Vorwissen über das zu bestimmende Insekt. Die Seite *Übersicht Insekten* bietet zwei Arten der Bestimmung, die jeweils an das Buch *Bestimmung wirbelloser Tiere – Bildtafeln für zoologische Bestimmungsübungen und Exkursionen* [Bä08] angelehnt sind und dessen Zeichnungen verwenden.

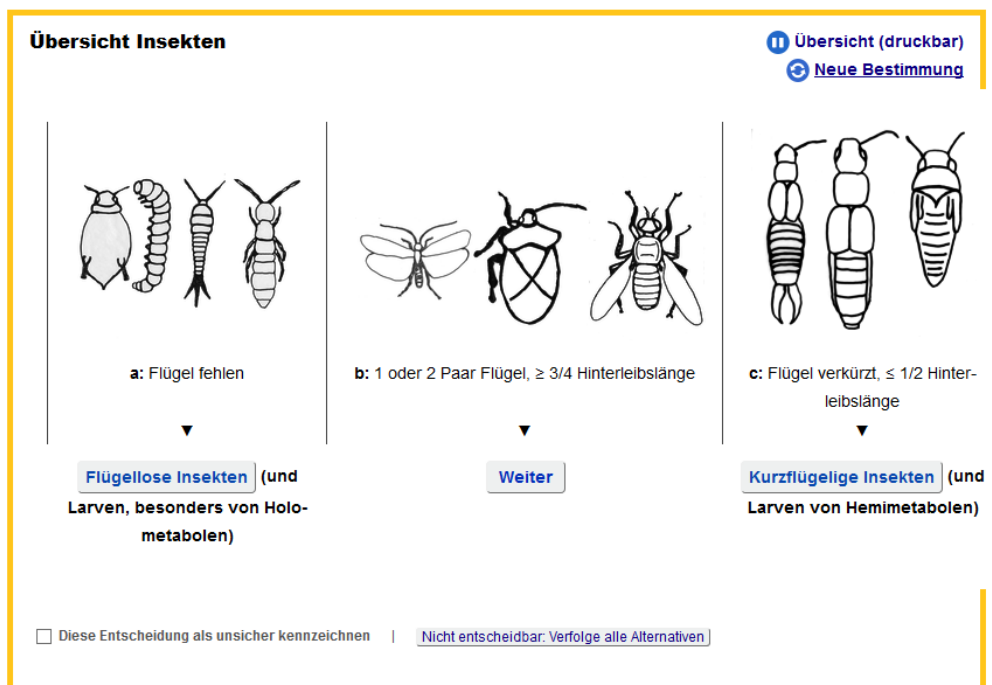


Abbildung 1: Die Startseite der interaktiven Bestimmung

Auf Abbildung 1 ist die Startseite der Bestimmung zu sehen. Hierbei handelt es sich um die selben Abfragen für die interaktive Bestimmung und die nicht interaktive Bestimmung. Beide Bestimmungen unterscheiden sich nur dadurch, dass bei der statischen Bestimmung mit Verlinkungen innerhalb der Seite gearbeitet wird, bei der interaktiven Bestimmung wird hingegen die nächste Abfrage isoliert angeboten, also als eine neue Seite in einem gelb umrandeten Kasten. Inhaltlich haben die Abfragen aber die selben Zeichnungen und den selben Text.

The screenshot shows a web interface titled "Übersicht Insekten". In the top right corner, there are two buttons: "Übersicht (druckbar)" and "Neue Bestimmung". Below the title, a grey box displays the result: "Ergebnis: Käfer Coleoptera". Underneath, a section titled "Bisherige Entscheidungen" lists seven characteristics, each followed by a blue button labeled "überprüfen". The characteristics are: 1. 1 oder 2 Paar Flügel, $\geq 3/4$ Hinterleibslänge; 2. 2 Paar Flügel; 3. Flügel nicht bandförmig, ohne Fransensäume; 4. Vorderflügel stärker sklerotisiert als Hinterflügel, \pm versteift; 5. Vorderflügel einheitlich, \pm geschlossene, steifere Decken bildend; 6. Mundwerkzeuge: Beißkiefer; 7. Flügeldecken starr, gewölbt, nicht geädert. At the bottom of this list, a red arrow points to the result: "Ergebnis: Käfer Coleoptera".

Abbildung 2: Die Ergebnisseite der interaktiven Bestimmung

Am Ende der Abfragen, wenn die Bestimmung abgeschlossen ist, zeigt die Ergebnisseite die bestimmte Kategorie und ihre Eigenschaften an, wie auf Abbildung 2 mit dem Coleoptera als Beispiel zu sehen ist. Dabei kann der Nutzer, falls er unsicher mit einer Eigenschaft des Insekts sein sollte, diese im Nachhinein noch einmal überprüfen. Ein weiterer Vorteil gegenüber dem Bestimmungsbuch ist, dass die Abfragen aus dem Buch, welche keine Zeichnung haben, mit einem Foto eines Insekts ergänzt worden sind. Das Insekt erfüllt dabei die jeweilige Eigenschaft, über die der Nutzer an der Stelle entscheidet.

Die große Stärke des *offene-naturfuehrer.de* ist es, dass dort nicht nur ein Bestimmungsschlüssel zu finden ist, sondern gleich mehrere gebündelt. Dies ist ein erheblicher Vorteil für Nutzer, die nicht nur ein Insekt oder eine Pflanze bestimmen wollen. Auch die Möglichkeit den Bestimmungsschlüssel für die Benutzung an Orten ohne Internetreichbarkeit herunterzuladen ist eine sehr gute Idee. Nachteile sind, dass er oft unübersichtlich wirkt und manche Texte unnötig gequetscht sind. Von der Bedienung der Bestimmung her wäre es von Vorteil, wenn besondere Merkmale auch eingebbar wären.

kerbtier.de Die Webseite *kerbtier.de*[ker19] ist spezialisiert auf Coleoptera. Sie bietet neben der Möglichkeit der Bestimmung durch ein Forum, wo Nutzer des Forums das Insekt gemeinschaftlich bestimmen, auch die Bestimmung nach Aussehen, und das Nachschlagen von Namen ist möglich. *Kerbtier.de* erlaubt dabei die Eingabe von Merkmalen, diese sind aber in der Auswahl beschränkt.



Abbildung 3: Die Auswahl bei Kerbtier.de

Die Eingabe erfolgt, wie in Abbildung 3 zu sehen, über *Radiobuttons*, die zu einer Zeichnung gehören, und *Dropdown Listen*, welche die wählbaren Merkmale bereithalten. Coleoptera, welche die eingegebenen Kriterien erfüllen, werden dann mit Foto angezeigt. Wenn mehr als ein Coleopteron als Ergebnis angezeigt wird, muss der Nutzer dann die Entscheidung treffen, welcher der Coleoptera ist, den er sucht. Dies kann zu Problemen führen, da nicht alle Unterscheidungsmerkmale bei dieser Art der Bestimmung zu erkennen oder zu unterscheiden sind. Unter dem Foto des bestimmten Käfers befinden sich noch Zusatzinformationen, wie sein geographisches und zeitliches Vorkommen, ob die Art bedroht ist, der Katalogstatus in Deutschland, ein Link für weitere Informationen und Berichte auf *kerbtier.de* und eine Erklärung der Namensherkunft. Die Seite *kerbtier.de* bietet für den entomologischen Neuling mehrere interaktive Erklärungen. Eine davon ist auf Abbildung 4 zu sehen. Sie enthält links die Morphologie eines Coleopterons und rechts die deutschen und wissenschaftlichen Namen der Körperabschnitte. Das Interaktive dabei ist der *hover*-Effekt des Textes. Dieser wird ausgelöst durch das Darüberfahren mit dem Mauszeiger. In der Abbildung ist der Mauszeiger über dem Text *Flügeldecke (Elytre)*, dieser ist dabei blau unterlegt, ebenso wie die Flügeldecken auf der Zeichnung des Käfers blau unterlegt sind.

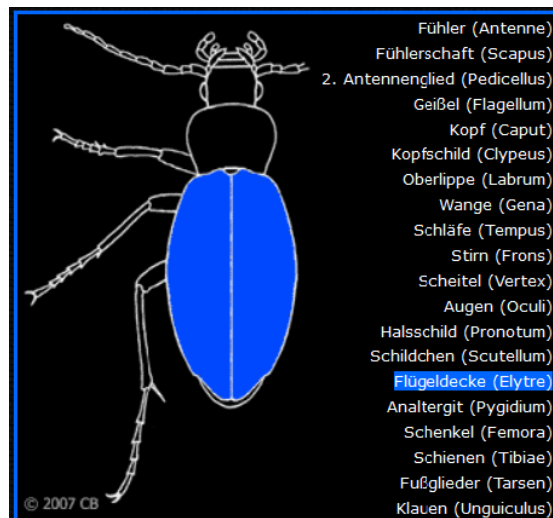


Abbildung 4: Interaktive Erklärung bei Kerbtier.de

coletonet.de und lompe.de Auf der Webseite *lompe.de* wird der Nutzer nach Aufrufen des Links *Bestimmungstabellen der Käfer Europas* weiter geleitet auf die Webseite *coletonet.de*[Lom19]. Beide Webseiten sind von Arved Lompe verfasst und beinhalten nur Insekten der Ordnung Coleoptera. Da die Bestimmungstabellen auf *coletonet.de* zu finden sind, wird diese Webseite im folgenden näher untersucht. Die Bestimmung der Webseite funktioniert mit binären Abfragen. Bei einer Abfrage gibt es zwei Auswahlmöglichkeiten. Die eine ist als # gefolgt von einer Zahl, die bei jeder solchen binären Abfrage hochgezählt wird, dargestellt, die andere befindet sich jeweils darunter und wird mit zwei - - markiert. Die Bestimmung findet über Text, Fotos und Zeichnungen statt, zu sehen in Abbildung 5. Wenn eine der beiden Abfragen zutrifft, wird der jeweilige Link geklickt, in der Abbildung sind es die blauen Punkte gefolgt von den Zahlen (...3 und ...7). Es besteht aber auch die Möglichkeit, dass der Link einer Unterordnung, Familie oder ähnlichem entspricht. Dabei kommt es auch vor, dass der Familienname keine Verlinkung bietet. Dies ist zum Beispiel bei den Urodontinae der Fall, hier wird dann der Name schwarz geschrieben um sich von den Verlinkungen abzuheben. Dies ist dann das Ende der Bestimmung. Bei den Familiennamen mit Verlinkung befindet sich hinter dem Link erst eine Beschreibung der Familie und anschließend eine weitere Bestimmung innerhalb der Familie. Am Ende der Bestimmung befinden sich Informationen zu dem bestimmten Objekt, weiterführende Links und bei manchen Arten eine *kml*-Datei, diese enthält Sichtungsorte der jeweiligen Art.



Abbildung 5: Interaktive Bestimmung bei Coleonet.de

coletonet.de bietet eine sehr große Auswahl an Coleoptera, aber leider auch nur Coleoptera. Die Bestimmung funktioniert nicht immer, da manche Links auf Seiten verweisen, die entweder nicht mehr oder noch nicht da sind (als Beispiel Melolontha). Verwirrend ist zudem, dass für die Verlinkungen zwei Farben gewählt wurden und der Internetnutzer so annehmen könnte, dass die andersfarbigen Verlinkungen schon besucht worden sind, was aber nicht der Fall ist. Sehr gut sind die Zeichnungen und Fotos der Bestimmung und deren *hover*-Effekt, welcher die Bilder vergrößert. Dieser funktioniert aber nur auf den Unterseiten. Auf der Startseite

der Bestimmung kann das Bild mit einem Doppelklick auf dessen Referenz im Text geöffnet werden, diese ist aber meist kleiner als das vorher angezeigte Bild.

amentsoc.org Die Webseite *amentsoc.org*[ame19] stammt von der *Amateur Entomologists' Society* und ist englischsprachig. Sie bietet viele Informationen über Insekten, deren Haltung, deren Schutz und wie der Besucher der Webseite dabei helfen kann. In Abbildung 6 ist der Start der Bestimmung zu sehen. Die Bestimmung ist rein textuell gehalten, aber sie bietet bei Unsicherheiten die Möglichkeit, sich über einen Link weitere Informationen einzuholen (der Link *vestigial* in der Abbildung). Diese zeigen dann meist ein Foto und einen Erklärungstext. Die Bestimmung ist mit binären Fragen aufgebaut, wobei am Ende jeder Auswahlmöglichkeit eine Verlinkung zu der nächsten binären Abfrage steht. Dies geschieht solange, bis ein Klassenname oder Ordnungsname anstatt einer *Go to* - Verlinkung da ist. Der Name ist dabei wieder eine Verlinkung, die Informationen über die Klasse oder Ordnung bereit hält.

Key to insect orders

Start at Question 1 and follow the links until you've identified your insect.


1. Insect has wings? [Go to 2](#)
Insect wingless or with poorly developed ([vestigial](#) ) wings. [Go to 29](#)
2. One pair of wings. [Go to 3](#)
Two pairs of wings. [Go to 7](#)

Abbildung 6: Interaktive Bestimmung bei amentsoc

Durch die Einfachheit der textuellen Bestimmung und deren Hilfe ist die Bestimmung sehr übersichtlich und intuitiv nutzbar. Leider reicht die Bestimmung nur bis zu den Klassen und Ordnungen.

1.2.2 Android Applikationen

Die Android Applikationen wurden mit einem *Acer* Tablet und der Android Version 5.1 getestet. Die Auswahl der Applikationen fand anhand der Bewertungen im *Google Playstore* und nach Empfehlungen von Artikeln auf *giga.de*[Bas19], *chip.de*[Pet19] und *hna.de*[dpa19] statt. Sie sind alle kostenlos im *Google Playstore* downloadbar.

NABU Die *NABU-Insektenwelt*[NAB19b] Android Applikation bietet laut ihrer Informationsseite 122 Insektenarten. Diese stammen von der Webseite *naturgucker.de* und sind die am häufigsten dort gemeldeten Insektenarten. Auf der

voreingestellten Hauptseite kann der Nutzer zwischen *Käfer und Wanzen*, *Schmetterlinge*, *Libellen* und *sonstigen Insekten* wählen. Diese Ansicht kann der Nutzer aber auch umstellen, so dass alle Insektenarten als alphabetisch geordnete Liste mit Foto zu sehen sind, oder in eine dritte Ansicht. Diese zeigt eine Liste mit den Ordnungen auf deutsch und dem wissenschaftlichen Namen in Klammern. Dort kann dann ein Dropdown-Menü geöffnet werden, welches die Arten zu der Ordnung enthält.

Die Abbildung 7 zeigt die Übersichtseite unter der Auswahl *sonstige Insekten*. Diese bietet auf der linken Seite eine Bilderliste der Insekten. Diese kann der Nutzer mit den Pfeilbuttons (unten rechts) durchlaufen, so dass immer das Nachbarinsekt als nächstes in der Auswahl ist. In diesem Fall ist es die Ackerhummel (*Bombus pascuorum*), die angezeigt wird. Es ist aber auch möglich, mit dem Button, der zwei übereinander gelegte Bilder zeigt, zwei oder mehr Insektenarten auszuwählen und zu vergleichen. Wenn dabei nur zwei Arten ausgewählt sind, werden deren Eigenschaften nebeneinander angezeigt und der Nutzer kann so leicht die Unterschiede bestimmen. Dies kann visuell durch die Fotos geschehen oder durch die textuellen Informationen, die aufgeteilt sind in *Taxonomie*, *Merkmale und Doppelgänger*, *Größe(Länge)*, *Größe(Flügelspannweite)*, *Vorkommen Zeitraum in Monaten*, *Vorkommen in Deutschland*, *Lebensraum(Habitat)*, *Lebensweise* und einen Punkt, der nicht zur Bestimmung dient, aber dem Nutzer Hilfe anbietet, falls er diesem Insekt im eigenen Garten helfen möchte.

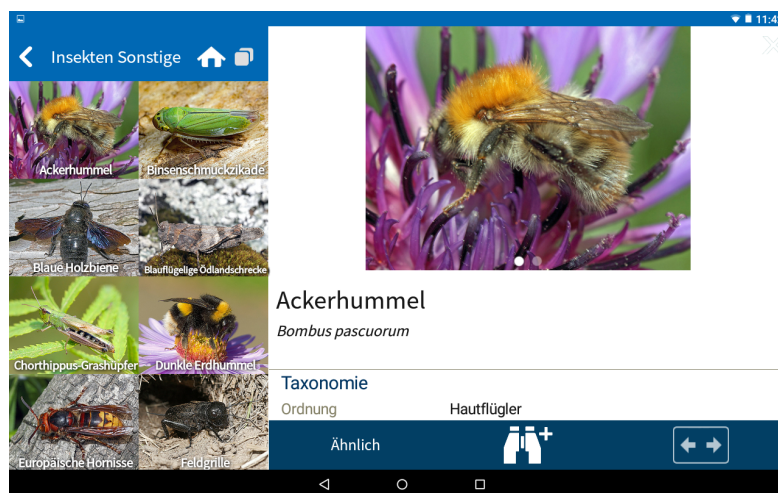


Abbildung 7: Übersicht der NABU Insekten

Die Applikation bietet außerdem eine Fotobestimmung, mit der eine gefundene Insektenart per Foto bestimmt werden kann. Dabei kann ein Foto des Insekts mit der Kamera erstellt werden oder eines aus der Galerie des mobilen Endgerätes geladen werden. Dies hat aber beides mit dem Tablet und mit einem leistungsstärkeren *Houawei Honor 7* mit *Android Version 6.0* nicht funktioniert. Es führte nur dazu, dass die Applikation ohne Ergebnis beendet wurde.

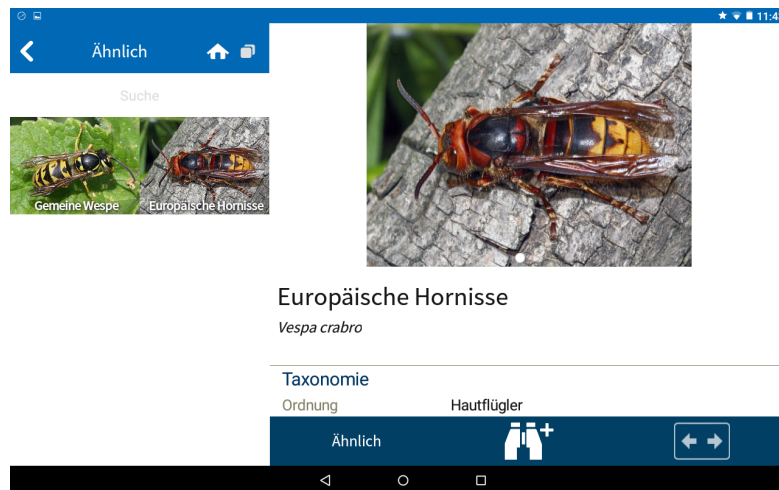


Abbildung 8: Vergleich von ähnlichen Insekten

In Abbildung 8 ist die Europäische Hornisse (*Vespa crabro*) zu sehen. Diese wurde ausgewählt als Beispiel für die Ähnlich-Funktion der Applikation. Dabei werden dem Nutzer Insekten gezeigt, welche ähnliche Merkmale oder ein ähnliches Aussehen haben, wie das Insekt, welches er ausgewählt hat. Nachteil dabei ist die beschränkte Anzahl der Insektenarten in der Applikation. Ein Beispiel dafür ist die Hornissenschwebfliege (*Volucella zonaria*). Diese ist nicht in der Applikation und wird so auch nicht als ähnliches Insekt vorgeschlagen, wenn die Hornisse (*Vespa crabro*) bestimmt werden soll. Da sie diese aber mimikriert, kann dies zu Verwechslungen führen. Auf sie wird zwar im Text *Merkmale und Doppelgänger* der Hornisse hingewiesen, besser wäre aber Merkmale zu nennen, welche die Arten unterscheiden, wie zum Beispiel “hat keinen Stachel“ oder ähnliches.

Map of Life Bei der Applikation *Map of Life*[Map18] wird zuerst das Gebiet, in welchem der Nutzer eine Bestimmung durchführen möchte, eingegeben. Dies kann über die Standortbestimmung des Mobilgerätes erfolgen oder aber per manueller Eingabe. Anschließend wird, wie in Abbildung 9 links zu sehen ist, eine Liste mit Tierklassen und Pflanzen angezeigt. Dies bietet den Vorteil, dass der Nutzer nur Tiere in der Bestimmungsauswahl hat, welche dort auch häufig vorkommen. Nachteil dabei ist aber, gerade im Zeitalter der Neozoen, dass diese für die Gegend fremden Arten dort nicht in der Bestimmung sind. Nach der Auswahl *Bombus* (Hummel, Bumblebee in der Abbildung 9) aus der Liste links in der Abbildung wird auf der nächsten Seite eine Liste von Hummeln angezeigt, zu sehen in der Mitte in Abbildung 9. Diese Liste kann nach Artnamen gefiltert oder manuell durchgescrollt werden. Im Beispiel in der Abbildung (rechts) wurde die *Bombus pomorum* ausgewählt. Die darauf folgende Seite beinhaltet Fotos der Art, eine Karte mit deren geographischen Vorkommen und einen kurzen Beschreibungstext,

der von WIKIPEDIA stammt. Der WIKIPEDIA-Artikel der Art ist auch über einen Link zu erreichen und bietet weitere Informationen.

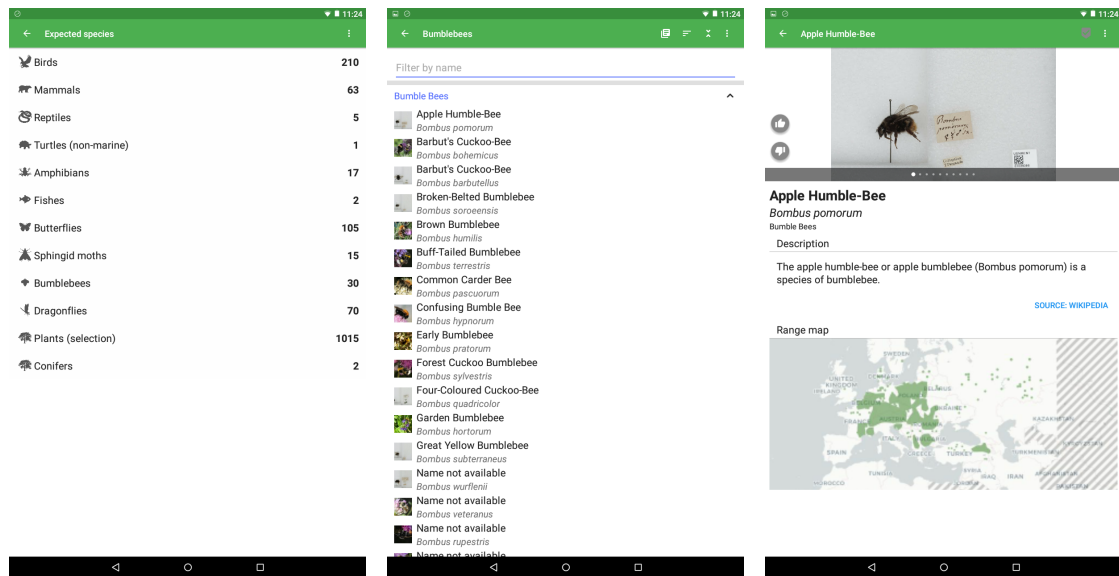


Abbildung 9: Vergleich von ähnlichen Insekten

Es wäre für jeden Biologen natürlich optimal, wenn er eine Applikation hätte, mit der er alles bestimmen kann und die außerdem noch Sichtungen von Tieren und Pflanzen von jedem Nutzer zur Verfügung stellt. Von den getesteten Applikationen ist diese die einzige, die eine geographische Eingrenzung anbietet, allerdings sind die Insekten leider nur mit ein paar Ordnungen vertreten. Es wäre außerdem von Vorteil, eine umfassendere geographische Eingrenzung angeben zu können, wie Europa, Asien oder die ganze Welt.

Nature Free - Europa Die Applikation *Nature Free*^[sar18] ist zwar kostenlos herunterladbar, will der Nutzer aber alle Funktionen haben, muss er sie per sogenannten *In-App-Käufen* frei kaufen. Auf der Startseite befindet sich das Auswahlmenü, welches auf dem ersten Blick einer Fotocollage gleicht. Dort kann der Nutzer zwischen den Themenüberschriften: *Säugetiere*, *Reptilien Amphibien*, *Vögel*, *Pflanzen Bäume*, *Insekten Wirbellose-Landtiere*, *Pilze Farnpflanzen Flechten/Moose* und *Fische Wirbellose-Wassertiere* wählen. Nachdem der Nutzer sich für einen der Menüpunkte entschieden hat, bekommt er eine nach deutschen Namen alphabetisch sortierte Liste der jeweiligen Kategorie. Auf diese Liste kann anschließend noch ein Filter angewandt werden, zu sehen in Abbildung 10. Dabei kann nach einem selbst eingegebenen Text gesucht werden oder nach Merkmalen für eine Gruppe, im abgebildeten Beispiel ein Käfer, nach der Größe, dem Aussehen, den Farben, der Zeichnung und dem Lebensraum. Die Auswahlkriterien sind dabei nicht frei eingebbar, sondern müssen aus einer *Pop-up-Seite*, die erscheint, wenn auf das Kriterium geklickt wurde, ausgewählt werden. Nach Anwenden des Fil-

ters erhält der Nutzer eine neue Liste, die nur noch die Arten enthält, welche den Suchkriterien entsprechen.



Abbildung 10: Auswahl von Merkmalen von Insekten

Leider sind einige bekannte Arten wie beispielweise der *Lucanus cervus* nicht mit in der Bestimmung. Positiv sind das Glossar und ein Quiz, in dem der Nutzer die Bestimmung anhand von Fotos üben kann.

1.2.3 Tabellarisches Fazit

Webseiten

	offene -naturfuehrer	kerbtier	coleonet	amentsoc
Arten	Amphibien, Pflanzen, Pilze, Insekten	Coleoptera	Coleoptera	Insekten
Umfang	30000-50000	ca. 2000	ca 10000	ca 50
interaktive Bestimmung	ja	ja	ja	ja
Eingabe von Merkmalen	nein	ja	nein	nein
Bestimmungsschlüssel als Grundlage	ja	ja	nein	ja
Sprache	deutsch	deutsch	deutsch	englisch
Erklärungshilfen zur Taxonomie	nein	ja	nein	ja

Tabelle 1: Auflistung der Unterschiede bei Webseiten

Android Apps

	NABU	Map of Life	Nature Free
Arten	Insekten	Pflanzen- und Tierarten	Pflanzen- und Tierarten
Umfang	122	30000	1800
interaktive Bestimmung	nein	nein	nein
Eingabe von Merkmalen	nein	nein	ja
Bestimmungsschlüssel als Grundlage	nein	nein	nein
Sprache	deutsch	deutsch	deutsch
Erklärungshilfen zur Taxonomie	nein	nein	(ja)
geographische Bestimmung	nein	ja	nein
Kostenlos	ja	ja	nein

Tabelle 2: Auflistung der Unterschiede bei Apps

2 Grundlage

2.1 Formale Begriffsanalyse

Bei der formalen Begriffsanalyse handelt es sich um ein Gebiet der angewandten Mathematik. Bei ihr werden aus binären Relationen wissensbasierte Strukturen aufgebaut. Das Standardwerk dabei ist „Formale Begriffsanalyse Mathematische Grundlagen“ von Bernhard Ganter und Rudolf Wille [Gan96]. In diesem schreiben die Autoren, dass die elementaren Grundbegriffe die des formalen Kontextes (Kapitel 2.1.1) und des formalen Begriffs (Kapitel 2.1.2) sind.

2.1.1 Formaler Kontext

Ein formaler Kontext K (Abbildung 11) besteht aus zwei Mengen G und M . Die Elemente von G werden Gegenstände genannt und die von M Merkmale. Außerdem gibt es eine Relation I zwischen G und M , die Inzidenzrelation genannt wird. Wenn nun ein Gegenstand g mit einem Merkmal m in Relation steht, wird dies $(g, m) \in I$ oder kürzer gIm geschrieben und gelesen als „der Gegenstand g hat das Merkmal m “.

$$K := (G, M, I) : \text{formaler Kontext} \quad (1)$$

$$G : \text{Menge der Gegenstände} \quad (2)$$

$$M : \text{Menge der Merkmale} \quad (3)$$

$$I \subseteq G \times M : \text{Inzidenzrelation} \quad (4)$$

Abbildung 11: Definition des formalen Kontext [Gan96]

Ein formaler Kontext lässt sich auch, wie in Tabelle 3 zu sehen ist, als Kreuztabelle darstellen. Dabei sind die Gegenstände (g) in den Zeilen, und die Merkmale (m) in den Spalten eingetragen. Ein Kreuz bedeutet dabei, dass ein Gegenstand ein Merkmal erfüllt.

2.1.2 Formaler Begriff

Ein formaler Begriff des Kontextes (G, M, I) ist ein Paar aus dem Umfang A und dem Inhalt B . Für eine Menge $A \subseteq G$ von Gegenständen und eine Menge $B \subseteq M$

		m
		\vdots
g	\dots	x

Tabelle 3: Kreuztabelle aus „formale Begriffsanalyse Mathematische Grundlagen“ [Gan96]

von Merkmalen gilt folgende Definition:

$$\begin{aligned}
A' &:= \{m \in M \mid gIm \ \forall g \in A\} \\
B' &:= \{g \in G \mid gIm \ \forall m \in B\} \\
(A, B) &\text{ heißt formaler Begriff des Kontextes } (G, M, I) \\
&\text{ mit Umfang } A \text{ und Inhalt } B, \text{ wenn} \\
&A \subseteq G, B \subseteq M, \\
&A' = B \text{ und } B' = A
\end{aligned} \tag{5}$$

Abbildung 12: Definition des formalen Begriffs [Gan96]

Bei der Erstellung eines formalen Begriffs wird zeilen- und spaltenweise überprüft, welche Gegenstände welche Merkmale haben und umgekehrt, welche Merkmale gehören zu welchen Gegenständen. Dabei ist noch darauf zu achten, dass die Umfänge der formalen Begriffe maximal groß sind. Dazu kommen noch das Infimum, ein formaler Begriff, dessen Gegenstände alle Merkmale erfüllen, und das Supremum, ein ebenfalls formaler Begriff mit allen Gegenständen und den Merkmalen, die diese erfüllen.

2.1.3 Begriffsverband

In diesem Abschnitt wird der mathematische Verband vorgestellt und darauf aufbauend der Begriffsverband der formalen Begriffsanalyse.

Verband Definition Eine partiellgeordnete Menge $V := (V, \leq)$ ist ein Verband, wenn zu je zwei Elementen $x, y \in V$ stets das Supremum $x \vee y$ und das Infimum $x \wedge y$ existieren. Ein Supremum ist dabei die kleinste obere Schranke und ein Infimum die größte untere Schranke.

Begriffsverband Definition Die Menge aller Begriffe von $\mathfrak{B}(G, M, I)$ zusammen mit der unten definierten partiellen Ordnung \leq ist der Begriffsverband [Stu05].

Die Abbildung 13 zeigt die allgemeine Definition einer partiellen Ordnung.
 Sei A eine Menge und R eine Relation mit $R \subseteq A \times A$, dann wird verlangt:

$$\forall a \in A : (a, a) \in R \quad \text{Reflexivität (6)}$$

$$\forall a, b \in A : (a, b) \in R \wedge (b, a) \in R \Rightarrow a = b \quad \text{Antisymmetrie (7)}$$

$$\forall a, b, c \in A : (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R \quad \text{Transitivität (8)}$$

Abbildung 13: Definition der partiellen Ordnung[Uni]

Sind (A_1, B_1) und (A_2, B_2) Begriffe eines Kontextes, dann heißt (A_1, B_1) Unterbegriff von (A_2, B_2) , wenn $A_1 \subseteq A_2$ zutrifft (gleichbedeutend dazu ist $B_2 \subseteq B_1$). Dies bedeutet, dass ein formaler Begriff ein Unterbegriff eines anderen formalen Begriffes ist, wenn sein Umfang ganz im Umfang von diesem enthalten ist. (A_2, B_2) ist dann ein Oberbegriff von (A_1, B_1) . Wir schreiben $(A_1, B_1) \leq (A_2, B_2)$. Die Relation \leq wird die hierarchische Ordnung der Begriffe genannt[Gan96] und erfüllt die Eigenschaften einer partiellen Ordnung.

2.1.4 Liniendiagramm eines Begriffsverbandes

Um ein Liniendiagramm für einen Begriffsverband darzustellen wurde hierfür ein einfaches Beispiel mit drei Coleopteras gewählt. Die Coleoptera sind dabei die Gegenstände und die Merkmale werden einfachheitshalber mit Buchstaben von A-F abgekürzt, zu sehen in der Abbildung 14. Dabei befindet sich rechts neben der Kontexttabelle die Erklärung, welcher Buchstabe für welches Merkmal steht.

	A	B	C	D	E	F
<i>Hylotrupes bajulus</i>	X		X			X
<i>Melolontha hippocastani</i>	X	X			X	
<i>Lucanus cervus</i>		X	X	X		

- A Halsschild und (oder) Flügeldecken nicht metallisch
- B Fühler mit einseitig beblätterter Keule
- C Fühler nicht deutlich gekniet oder ohne Keule
- D Mandibel geweihartig
- E Fühler deutlich gekniet Basisglied kurz
- F Stirn schräg abfallend

Abbildung 14: Beispiel anhand drei Coleopteras

Um das Beispiel übersichtlich zu gestalten wurden bestimmte Merkmale ausgewählt. Diese sind: ein Merkmal pro Art, welches nur die Art besitzt, und dazu je ein Merkmal, was sich zwei Arten teilen, die dritte Art aber nicht hat. Daraus entstehen acht formale Begriffe, welche in Abbildung 15 zu sehen sind. Dabei ist das Infimum in Zeile 9 zu sehen. Das Infimum hat als Umfang dabei die Menge

an Gegenständen die alle Merkmale erfüllen, in diesem Fall ist es die leere Gegenstandsmenge. In Zeile 16 ist das Supremum zu sehen. Es hat als Inhalt die Menge an Merkmalen, die alle Gegenstände gemeinsam haben, hierbei handelt es sich im Beispiel um die leere Merkmalsmenge.

$$()('A', 'B', 'C', 'D', 'E', 'F') \quad (9)$$

$$('Hylotrupes bajulus')('A', 'C', 'F') \quad (10)$$

$$('Melolontha hippocastani')('A', 'B', 'E') \quad (11)$$

$$('Lucanus cervus')('B', 'C', 'D') \quad (12)$$

$$('Hylotrupes bajulus', 'Melolontha hippocastani')('A') \quad (13)$$

$$('Hylotrupes bajulus', 'Lucanus cervus')('C') \quad (14)$$

$$('Melolontha hippocastani', 'Lucanus cervus')('B') \quad (15)$$

$$('Hylotrupes bajulus', 'Melolontha hippocastani', 'Lucanus cervus')() \quad (16)$$

Abbildung 15: Begriffe, aus der Tabelle 3 gebildet

Bei den restlichen formalen Begriffen (Zeile 10-15) handelt es sich um die Merkmals- und Gegenstandskombinationen, welche neben dem Infimum und Supremum aus dem formalen Kontext gebildet wurden. Dabei wird, wie schon in Kapitel 2.1.2 beschrieben, zeilen- und spaltenweise überprüft, welche Gegenstände welche Merkmale haben, und umgekehrt, welche Merkmale gehören zu welchen Gegenständen. Das aus dem formalen Begriffen erstellte Liniendiagramm ist in Abbildung 46 zu sehen. Die Knoten des Liniendiagramms stehen für die Begriffe, und anhand der Kanten lässt sich die Ordnung des Verbandes ablesen. Der oberste Knoten in der Abbildung steht für das Supremum und der unterste für das Infimum. Der Knoten unten links, in der Abbildung mit der Beschriftung *Melolontha hippocastani* E, ist der Knoten, der den *Melolontha hippocastani* bestimmt. Dies ist so, weil er der einzige formale Begriff ist, neben dem Infimum, der Merkmal E (Fühler deutlich gekniet Basisglied kurz) enthält, und da *Melolontha hippocastani* die einzige Insektenart in der Bestimmung ist, die das Merkmal E erfüllt. Es handelt sich bei dem Knoten also um den formalen Begriff (*'Melolontha hippocastani'*) (*'A', 'B', 'E'*) aus der Abbildung 15 in Zeile 11. Der Knoten darüber, mit A beschriftet, repräsentiert den formalen Begriff (*'Hylotrupes bajulus', 'Melolontha hippocastani'*) (*'A'*) aus der Abbildung 15 in Zeile 13.

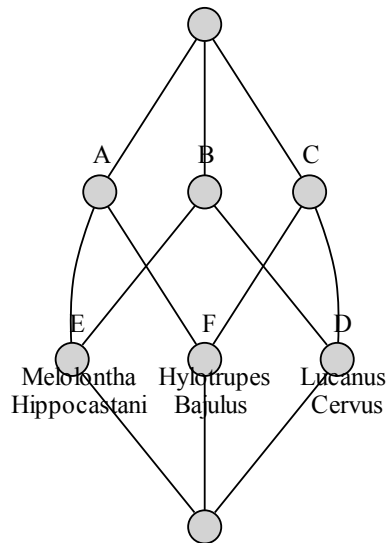


Abbildung 16: Beispiel anhand drei Coleoptera

Bei einer Bestimmung ohne Wissen über die Insektenart würde der Nutzer beim Supremum starten und sich Richtung Infimum bewegen. Dabei würde er die binären Abfragen bei jedem Knoten beantworten, wie zum Beispiel bei Knoten A sind *Halsschild und (oder) Flügeldecken nicht metallisch*. Wenn dies zutrifft, befindet sich der Nutzer bei dem Knoten, der mit A beschriftet ist. Trifft aber das Merkmale von Knoten B zu, dann ist er bei Knoten B. Das selbe gilt für den Knoten C. Wenn der Nutzer sich nun zum Beispiel bei Knoten A befindet, bekommt er die beiden Nachbarn in Richtung Infimum als nächste Frage gestellt. In dem Liniendiagramm aus Abbildung 15 wären es also der Knoten E mit dem Merkmal *Fühler deutlich gekniet Basisglied kurz* und der Knoten F mit dem Merkmal *Stirn schräg abfallend*. Falls das Merkmal von Knoten E zutrifft, befindet sich der Nutzer dann bei dem Knoten E im Liniendiagramm. Das entsprechende gilt für den Knoten F. Mit Erreichen eines Nachbarn vom Infimum ist in diesem Beispiel die Bestimmung beendet, da sich im formalen Begriff des jeweiligen Knoten nur ein Gegenstand befindet, als Beispiel Knoten E mit dem formalen Begriff (*'Melolontha hippocastani '*) (*'A', 'B', 'E'*). Somit ist die gesuchte Insektenart in diesem Fall der *Melolontha hippocastani*.

2.1.5 Strukturanalyse in der Forstökologie als weiteres Anwendungsbeispiel

In der Dissertation von Hubert Röder wird mit der formalen Begriffsanalyse eine Strukturanalyse der Standort- und Vegetationsdaten durchgeführt[Rö97]. Das Ziel

seiner Strukturanalyse ist dabei eine Struktur in den Daten zu finden. Er untersucht dabei unter anderem die Struktur von Datensätzen über bodenphysikalische Eigenschaften von Waldböden und Daten zu einem Auerhuhnhabitat. Er visualisiert auch eine Literaturanalyse und stellt dabei die Beziehungen der Autoren untereinander und zu Waldbewirtschaftungsformen mittels der formalen Begriffsanalyse dar. In der Arbeit zeigt er, dass die Explorative Datenanalyse, die zum Zeitpunkt des Verfassens der Arbeit hauptsächlich auf numerischen Methoden basiert, besser mit der formalen Begriffsanalyse funktioniert, da bei ihr nicht, wie bei vielen statistischen Verfahren, Kennzahlen, die für Datenreduktion stehen, interpretiert werden. Diese Kennzahlen sind laut Röder für den Betrachter schwerer zu interpretieren als die Verwendung von geeigneten Begriffen und Darstellungen, wie sie bei der formalen Begriffsanalyse gebräuchlich sind. Röder verknüpft die formale Begriffsanalyse auch mit einem geographischen Informationssystem für eine bessere Visualisierung seiner Ergebnisse. Voraussetzung dafür ist, dass die Aufnahmepunkte der Daten Koordinaten besitzen. Abschließend sagt Röder, dass die formale Begriffsanalyse der kognitiven Wissensaufnahme des Menschen sehr entgegenkommend ist und dass das Einbringen von vorhandenem Wissen zu jedem Zeitpunkt möglich ist.

2.2 Technik

In diesem Abschnitt werden die verwendeten Programmiersprachen und Frameworks vorgestellt. Es wird dabei auch begründet, warum eine Technologie ausgewählt wurde.

2.2.1 Python

PYTHON wurde ausgewählt, da es sich um eine einfache Programmiersprache handelt. Wenig Schlüsselwörter und eine gut lesbare Syntax waren die Hauptabsichten von Guido van Rossum beim Entwickeln von PYTHON[Kle17]. Die meisten Linux-Distributionen haben PYTHON bereits vorinstalliert, und die Installation bei Windows- und Mac-Systemen ist über ausführbare Installationsdateien sehr einfach möglich. Zum Erlernen oder zum Schreiben kurzer Skripte bietet PYTHON die Möglichkeit einer interaktiven PYTHON-Shell. Es gibt aber auch eine Vielzahl an Entwicklungsumgebungen für das professionelle Entwickeln von PYTHON. Wegen der geringen Einstiegshürde bietet sich PYTHON als Sprache für erfahrene und unerfahrene Programmierer gleichermaßen an.

Es gibt zwei PYTHON-Versionen, eine ältere, die Version PYTHON 2, und eine neuere, die Version PYTHON 3. Für den Online-Bestimmungsschlüssel wurde die

Version PYTHON 3 ausgewählt, da diese einige Verbesserungen der Sprache bietet und die Zukunft von PYTHON darstellt. Die Versionen sind nicht abwärtskompatibel: ein Skript, welches in PYTHON 2 geschrieben wurde, kann nicht mit einem PYTHON 3 Interpreter interpretiert werden, wenn es dabei gegen die Neuerungen der Sprache verstößt. Ein weiterer Grund für die Wahl von PYTHON ist, dass diese Sprache auf dem TIOBE-Index auf Platz 3 (Stand März 2019) steht[TIO19]. Dieser Index sortiert Programmiersprachen nach ihrer Popularität, die abgeleitet wird nach der Häufigkeit des Vorkommens des Sprachennamens, zum Beispiel auf Internetseiten, Suchanfragen und ähnlichem. Gerade in der Wissenschaft ist PYTHON sehr beliebt, da es für viele Anwendungsgebiete schon vorgefertigte Frameworks bereitstellt. Außerdem ist für Anwender aus informatikfernen Studienrichtungen die Einstiegshürde kleiner als bei anderen Sprachen wie zum Beispiel JAVA. Für den Online-Bestimmungsschlüssel ist dies ein klarer Vorteil, da kleine Änderungen am Programm nicht zwangsläufig von einem Informatiker oder Programmierer getan werden müssen. Auch die Unterstützung vieler Online-Foren und Tutorials, sowie gute Dokumentationen und Hilfeseiten haben dazu beigetragen, dass PYTHON als Programmiersprache ausgewählt wurde.

2.2.2 Django

DJANGO ist ein auf PYTHON basierendes Webframework. Ein Webframework erleichtert dem Programmierer im allgemeinen die Kommunikation mit dem Webserver, die Generierung von HTML-Seiten nach Templatevorlagen und den Zugriff auf eine Datenbank. DJANGO ist nach dem Model-View-Prinzip aufgebaut. Das Model ist dabei das Datenmodell, was ähnlich einer Objektklasse angelegt werden kann. Es bietet sozusagen die Schnittstelle zur Datenbank. Der Programmierer muss dabei keine SQL-Abfragen benutzen, sondern kann per PYTHON-Code mit der Datenbank kommunizieren. Der View ist die Ausgabe, die der Benutzer sieht und mit der er interagieren kann. Er ist also die Schnittstelle zum Benutzer. Bei DJANGO wird zwischen Projekten und Applikationen unterschieden. Eine Webseite ist meist ein Projekt und einzelne Funktionen der Webseite werden in Applikationen aufgeteilt. Applikationen in DJANGO entsprechen dem *Loose Coupling* Prinzip, das bedeutet, dass möglichst viel unabhängig voneinander entwickelt wird. Das hat den Vorteil, dass die Applikationen dadurch gut in anderen Projekten wieder benutzbar sind. Das Prinzip unterstützt dadurch auch die Wartbarkeit eines Projekts, da nur die Applikation verändert und gegebenenfalls noch die Schnittstelle angepasst werden muss.[Ernge]

2.2.3 Aufbau eines django Projekts

Mit dem Befehl `django-admin.py startproject`, gefolgt von dem Namen, den das Projekt haben soll, wird ein DJANGO-Projekt erstellt. In Abbildung 17 sind der Aufruf mit der danach erstellten Ordnerstruktur zu sehen. Der übergeordnete Ordner enthält die Dateien, die wichtig sind für die Verwaltung des Projekts. In dem untergeordneten Ordner befinden sich die Dateien, die das eigentliche Projekt darstellen.

```
1 | django-admin.py startproject projectName
```

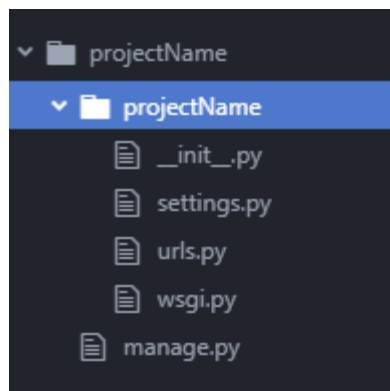


Abbildung 17: Befehl zum Erstellen der Projektstruktur

Mit der Datei `manage.py` können projektspezifische Aufgaben ausgeführt werden, wie das Starten des Entwicklungsservers, Datenbanken erstellt und geändert werden und Applikationen dem Projekt hinzugefügt werden. Die Datei `__init__.py` ist meist leer und dient nur dazu, dass das Projekt als PYTHON-Modul eingebunden werden kann. Die Einstellungen für das Projekt werden in der Datei `settings.py` vorgenommen. Unter welcher URL die jeweilige Seite und Subseite der Homepage erreichbar ist, wird in der Datei `urls.py` geregelt. Die Datei `wsgi.py` dient der Kommunikation mit Webservern.

2.2.4 Django Befehle

Der wohl wichtigste Befehl von DJANGO ist der „runserver“ Befehl. Dieser Befehl startet den Server auf dem Localhost auf dem Port 8000. In Abbildung 18 befindet sich in Zeile 1 die vollständige Konsoleneingabe des Befehls. Dabei ruft „python“ den Python Interpreter auf mit dem Argument „manage.py“, was wiederum das Argument „runserver“ bekommt. Wie der Name Localhost schon andeutet, handelt es sich dabei um einen lokalen Server, der vor allem für die Entwicklung einfacher ist als das Aufsetzen eines realen Servers. In Zeile 2 der Abbildung 18 befindet

sich die Adresse, unter der der Server zu erreichen ist. Wird diese Adresse in den Browser eingegeben, ist die Startseite des Projekts zu sehen. Diese wurde vorher in der Datei „urls.py“ festgelegt.[Dja18]

```
1 | python manage.py runserver
2 | http://127.0.0.1:8000/
```

Abbildung 18: Befehl zum Starten des Servers

Ein großer Vorteil von DJANGO ist das Erstellen eines Datenbankmodells nach Klassen, die an die Objektorientierte Programmierung angelehnt sind. Dies funktioniert über Migrations, dies sind PYTHON-Programme, welche Änderungen des Datenmodells beschreiben. Ausgeführt werden sie mit dem Befehl 1 aus der Abbildung 19 „makemigrations“. Gab es nach der Ausführung des Befehls keine Fehlermeldung, befindet sich eine PYTHON-Datei im Ordner „migrations“. Um diese auf die Datenbank zu übertragen, muss der zweite Befehl in der Abbildung 19 („migrate“) ausgeführt werden. Dieses Vorgehen ermöglicht Nutzern ohne jegliche SQL-Kenntnisse eine Datenbank zu entwerfen und zu verwalten[Ernge].

```
1 | python manage.py makemigrations
2 | python manage.py migrate
```

Abbildung 19: Befehle für die Datenbank

2.2.5 Python concepts

concepts ist ein Framework, das die formale Begriffsanalyse in PYTHON umsetzt. Es wurde von Sebastian Bank entwickelt[Ban18] und bietet eine Vielzahl von Funktionen. So erleichtert es das Laden und Speichern von Kontexten, das Erstellen von Verbänden und das Navigieren auf diesen. Es bietet vorgefertigte Methoden um das Supremum und Infimum, sowie die unteren und oberen Nachbarn eines Begriffes abzufragen. Außerdem bietet es die Möglichkeit, nach Merkmal- und Gegenstandskombinationen zu suchen oder sich alle Gegenstände und Merkmale des Kontexts anzeigen zu lassen. Eine weitere Funktion des Framework *concepts* ist, dass die Liniendiagramme auch in unterschiedlichen Formaten visualisiert werden können.

2.2.6 JavaScript und D3

Für die Animationen und jegliche Interaktion, die nicht erst zum Server geschickt werden muss, sondern lokal, also beim Client bearbeitet werden kann, wird eine clientseitige Programmiersprache benötigt. Eine weitverbreitete clientseitige Programmiersprache ist JAVASCRIPT. Sie wird von allen gängigen Browsern interpretiert und kann über das HTML-Tag *script* eingebunden werden. In der Bestimmungssoftware ist JAVASCRIPT für die interaktiven Infografiken gedacht, die dem Nutzer den morphologischen Aufbau von Insekten erklären sollen. Es wird dafür das von Mike Bostock entwickelt Framework D3 benutzt[Bos19]. Eigentlich ist es, wie der Name *Data-Driven Documents* andeutet, zum Visualisieren großer Daten-Mengen gedacht und so dem Big-Data-Umfeld zugeordnet, doch durch seine Einfachheit findet es immer mehr Anhänger, die es in anderen Kontexten der Visualisierung verwenden. Beim Online-Bestimmungsschlüssel wird das Framework D3 hauptsächlich zum Animieren von SVGs und zum Anzeigen der Autovervollständigung genutzt.

2.2.7 css3

CSS steht für Cascading Style Sheets, das sind Gestaltungsanweisungen. Sie wurden hauptsächlich zur Verschönerung von Webseiten genutzt. In der Version CSS 3 werden auch Animationen unterstützt. Sie werden im Online-Bestimmungsschlüssel zur Verschönerung und zur besseren Nutzbarkeit eingesetzt. Die Nutzbarkeit wird mit sogenannten Media Queries verbessert. In Abbildung 20 ist ein Beispiel zu sehen. Dabei beschreibt die erste Zeile die Auflösung der Bildschirmbreite, unter der die Anweisungen gültig sind, also in diesem Beispiel alle Auflösungen, die unter 600 Pixeln groß sind. Alle Anweisungen, die in dem Klammerpaar (Zeile 1 und Zeile 5) eingeschlossen sind, werden dann ausgeführt, in dem Fall des Beispiels wird die Breite des *body*s auf 400 Pixel gesetzt. Der *body* ist das HTML-Element, in dem der größte Teil des Inhalts einer Webseite sein sollte.

```
1 @media screen and (max-width: 600px) {  
2   body {  
3     width: 400px;  
4   }  
5 }
```

Abbildung 20: Media Queries Beispiel

Die *Media Queries* eignen sich mit dem vorgestellten Prinzip besonders gut zum Erstellen von *Responsive-designs*. Ein *Responsive-design* ist ein flexibles Design, welches sich an die Auflösung des Bildschirms anpasst und so eine bestmögliche Benutzbarkeit für die jeweilige Auflösung gewährleistet. Es wäre auch eine Abfrage nach Endgeräten möglich: ist es ein Tablet, dann zeige dieses Design, ist es ein Smartphone zeige ein anderes Design. Da es aber zukünftig Endgeräte geben wird, die nicht in Gruppen heutiger Endgeräte passen werden, wie beispielsweise ein faltbares Smartphone, bietet sich die Abfrage nach Bildschirmgrößen eher an. Da die Bildschirmgrößen in Klassen eingeteilt werden können, lassen sich so die Bildschirmauflösungen unterschiedlicher Endgeräte abdecken.

2.2.8 Spracheingabe

Da die Eingabe von Daten für die Insektenarten per Tastatur sehr zeitraubend ist, wurden vier Opensource-Möglichkeiten der Spracheingabe dafür getestet. Dabei stach die Spracheingabe von GOOGLE DOCS[Goo18] besonders hervor. Diese ist ohne weiteren Aufwand nutzbar, sofern der Nutzer den CHROME Browser zum Besuchen der Webseite *docs.google.com* verwendet. In anderen Browsern steht die Spracheingabe leider nicht zur Verfügung. Nach der Öffnung eines Dokuments kann die Spracheingabe über den Menüreiter *Tools* aktiviert werden. Alternativ kann sie auch über die Tastenkombination Steuerungstaste, Umschaltaste und S-Taste aktiviert werden. Nach der Aktivierung erscheint ein *Pop-Up*-Fenster im Browser, auf dem ein Mikrophon-Icon zu sehen ist (siehe Abbildung 21). Dieses muss mit der Maus angeklickt werden, und danach kann der zu erfassende Text gesprochen werden. Gestoppt wird die Aufnahme ebenfalls wieder mit einem Klick auf das Mikrophon-Icon.

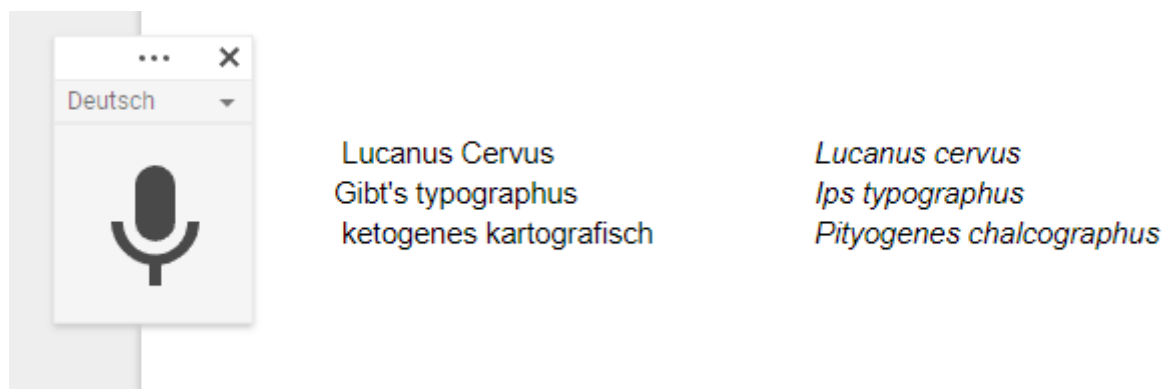


Abbildung 21: Beispieleingabe Googles Doc

Wie in Abbildung 21 zu sehen ist, ist gerade die Eingabe von wissenschaftlichen Namen schwierig. Bekanntere Namen wie *Lucanus cervus* erkennt das Tool zwar, aber schreibt sie nicht richtig. In der Abbildung enthält dabei die linke Spalte die

per Spracheingabe resultierenden Ergebnisse und die rechte Spalte die eingesprochenen Namen und wie sie geschrieben werden sollten. Bei *Lucanus cervus* fällt dabei auf, dass die Software den Artnamen groß schreibt. Die beiden anderen Namen erkennt er teilweise oder auch gar nicht. Generell ist die Erkennung besser, wenn ein Text vorgelesen wird, als wenn er frei gesprochen wird. Außerdem bedarf es immer wieder Nachbesserungen manueller Art, wenn die Erkennung beispielsweise den Steuerbefehl „Komma“ ausschreibt, anstatt ein Komma zu setzen. All diese Verbesserungen benötigen Zeit. Hinzugerechnet zu der Zeit, welche für das Einsprechen benötigt wird, besteht keine Zeiteinsparung gegenüber dem Abschreiben der Texte. Dies gilt gerade für Nutzer, die einen schnellen Anschlag an der Tastatur haben, also schnell „tippen“ können.

3 Implementierung

In dem Abschnitt Implementierung werden beispielhaft für den ganzen Online-Bestimmungsschlüssel besonders wichtige, schöne oder schwer verständliche Implementierungen vorgestellt, damit diese gut nachzuvollziehen sind und das Projekt bestenfalls lange weiter gepflegt werden kann.

3.1 Die Implementierung der Bestimmung

In diesem Abschnitt wird die Implementierung und Funktion der Bestimmung mittels der formalen Begriffsanalyse vorgestellt. Dabei werden im Abschnitt *Clientseite der Bestimmung* der Code vorgestellt, welcher clientseitig, also im Browser, gerendert wird, und im Abschnitt *Serverseite der Bestimmung* der Quellcode von django.

3.1.1 Clientseite der Bestimmung

Der Quellcode für die Webseite der Bestimmung auf der Grundlage der formalen Begriffsanalyse befindet sich in der *bestimmung.html*. Der gesamte Inhalt der Datei wird in Abbildung 22 aufgelistet. In Zeile 1 erbt die *bestimmung.html* von der *base.html* die Funktionen, die sich alle Webseiten des Online-Bestimmungsschlüssels teilen, diese sind zum Beispiel der Header und die Navigation. Die Zeilen 2 und 38 begrenzen den Quellcode, welcher in die *base.html* eingefügt werden soll. In Zeile 4 wird angewiesen, dass der Quellcode zwischen den Zeilen 4 und 37 für jedes Triple ausgeführt wird, welches von der Methode *post_attr* des Moduls *views.py* übergeben wurde. Ein Triple besteht dabei aus den drei Komponenten, und zwar Insektenarten, Merkmalen und Merkmalen, welche nicht in der Schnittmenge der Merkmale der anderen Begriffe sind, welche an diesem Punkt der Bestimmung vorgeschlagen werden. Insektenarten werden der Variablen *insects* zugewiesen. Merkmale werden der Variablen *attrs* zugewiesen und Merkmale, die nur in diesem Formalen Begriff vorkommen, werden der Variablen *new_attr* zugewiesen.

```

1 {% extends 'base.html' %}
2 {% block content %}
3 {%load static%}
4 {% for insects, attrs, new_attr in attributes %}
5   <div class="begriff">
6     <p>
7       <b>
8         <i>
9           {% for insect in insects %}
10            <a href="/insect/{{ insect }}/">
11              {{ insect }}
12            </a>,
13          {% endfor %}
14        </i>
15      </b>
16    </p>
17    {%load check_infimum%}
18    <div class="begriff">
19      {% for a in new_attr %}<p>{{a}}</p>
20      {%load get_has_image%}
21      {%if a|get_has_image %}
22        
25        </img>
26      {% endif %}
27    {% endfor %}</div>
28    <form method="POST" >{% csrf_token %}
29      <input hidden id="attr" type="text" name="attr"
30      value="{%for attr in attrs%}{{attr}};{%endfor%}">
31      {%if attrs|check_infimum %}
32        <input class="weiter_button" type="submit"
33        value="hier weiter">
34      {% endif %}
35    </form>
36  </div>
37 {% endfor %}
38 {% endblock %}

```

Abbildung 22: Die HTML-Datei der Bestimmung

Als Beispiel dient der erste Begriff, der in Abbildung 22 zu sehen ist. In der ersten Iteration haben die Variablen folgenden Inhalt:

- insect = 'Bupalus piniaria', 'Dendrolimus pini', 'Limenitis populi', 'Lymantria monacha', 'Operophtera brumata', 'Panolis flammea'
- attrs = 'Flügel 1 oder 2 Paar größer gleich dreiviertel Hinterleibslänge', 'Flügel nicht bandförmig ohne Fransensäume', 'voll entwickelte Flügel 2 Paar', 'Flügel Oberfläche mit Haaren und oder schuppenbedeckt', 'Vorder- u. Hinterflügel bunt Bzw. schwarz-weiß beschuppt',
- new_attr = 'Flügel Oberfläche mit Haaren und oder schuppenbedeckt', 'Vorder- u. Hinterflügel bunt Bzw. schwarz-weiß beschuppt'

Dabei sind die grünen Merkmale die Merkmale, die sich die Insektenarten mit dem *Diprion pini* teilen, und so gehören die Merkmale zu beiden Begriffen. Die roten Merkmale hingegen sind die Merkmale, die die Insektenarten vom *Diprion pini* unterscheiden, also die Merkmale, welche für eine weitere Bestimmung angezeigt werden sollten. Da beide Begriffe die grünen Merkmale teilen, können diese zur Unterscheidung in diesem Schritt der Bestimmung nicht beitragen und wurden deshalb weggelassen.

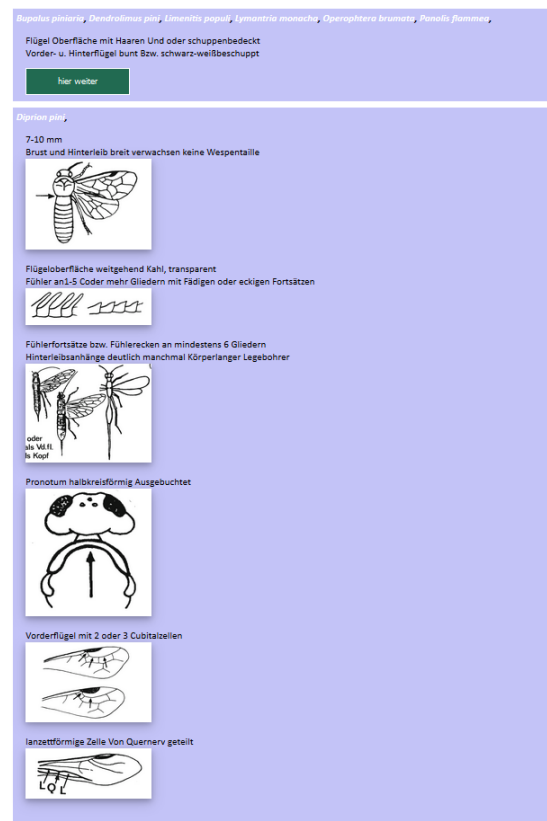


Abbildung 23: Bestimmungseite der Insektenarten

In Zeile 17 der Abbildung 22 wird der DJANGO-Filter `check_infimum` geladen, der in Zeile 31 darüber entscheidet, ob es einen Weiterbutton gibt oder nicht, je nachdem ob die aktuelle Position im Begriffsverband Nachbar des Infimums ist oder nicht. Genauer erklärt wird der Filter in Abschnitt 3.5. Die Auswirkung auf die HTML-Seite ist dabei in der Abbildung 23 zu erkennen. Der obere Begriff (lila blauer Hintergrund) ist noch nicht fertig bestimmt, deshalb gibt es einen Weiterbutton, der Begriff darunter hingegen beinhaltet nur noch die Insektenart *Diprion pini* und ist damit fertig bestimmt, daher gibt es keinen Weiterbutton. In Zeile 28 wird ein HTML-Formular mit dem Tag `form` erstellt. Damit das Formular Daten an DJANGO senden darf, muss ein `csrf-Token`[dja19a] gesetzt werden. `csrf` steht dabei für Cross Site Request Forgery protection und verhindert zum Beispiel eine SQL-Injektion, dabei werden SQL-Befehle eingegeben, um damit etwas über die Datenbank zu erfahren oder diese zu löschen. Auch andere Angriffe auf Server oder Datenbank werden mit dem Token verhindert. Wie genau django aus dem HTML-Template eine Webseite erstellt, wird in dem Abschnitt 3.3 genauer erklärt.

3.1.2 Server der Bestimmung

In diesem Abschnitt wird der Server-Code oder auch django-Code vom Online-Bestimmungsschlüssel vorgestellt, welcher die Bestimmung auf Grundlage der formalen Begriffsanalyse ermöglicht. Dieser spiegelt die programmierte Logik des Projekts wieder und ist aufgebaut auf dem PYTHON Framework CONCEPTS von Sebastian Bank[Ban18]. Der Code befindet sich in der Methode `post_attr` in dem Python-Modul `view.py`, zu sehen ist er in Abbildung 24. Die Methode ist dabei nach ihrer Funktion benannt, und zwar ist sie benannt nach der *Hypertext Transfer Protocol(HTTP)* Anfragemethode POST[sel19]. Der zweite Teil des Namens kommt von dem Variablennamen `attr`. Dies steht für Attribute und soll als eine Art sprechender Bezeichner dem Leser des Quellcodes gleich vermitteln, dass hier die Attribute oder auch Merkmale der Insektenarten vom HTML-Client an den Server geschickt werden. In Zeile 2 der Abbildung 24 wird abgefragt, ob es sich um die Anfragemethode POST handelt. Ist dies nicht der Fall, wird zum `else`-Zweig der Abfrage gesprungen, der sich in Zeile 26 befindet. Dort wird die Methode `list_objects_attrglobal` aufgerufen, welche die Nachbarn des Supremums zurückliefert. Diese werden an die `render`-Methode von DJANGO weitergereicht und erstellen die Startwebseite der Bestimmung ohne Eingabe. Es wird also die Webseite erstellt, die beim Klick auf den Navigationspunkt Bestimmung geöffnet wird. Hat der Nutzer aber eine Eingabe getätigt, per Klick auf den Button *hier weiter* oder per Eingabe der Merkmale auf der Seite *Bestimmung mit Eingabe*, dann handelt es sich dabei um eine POST Anfrage. Damit ist auch die `if`-Abfrage auf POST `True` und der Code von Zeile 3 bis 25 wird ausgeführt. Dabei werden in den Zeilen 4 bis

14 die übergebenen Merkmale aus dem POST herausgenommen, Whitespaces und Transferdaten gelöscht und eine String-Liste der Merkmale namens *form* erstellt. Diese wird an die Methode *get_neighbours* übergeben, die im Verband nachschaut, welcher Begriff am besten zu den Merkmalen passt, welche mit *form* übergeben wurden, und sie gibt dann die Nachbarn des Begriffs Richtung Infimum zurück. Diesen Nachbarn wird eine Liste *z* angefügt. Die Liste wurde mit einem Python Generator erstellt, zu sehen in Zeile 23. Dabei ist *x* die Laufvariable, *j* die Laufvariable der äußeren Schleife über alle Merkmale, und *form* sind die Merkmale, die per POST gesendet wurden. Es wird an *z* dabei jedes Merkmal aus *j* anhängt, welches sich nicht auch noch in *form* befindet.

```

1 def post_attr(request):
2     if request.method == "POST":
3         global listobjects
4         form = []
5         formtemp = request.POST.get("attr", "")
6         try:
7             if type(formtemp) is not None:
8                 formtemp = formtemp.split(';')
9                 formtemp.pop(-1)
10        except TypeError:
11            print('Error')
12        for i in formtemp:
13            if i is not None:
14                form.append(i.strip())
15        listobjects = get_neighbours(form)
16        x = []
17        y = []
18        z = []
19        if type(listobjects) == list or type(listobjects)
20            == tuple:
21            for i, j in listobjects:
22                x.append(i)
23                y.append(j)
24                z.append([x for x in j if x not in form])
25            attr = list(zip(x,y,z))
26        return render(request, 'Bestimmung/bestimmung.html',
27            {'attributes': attr})
28    else:
29        return render(request, 'Bestimmung/bestimmung.html',
30            {'attributes': listobjects_attrglobal()})

```

Abbildung 24: Der DJANGO-Code der Bestimmung

Anschließend wird in Zeile 24 eine Liste aus Tripeln der Variablen *attr* zugewiesen. Dabei erstellt hier die Python Methode *zip* aus drei Listen Triple, welche mittels der *list* Methode zu einer Liste werden. Diese Liste wird dann als *attr* Variable in Zeile 25 dem HTML-Template übergeben und auf der Webseite durchiteriert und pro Iteration als ein Begriff ausgegeben.

3.2 Das Datenbankmodell

Das Datenbankmodell des Online-Bestimmungsschlüssels besteht aus zwei Klassen. Dies sind die Klasse *Insect*, die alle Daten über die Insektenart enthält, und die Klasse *Merkmal*, diese Klasse repräsentiert die Merkmale, die zur Bestimmung gebraucht werden. Beide Klassen stehen zueinander in einer *n zu m* oder auch *many to many* Relation. Dies bedeutet, dass ein Insekt mehrere Merkmale haben kann und ein Merkmal mehreren Insekten zugeordnet werden kann.

```
1 from django.db import models
2 class Insect(models.Model):
3     wissenschaftlicher_name =
4     models.CharField(max_length=100, unique=True)
5     deutscher_name =
6     models.CharField(max_length=100, blank=True, null=True)
7     [...]
8     forstliche_bedeutung =
9     models.TextField(null=True, blank=True)
10
11     merkmal = models.ManyToManyField(Merkmal)
12
13     class Meta:
14         ordering = ('wissenschaftlicher_name',)
15
16     def __str__(self):
17         return self.wissenschaftlicher_name
```

Abbildung 25: Datenbank Klasse der Insekten

In Abbildung 25 wird in der ersten Zeile die Klasse *models* von DJANGO importiert. Diese stellt Funktionen zur Verfügung, die dazu dienen das Datenbankmodell zu erstellen. *models.Model* wird wie in Zeile 2 dem Konstruktor der Klasse *Insect* übergeben, wodurch die Klasse zu einem Datenbankmodell wird. Außerdem werden die Datenbankfelder mittels *models* zugewiesen, dies ist in Zeile 3 und 4 zu sehen. Dabei wird ein Feld mit dem Namen *wissenschaftlicher_name* erzeugt. Der

Typ des Feldes ist *CharField*, das bedeutet, dass es hauptsächlich für die Speicherung von Wörtern genutzt wird. Der Methode *CharField* werden in Zeile 4 zwei Argumente übergeben, diese sind *max_length=100* und *unique=True*. *max_length=100* gibt die maximale Länge des Wortes an, in diesem Fall darf das Wort oder der wissenschaftliche Name der Insektenart einhundert Zeichen haben. Das Argument *unique=True* stellt sicher, dass der wissenschaftliche Name der Insektenart nur einmal vorkommt. Es kann so in der Datenbank keine zwei oder mehr Insektenarten mit dem gleichen Namen geben. In Zeile 11 wird die Relation der Datenbanktabelle *Insect* und der Datenbanktabelle *Merkmal* hergestellt. Dies wird mit der *ManyToManyField* Methode von *models* gemacht, der als Argument die selbsterstellte Klasse *Merkmal* (siehe Abbildung 26) übergeben wird. In der Subklasse *Meta* werden die Metadaten für die obere Klasse, hier die Klasse *Insect*, festgelegt. Dabei handelt es sich in Zeile 14 um die Festlegung der alphabetischen Sortierreihenfolge, also nach welchem Feld der Datenbank sortiert werden soll. Es wird, wie in der Abbildung zu sehen, nach dem wissenschaftlichen Namen sortiert. Außerdem wird die *__str__* Methode überschrieben, so dass sie nicht mehr die Stringrepräsentation des Objekts zurückliefert, sondern den wissenschaftlichen Namen der Insektenart.

```

1 class Merkmal(models.Model):
2     merkmal = models.CharField(max_length=10000, unique=True)
3     has_image = models.BooleanField(default=False)
4
5     class Meta:
6         ordering = ('merkmal',)
7
8     def __str__(self):
9         return self.merkmal

```

Abbildung 26: Datenbank Klasse der Merkmale

In der Abbildung 26 ist die Klasse *Merkmal* zu sehen; diese enthält die Merkmale, die für die Bestimmung mittels formaler Begriffsanalyse benötigt werden. Da es bei den Merkmalen ebenfalls keine zwei Merkmale geben soll, die den gleichen Inhalt haben, wurde auch hier *unique* auf *True* gesetzt. Da es nicht zu jedem Merkmal auch ein Bild gibt, hat die Klasse *Merkmal* ein *BooleanField*, zu sehen in Zeile 3 der Abbildung 26, es heißt *has_image* und gibt Auskunft darüber, ob ein Bild zu dem Merkmal vorhanden ist (*True*) oder nicht (*False*). Der voreingestellte Wert wurde dabei auf *False* gesetzt, wird also keine Angabe beim Erstellen eines Merkmals zum Vorhandensein eines Bildes gemacht, wird der *Boolean* auf *False* gesetzt.

3.3 Wie django eine HTML-Seite erstellt

In diesem Abschnitt wurde ein Minimalbeispiel aus dem Online-Bestimmungsschlüssel entnommen, um den Zusammenhang der Datenbank, DJANGO und der daraus erzeugten HTML-Seite aufzuzeigen.

```
1 {% extends 'base.html' %}
2 {% block content %}
3 <body>
4 <h3>Alle Insekten in der Bestimmung:</h3>
5 {% for name in names %}
6 <div class="begriff">
7     <form method="POST" >{% csrf_token %}
8     <a href="/insect/{% name %}"/>
9     <p>
10        <b>
11            <i>
12                {% name %} - {% name.deutscher_name %}
13            </i>
14        </b>
15    </p>
16 </form>
17 </div>
18 </a>
19 {% endfor %}
20 </body>
21 {% endblock %}
```

Abbildung 27: Einfaches Beispiel eines HTML Templates

Auf der Abbildung 27 ist der gesamte Quellcode der Datei *list_all.html* zu sehen. Die Datei ist ein DJANGO-Template. Das bedeutet, in ihr wird festgelegt, was auf der fertigen HTML-Seite zu sehen ist. In Zeile 1 erbt das Template von einem anderen Template mit dem Namen *base.html*, dies enthält den Header und die Navigation des Online-Bestimmungsschlüssels. Der Vorteil einer solchen Auslagerung ist, dass, wenn sich etwas an der Navigation oder dem Header ändert, dies nur in der Datei *base.html* geändert werden muss und nicht auf allen HTML-Seiten einzeln. Es wird dabei alles zwischen den Anweisungen `{% block content %}` in Zeile 2 und `{% endblock %}` in Zeile 21 in die Vorlage *base.html* eingefügt. Dort gibt es diese beiden Zeilen auch, und genau an der Stelle, wo diese sich befinden, wird der Inhalt der anderen Datei eingefügt. Soll auf die von View übergebenen Variablen zugegriffen werden, dann passiert das über den Namen der Variablen.

Dieser wird in der *view.py* als Tuple festgelegt (siehe Abbildung 28 Zeile 4). Der Name der Variablen wird in doppelte geschweifte Klammern geschrieben, dies ist die Syntax vom Python, um den Inhalt der Variablen auszugeben. Zu sehen ist dies in Abbildung 27 in Zeile 12, nur dass in dem Beispiel vom *view.py* eine Liste namens *names* übergeben wurde und in Zeile 5 durch die Liste durchiteriert wird, die Laufvariable dieser Iteration wurde *name* genannt. Die doppelten geschweiften Klammern rufen die überschriebene `__str__`-Methode auf, die den wissenschaftlichen Namen zurückgibt. Ein Bindestrich trennt den wissenschaftlichen und den deutschen Namen der Insektenart. Der deutsche Name wird über den Aufruf *name.deutscher_name* ausgegeben.

```
1 def post_list(request):
2     names = Insect.objects.all()
3     return render(request, 'all/all_insects.html',
4                   {'names': names})
```

Abbildung 28: Einfaches Beispiel einer View-Methode

Die Methode *post_list* des Moduls *view.py* ist in Abbildung 28 zu sehen. Sie bekommt als Argument den *request*, also die Anfrage des Webseiten-Aufrufs. In Zeile 2 werden der Variablen *names* alle Insektenarten der Datenbank zugewiesen. Diese werden der Methode *render* als Argument übergeben. Außerdem bekommt die Methode noch ein *request* als Standardparameter übergeben, dies fordert DJANGO, und einen Pfad, der angibt, wo sich das Template, nach dem gerendert werden soll, befindet.

```
1 urlpatterns = [
2     path('list_all', views.list_all, name='list_all'),
3     path('all', views.post_list, name='post_list'),
4     [...]
5     path('insect/<str:insect>/', views.insect),
6 ]
```

Abbildung 29: *urls.py*

Damit DJANGO weiß, welche Methode es bei welcher eingegebenen Browseradresse aufrufen soll, wird dies in der *urls.py* eingetragen. Abbildung 29 zeigt einen Ausschnitt der *urls.py*. In Zeile 3 wird mit der *path*-Methode gesagt, dass, wenn an die URL-Adresse */all* dran gehängt wird, die Methode *post_list* des Moduls *view.py* aufgerufen wird. Der Name, der am Ende der Methode noch vergeben wird, dient

dazu, dass die Methode auch über andere Wege aufrufbar ist, beispielsweise als Aktion eines Button-Klicks.



Abbildung 30: Die Auswahlseite der Insektenarten

Die fertig gerenderte HTML-Seite ist auf der Abbildung 30 zu sehen. Dabei sind der blaue Header und das grüne Navigationsmenü aus der *base.html* und die Überschrift *Alle Insekten in der Bestimmung:* und die Aufzählung der Insekten aus der *list_all.html*. Die Insektenarten werden aufgrund der Meta-Klasse in einer nach dem wissenschaftlichen Namen alphabetisch geordneten Aufzählung ausgegeben.

3.4 Das Python-Modul *view.py*

Das Python-Modul *view.py* bildet das Herzstück eines jeden Django-Projekts. Es verbindet sich mit der Datenbank, falls eine vorhanden ist, und rendert mit Hilfe der HTML-Templates die anzuzeigende HTML-Seite. Es wird in diesem Abschnitt nicht der ganze Quellcode der *view.py* vorgestellt, sondern zum einen Funktionen, die für den Online-Bestimmungsschlüssel besonders wichtig sind, und zum anderen Funktionen, die stellvertretend für die Erklärung anderer, ähnlicher Funktionen stehen.

3.4.1 Die Methode *create_context_from_db*

In der Methode *create_context_from_db* wird der Formale Kontext für den Online-Bestimmungsschlüssel gebildet. Dazu wird, wie in Abbildung 31 in Zeile 2 zu sehen ist, eine leere Definition erzeugt. In Zeile 3 wird anschließend über alle Insektenarten in der Datenbank iteriert und deren Merkmale an eine Liste angehängt (Zeile 4-6). In Zeile 7 wird die Insektenart und eine Liste von Merkmalen in die Defi-

nition aufgenommen. In Zeile 9 wird ein Formaler Kontext kreiert und dieser als Rückgabewert der Funktion dem Aufrufer übergeben.

```
1 def create_context_from_db():
2     deffca = Definition()
3     for n in Insect.objects.all():
4         list_probs = []
5         for m in n.merkmal.all():
6             list_probs.append(m.__str__())
7         deffca.add_object(n.__str__(), list_probs)
8
9     return Context(deffca.objects, deffca.properties,
                    deffca.bools)
```

Abbildung 31: Erstellen des Formalen Kontext

Der Aufruf befindet sich in keiner Methode, sondern wird bei Starten des Servers durch den imperativen Ablauf von Python ausgeführt. Wichtig ist noch die Zuweisung des Verbands, auf dem die Bestimmung durchgeführt wird, zu sehen ist dies in Abbildung 32 Zeile 2. Dabei liefert `c.lattice` den Verband zurück und dieser wird `l` zugewiesen, welches eine globale Variable ist und von der ganzen `view.py` genutzt wird.

```
1 c = create_context_from_db()
2 l = c.lattice
```

Abbildung 32: Aufruf und Zuweisungen der Methode

Ein neues Erstellen des Begriffsverbandes ist nötig, wenn neue Insektenarten erstellt wurden, bestehende Insektenarten verändert wurden oder Merkmale verändert wurden. Dass dies geschieht, ist dadurch sichergestellt, dass die Methode `create_context_from_db` von jeder anderen Methode, die etwas an den Datenbanktabellen `Insect` oder `Merkmal` ändert, nach der Änderung aufgerufen wird. So ist sichergestellt, dass immer ein aktueller Verband vorliegt und dieser nicht von den Daten in der Datenbank verändert wird.

3.4.2 Die Methode *insect*

Bei der Methode `insect` handelt es sich um die Methode, die die Seite erstellt, auf der die fertig bestimmten Insektenarten angezeigt werden. Auf Abbildung 33

ist als Beispiel die Informationsseite der Insektenart *Melolontha hippocastani* des Waldmaikäfers zu sehen. Je nachdem ob es mehr als ein Foto zu der Insektenart gibt, wird ein Foto-Slider angeboten, um alle Fotos zu betrachten. Der Slider ist mit JAVASCRIPT geschrieben und ist von der Internetseite w3schools.com[w3s19] entnommen. Er wurde so angepasst, dass DJANGO den Bilderpfad automatisch nach der Anzahl der Bilder im Ordner der jeweiligen Insektenart an die HTML-Seite übergibt. Dort kann dann durch Klicken auf einen der weißen Pfeile am Bildrand das nachfolgende oder vorangegangene Bild angezeigt werden. Diese Pfeile werden auch nur dann angezeigt, wenn die Insektenart mehr als ein Bild hat.



Melolontha hippocastani

Deutschername: Waldmaikäfer
Gattung: Melolontha
Unterfamilie: Melolonthinae
Familie: Scarabaeidae
Unterordnung: Polyphaga
Ordnung: Coleoptera
Anzeigemerkmale: Länge 22-26 mm. Unterscheidung zum Feldmaikäfer: Spitze des Telsons knotig erweitert.
Überwinterung als: 3 x mal Larve, zuletzt als Imago
Generationen: Zumeist 4 jährig, 3-5 jährige Larvenzeit möglich (je nach Standortbedingungen)
Flugzeiten: Ende April-Anfang Juni
Hauptwirtspflanzen: Eiche
Nebewirtspflanzen: Buche, Ahorn
forstliche Bedeutung: In Süddeutschland in Sandgebieten teilweise keine Bewaldung mehr möglich, Larven fressen an Wurzel, Blattschaden durch Käferfraß wird durch Johannistriebe kompensiert, Auftreten von Kalamitäten

Abbildung 33: Die Informationsseite der Insektenart

Unter dem Bild befinden sich noch alle Informationen über die Insektenart, die sich in der Datenbank befinden. Dabei wurden die Merkmale zur Bestimmung weggelassen, da die Informationen nur über diese eine Insektenart Auskunft geben sollen. Ist eines der Felder der Datenbank für die Insektenart leer oder es befindet sich der Datentyp *None* in dem Datenbankfeld, dann wird dies auf der HTML-Seite nicht angezeigt. Als Beispiel wird das Datenbankfeld *Tribus* hier nicht angezeigt, da es nichts enthält. Die angezeigten Daten stammen aus Herrn Rohes Insektenliste. Eine Besonderheit bei der Erstellung der HTML-Seite der Insektenart ist, dass die Insektenart per URL-Parameter übergeben wird. Dabei wird nicht, wie üblich, ein Schlüssel-Wert-Paar übergeben[wik19b], sondern ein in DJANGO sogenannter URL-dispatcher[dja19b]. Dieser URL-dispatcher wird vorher in der *urls.py* mit einem Datentyp festgelegt. Zu sehen ist dies in Abbildung 34 in Zeile 1. Dort wird an den Namen der URL, *insect* in zwei Schrägstriche gehüllt, der Datentyp

mit Doppelpunkten getrennt von dem Namen, unter dem der Parameter abgefragt werden kann, eingegeben. Der Datentyp ist hier ein String, der mit *str* abgekürzt wurde, und der Parametername ist *insect*. Die Adresse mit dem URL-dispatcher wird durch die *path*-Methode deren zweitem Parameter (die Methode *views.insect*) zugewiesen.

```
1 | path('insect/<str:insect>/', views.insect),
```

Abbildung 34: Übergabe Parameter per GET

In der Methode *insect* des Moduls *views.py* wird der Parameter als zweites Argument(*insect*) der Methode übergeben. Zu sehen ist dies in Abbildung 35 in Zeile 1. In der Methode kann dann ohne weiteren Aufwand der Parameter als normale Variable behandelt werden. In Zeile Zwei werden alle Informationen der Insektenart aus der Datenbank geholt und der Variablen *insect_db* zugewiesen. Eine Liste mit den Pfaden zu den Bildern wird in Zeile 3-6 erstellt, dabei wird in Zeile 4 der Pfad zum Ordner der Insektenart angegeben, in dem die Bilder sind. Anschließend wird mit einer *for*-Schleife (Zeile 5) über die Bilder iteriert und diese an die Liste *images* angehängt(Zeile 6). Diese Liste wird am Ende der Funktion, zusammen mit den Informationen über die Insektenart, mit einer *return* Anweisung an das HTML-Template weitergegeben.

```
1 | def insect(request, insect):
2 |     insect_db = Insect.objects.get(
3 |         wissenschaftlicher_name = insect)
4 |     images = []
5 |     path = os.path.join( os.path.dirname ( __file__ ))
6 |         + "/static/img/arten/" + insect
7 |     for name in os.listdir(path):
8 |         images.append(name)
9 |     return render(request, 'all/insect.html',
10 |        {'insect':insect_db, 'images':images})
```

Abbildung 35: Die Methode insect

3.4.3 Die Methode *upload_image*

Die Methode *upload_image* dient zum Hochladen des Bildes eines Merkmales. Dabei wird das Bild als Datei als Rückgabewert der *request.FILES* Methode der Variablen *inMemoryUploadedFile* zugewiesen, dies ist in Abbildung 36 in Zeile 2 zu sehen. Die Methode liefert aber nur Binärdaten, die wieder in ein Bild umgewandelt

werden müssen. Dies geschieht in Zeile 3 der Abbildung 36. Dabei kommt das PYTHON Modul PIL[[Cla19](#)] zum Einsatz. Danach, in Zeile 4, wird die Endung der Datei ausgelesen; da die meisten Bildformate drei Buchstaben haben, reicht es hier aus, die letzten vier Literale des Strings zu entnehmen. Vier Literale, da der Punkt mitgenommen wird, weil dieser zur Speicherung des Bildes benötigt wird. In Zeile 5 wird der Name des Merkmals ausgelesen und in Zeile 6 zusammen mit dem Format des Bildes als Name des Bildes zusammengefügt. Als Beispiel wird das Bild des Merkmals *Flügel einfarbig*, zu dem es ein Bild im PNG-Format gibt, serverseitig als *Flügel einfarbig.PNG* gespeichert. Dann wird das Bild in dem Ordner *bestimmung* gespeichert, dies ist zu sehen in Zeile 7. Nach erfolgreichem Hochladen des Bildes erfolgt in Zeile 8 und 9 eine Rückmeldung an den Nutzer, dass das Bild erfolgreich hochgeladen wurde. Hierfür dient die *feedback.html*, die für Rückmeldungen oder Fehlermeldungen gedacht ist. Die *feedback.html* bekommt die Variable *notice* und gibt diese auf der Webseite aus. Neben der Variable *notice* kann der *feedback.html* auch eine Variable *error* oder *insect* übergeben werden. Diese Variablen werden in der Reihenfolge *notice*, *insect* und *error* ausgegeben, um im Fehlerfall möglichst viele Information zu erhalten. *notice* sollte dabei Informationen darüber liefern, was gemacht wurde. Die Insektenart, bei der etwas gemacht wurde, liefert die Variable *insect*, und der Fehler wird mit der Variablen *error* beschrieben.

```

1 def upload_image(request):
2     inMemoryUploadedFile = request.FILES['datei']
3     image = Image.open(inMemoryUploadedFile)
4     format = request.FILES['datei'].name[-4:]
5     name = request.POST.get("merkmal", "")
6     name = name + format
7     image.save(os.path.dirname(os.path.realpath(
8         __file__))+"/static/img/bestimmung/"+name)
9     notice = "bild erfolgreich hochgeladen"
    return render(request, 'notice/feedback.html',
        {'notice': notice} )

```

Abbildung 36: Die Methode `upload_image`

3.5 Django-Template-Filter

Mit Django-Template-Filtern kann clientseitig die Darstellung von serverseitig gesendeten Daten manipuliert werden. Im Beispiel in Abbildung 37 wird überprüft, ob der Formale Begriff des Begriffsverbandes das Infimum ist oder dessen Nachbar. In beiden Fällen wäre die Bestimmung zu Ende. Wenn der Begriff Nachbar das Infimum ist, ist die Insektenart fertig bestimmt. Sollte der Begriff selbst das Infi-

mum sein, wurde bei der Eingabe eine Merkmalskombination ausgewählt, welche bei keiner Insektenart vorkommt, und nur das Infimum erfüllt diese Merkmalskombination. Um diese Überprüfungen aus dem Template aufrufen zu dürfen, muss das Modul als Django-Filter registriert werden. Dies geschieht in Zeile 4 und 6 der Abbildung 37. Nach Aufruf der Methode wird zuerst eine interne Repräsentation des Begriffsverbands erstellt (Zeile 8 und 9). Anschließend wird mittels übergebener Merkmale deren Position im Verband abgefragt. Dies geschieht durch $l[value]$, und das Ergebnis davon wird mit dem Infimum des Verbands verglichen. Sollte die Position nicht dem Infimum entsprechen, folgt eine Abfrage, ob das Infimum nicht ein Nachbar der aktuellen Position des Verbandes ist. Diese Art der Abfrage, also die negierte Abfrage, hat den Vorteil, dass im Template selbst nicht mehr negiert werden muss. Es wird nun also ein *True* zurückgeliefert, wenn die aktuelle Position im Verband nicht der Nachbar des Infimums ist, und *False*, wenn die Position Nachbar des Infimums ist.

```

1 from django import template
2 from Bestimmung.views import create_context_from_db
3
4 register = template.Library()
5
6 @register.filter
7 def check_infimum(value):
8     c = create_context_from_db()
9     l = c.lattice
10    if (l[value] == l.infimum):
11        return False
12    else:
13        return not l.infimum in l[value].lower_neighbors

```

Abbildung 37: Ein Django-Template Filter

In der HTML-Datei muss der Template-Filter erst geladen werden. Dies ist in Abbildung 38 in Zeile 1 zu sehen. Aufgerufen wird der Filter dann, wie eine Methode, über seinen Namen, zu sehen in Zeile 3. Die Syntax der Übergabe der Argumente erinnert dabei an das Popen in der Unixshell. Es wird hier also die Variable *attr* durch den Querstrich an die Methode *check_infimum* übergeben.

```

1 {%load check_infimum%}
2 [...]
3 {%if attrs|check_infimum %}
4 <input class="weiter" type="submit" value="hier weiter">
5 {%endif %}

```

Abbildung 38: Aufruf eines Django-Template Filters

Der Button in Zeile 4 wird nur angezeigt wenn *check_infimum* *True* zurück liefert, also wenn die aktuelle Position nicht das Infimum oder dessen Nachbar ist und so die Bestimmung weiter gehen soll. Zur weiteren Bestimmung dient dieser Button auch, denn auf dessen Klick werden dem Nutzer die Nachbarn der aktuellen Position in Richtung Infimum hin vorgeschlagen.

3.6 Infoseite mit d3 und SVG

Unter dem Hauptmenüpunkt „Info“ befindet sich eine interaktive Erklärung der Morphologie von Insekten. Da noch weitere Erklärungsseiten oder beispielsweise ein Glossar, das die Begriffe erklärt, die Benutzung der Seite erleichtern würden, wurde der Name des Menüpunktes so gewählt, dass solche Erweiterungen dann unter dem Hauptmenüpunkt „Info“ eingegliedert werden können. Auch die Position des Auswahlpunktes ist angelegt an andere Webseiten, wo erklärende Verlinkungen sich meist im rechten oberen Drittel einer Webseite befinden sollen.

Die Abbildung 39 zeigt einen Ausschnitt der HTML-Seite mit einem Insektenquerschnitt. Die SVG-Datei, die ein Insekt darstellt, wurde von Wikipedia entnommen und der Übersicht halber leicht verändert[Jaw19]. Ebenfalls auf der Abbildung 39 zu sehen ist der Hovereffekt des Mauszeigers, bei der Abbildung ist der Mauszeiger gerade über den gezeichneten Mandibeln oder dem Wort Mandibel-Kiefer. Dies ist daran zu erkennen, dass beide grün sind.

Morphologie der Insekten

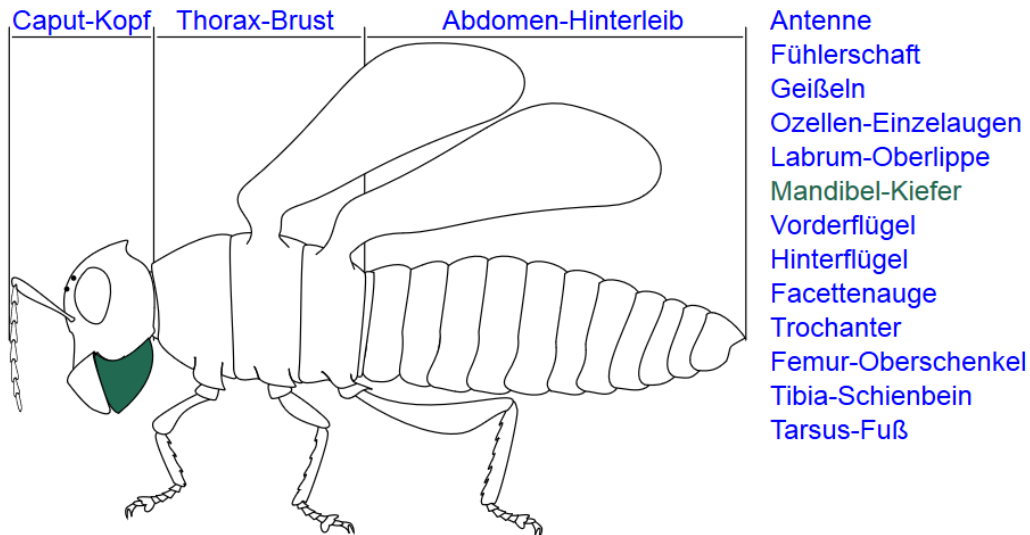


Abbildung 39: Interaktive Insektenmorphologie Auswahl Mandibel

Wenn der Mauszeiger nun über ein anderes Wort oder Körperteil des Insekts bewegt wird, werden diese beiden grün gefärbt. Dies soll eine schnellere und eindeutiger Zuordnung ermöglichen.

Morphologie der Insekten

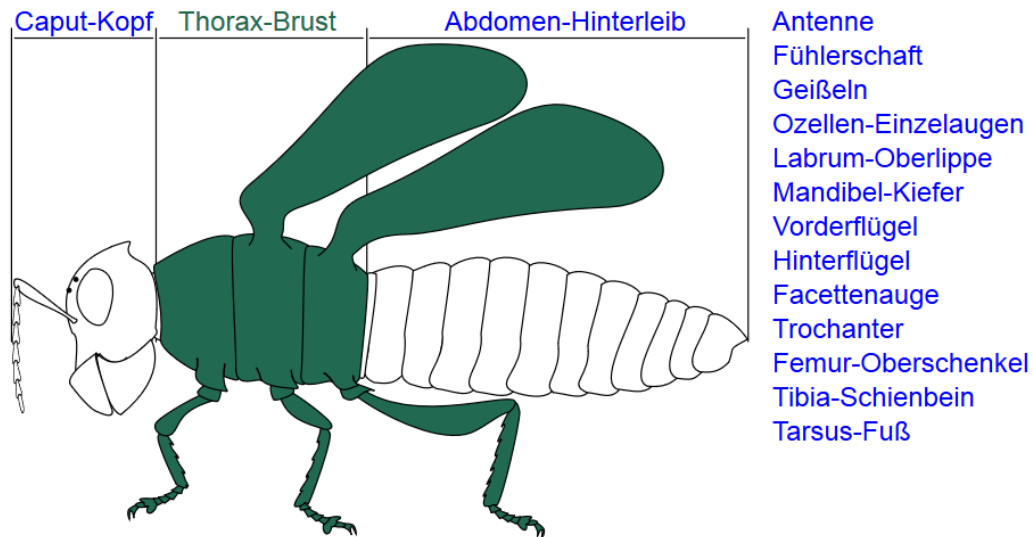


Abbildung 40: Interaktive Insektenmorphologie Auswahl Thorax

Bei der Abbildung 40 befindet sich der Mauszeiger über dem Wort Thorax-Brust, dadurch wird wieder der Hovereffekt ausgelöst und so das Wort und der zugeordnete Bereich in der SVG-Abbildung grün gefärbt. Dies funktioniert aber nur einseitig, da davon auszugehen ist, dass der Nutzer den Mauszeiger über den Flügel bewegt, um zu wissen, was da unter dem Mauszeiger ist, und nicht, zu welchem Körperbereich es gehört. Beide Wörter grün anzuzeigen, also in diesem Fall die

Wörter *Thorax-Brust* und *Vorderflügel*, könnte den Nutzer verwirren, da dies dann nicht mehr eindeutig ist. Der Nutzer müsste dann einen größeren kognitiven Aufwand betreiben, wie die Dreierkombination, die beiden grünen Wörter und der eingefärbte Teil der Abbildung, dann zuzuordnen ist, und dies wäre dann nicht mehr selbsterklärend.

4 Diskussion

In diesen Abschnitt wird ein endgültiges Fazit gezogen und resümiert, was gut und schlecht bei der Masterarbeit verlaufen ist. Danach werden im Ausblick Tipps zum Veröffentlichen und Vorschläge für Verbesserungen gegeben.

4.1 Fazit

Die formale Begriffsanalyse eignet sich hervorragend zur Bestimmung von Insekten. Als Beispiel dafür wird die Bestimmung des *Lucanus cervus* aus der Projektarbeit[Bet19] erneut aufgegriffen.

4.1.1 Hirschkäfer-Bestimmung

Dort wurde als Beispiel der Hirschkäfer (*Lucanus cervus*) bestimmt. Er wurde ausgewählt, weil er zu den bekanntesten Insekten Deutschlands zählt und eindeutige Merkmale des Käfers auch die Bestimmung vereinfachen.

Die Systematik des Hirschkäfers:

Klasse:	Insekten (Insecta)
Ordnung:	Käfer (Coleoptera)
Unterordnung:	Polyphaga
Familie:	Schröter (Lucanidae)
Gattung:	Lucanus
Art:	Lucanus cervus Hirschkäfer

Die Merkmale aus dem Buch *Bestimmung wirbelloser Tiere*[Bä08], die der *Lucanus cervus* erfüllt:

1. Flügel 1. oder 2. Paar $\geq \frac{3}{4}$ Hinterleibslänge
2. Voll entwickelte Flügel 1. Paar
3. Reduziertes 2. Flügelpaar schuppenförmig
4. Vd – Flügel ledrig hart
5. Habitat Land
6. ohne Leuchtvermögen

7. Flügeldeckel ohne Stacheln
8. ohne oder mit breiten Rüssel
9. Fühler nicht deutlich gekniet, mit oder ohne Keule
10. Fühler mit einseitig beblätterter Keule
11. Fühler gekniet, Basalglied sehr lang
12. Größe 8-75mm
13. Fühlerfächer 4-6 gliedrig
14. Färbung nicht metallisch
15. Mandibel geweihartig

Die Merkmale sind dabei in der Reihenfolge des Bestimmungsbuches aufgelistet. Der Nutzer muss also fünfzehn Fragen beantworten, bevor er zu der Seite gelangt, auf der sich der *Lucanus cervus* befindet. Zu sehen ist dies Vorgehen in Abbildung 41, wo zur Veranschaulichung der Weg von der ersten Seite über die Insektenarten des Buches bis hin zu der Seite der Art des Hirschkäfers gezeigt wird. Die rote Einfärbung zeigt den Weg, den der Nutzer zurücklegen muss, um zum Hirschkäfer zu gelangen. Dabei ist die Abbildung aus fünf Seiten des Bestimmungsbuches zusammengefügt.

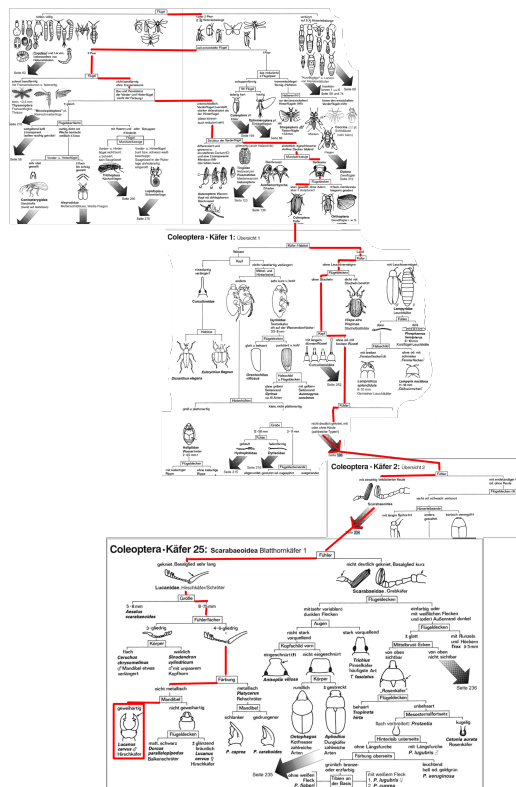




Abbildung 41: Bestimmung des Hirschkäfers

Bei dem Online-Bestimmungsschlüssel hingegen reicht die Eingabe des Wortes „geweihartig“. Wird das Wort „geweihartig“ auf der Webseite Bestimmung mit Eingabe eingegeben, zu sehen ist dies in der Abbildung 42 auf der linken Seite, führt dies direkt zum *Lucanus cervus*. Dies ist auf der Abbildung auf der rechten Seite zu sehen.




Lucanus cervus,

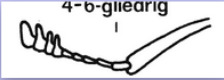
Flügel-1 oder 2 Paar größer gleich dreiviertel Hinterleibslänge
 Flügeldecke ohne Stacheln
 Fühler nicht deutlich gekniet, mit oder ohne Keule
 Habitat Land
 Kopf ohne, oder mit breiten, Rüssel
 Käfer ohne Leuchtvermögen
 Reduziertes 2. Flügelpaar schuppenförmig
 Vorderflügel ledrig hart
 voll entwickelte Flügel 1 Paar
 Färbung nicht metallisch
 Fühler gekniet, Basalglied sehr lang



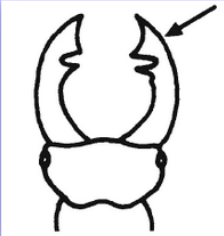
Fühler mit einseitig beblätterter Keule



Fühlerfächer 4-6 gliedrig
4-gliedrig



Große 8-75 mm
Mandibel Geweihartig



Männchen 35-80 mm

Abbildung 42: Beispiel mit der Eingabe geweihartig

Diese Bestimmung funktioniert aber leider nur so schnell, wenn der Nutzer Vorwissen über die Insekten hat. Sollte er allerdings kein Vorwissen haben bei der Bestimmung des *Lucanus cervus*, klickt sich der Nutzer durch die Bestimmung ohne Eingabe, welche sich hinter dem Menüpunkt „Bestimmung“ befindet. Dort werden weniger Abfragen als im Buch gebraucht, da mit der Technik der formalen Begriffsanalyse hier Abfragen zu Begriffen zusammengefasst worden sind und so mehrere Abfragen zur gleichen Zeit beantwortet werden können.

Beispielhaft eine Abfragereihenfolge zur Bestimmung des *Lucanus cervus*:

1. Flügel-1 oder 2 Paar größer gleich dreiviertel Hinterleibslänge
Flügeldecke ohne Stacheln
Habitat Land
Kopf ohne, oder mit breiten, Rüssel
Käfer ohne Leuchtvermögen
Reduziertes 2. Flügelpaar schuppenförmig
Vorderflügel ledrig hart
voll entwickelte Flügel 1 Paar
2. Fühler nicht deutlich gekniet, mit oder ohne Keule
3. Fühler mit einseitig beblätterter Keule
4. Färbung nicht metallisch
Fühler gekniet, Basalglied sehr lang
Fühlerfächer 4-6 gliedrig
Größe 8-75 mm
Mandibel geweihartig
Männchen 35-80 mm

Wie in der Aufzählung zu sehen, sind es nun nur noch 4 Abfragen, die in diesem Beispiel zum *Lucanus cervus* führen. Da der Verband mehrere Wege zu einer Insektenart bereit hält, könnte ein anderer Weg auch länger oder kürzer sein. Mit den 4 Abfragen ist der Online-Bestimmungsschlüssel von der Schnelligkeit der Bestimmung her den 15 Abfragen des Bestimmungsbuches weit überlegen. Im schlimmsten Fall ist die Anzahl der Abfragen des Online-Bestimmungsschlüssels aber gleich der Anzahl der Abfragen des Bestimmungsbuches. Dies bedeutet, dass die Bestimmung mit dem Online-Bestimmungsschlüssel immer mindestens genauso schnell ist, wie die Bestimmung mit Bestimmungsbuch. Hierbei sind die physischen Abläufe, wie das Umblättern und Suchen der Seite, auf der die Bestimmung weiter geht, noch außer Betracht gelassen. Diese würden die Bestimmung mittels eines Bestimmungsbuches noch verlangsamen und noch die Gefahr beherbergen, die Insektenart falsch zu bestimmen, da zum Beispiel die Bestimmung auf der falschen Seite weiter geführt werden könnte.

4.2 Ausblick

In Kapitel Veröffentlichung werden Vorschläge für das Veröffentlichende, das sogenannte *deployment*, des Online-Bestimmungsschlüssels gegeben. In dem Abschnitt

Hinweise für Entwickler werden Benutzung und Entwicklungstipps gegeben. In dem Abschnitt Vorschläge zur Weiterentwicklung werden Erweiterungen vorgeschlagen, die aus Zeitgründen nicht implementiert wurden, aber die den Online-Bestimmungsschlüssel verbessern würden.

4.2.1 Veröffentlichung

Für die Veröffentlichung des Online-Bestimmungsschlüssels eignen sich besonders gut DOCKER-Container[doc19a]. Vorteile sind neben dem flexiblen Einsatz und der Skalierbarkeit vor allem die schnelle Bereitstellung, da das Installieren und Konfigurieren des Servers und des darauf laufendem DJANGO erspart wird. Da es für alle LINUX basierten Server, aber auch für die neusten WINDOWS-Server, DOCKER-Unterstützung gibt, bietet sich eine Installation von DOCKER auf dem Server an. Anschließend kann dann ein sogenanntes Image[doc19b] von DJANGO installiert werden[doc19c]. Dies enthält alle nötigen Dateien, um ein DJANGO-Projekt darauf laufen zu lassen. Momentan ist die Projektstruktur noch so, dass alles in einem Container laufen könnte, angedacht ist aber, dass die Datenbank in einem Container läuft, die Anwendung für die Nutzer in einem anderem Container und die Eingabefunktionen wieder in einem anderen. Den Online-Bestimmungsschlüssel auf drei Container aufzuteilen hat die Vorteile, dass, falls es eine SQL-Injektion oder einen ähnlichen Angriff geben sollte, die Datenbank dadurch geschützt wäre, dass der Angreifer keinen Zugang zu dem Container mit der Datenbank bekommt. Ein weiterer Vorteil ist, dass die Eingabemethoden, die nur geschulte Anwender benutzen sollten, getrennt sind von dem Online-Bestimmungsschlüssel, der für alle Anwender erreichbar sein sollte. So könnte beispielsweise der Container für die Eingabefunktionen lokal auf einem Rechner laufen, oder nur im Intranet der Universität erreichbar sein. Besonders sicher wäre es, wenn der Container nur gestartet wird, wenn er wirklich gebraucht wird, also bei dem Vorhaben, eine neue Insektenart einzugeben oder eine bestehende zu bearbeiten. Ist dies nicht der Fall, dann laufen nur die beiden Container, die die Datenbank und die Bestandteile des DJANGO-Projekts zur Bestimmung enthalten. Dieses Vorgehen hat den offensichtlichen Vorteil, dass wenn die Eingabefunktionen nicht auf dem Server laufen oder noch besser gar nicht laufen, sie dann auch nicht oder aber schwerer für böswillige Aktionen missbraucht werden können.

4.2.2 Hinweise für Entwickler

Dieser Abschnitt ist unterteilt in zwei Aufgabengebiete. Zum einen ist dies die Eingabe neuer Insektenarten und zum anderen wird beschrieben, wann ein Starten

des Servers notwendig ist, und andere hilfreiche Informationen, die der Weiterentwicklung des Online-Bestimmungsschlüssels dienen sollen.

Eingabe neuer Insektenarten

Die Eingabe neuer Insektenarten in die Datenbank funktioniert über die HTML-Datei *eingabe_insekt* und die daran gekoppelte und gleich heißende Methode *eingabe_insekt* aus dem Python-Modul *view.py*. Die gerenderte HTML-Seite ist in Abbildung 43 zu sehen. Sie enthält einen Beschreibungstext, ein Eingabefeld und einen Button zum Bestätigen der Eingabe. Als Beispiel wird die Insektenart *Ips amitinus* nachfolgend neu angelegt. Der erste Schritt ist auf der Abbildung 43 im Eingabefeld zu sehen, und zwar ist es das Eingeben des wissenschaftlichen Namens der Insektenart. Anschließend wird in der Datenbank überprüft, ob es die Insektenart schon gibt. Dies geschieht mit dem wissenschaftlichen Namen der Art. Befindet sich bereits eine Art mit dem gleichen Namen in der Datenbank, wird diese Art zur Bearbeitung vorgeschlagen und in die Eingabefelder der *eingabe_insekt_attribute.html* (Abbildung 44) werden die Daten der Insektenart geschrieben. Dort könnten sie dann bei Bedarf geändert werden.

Eingabe der Insektenart



The image shows a web form with the title "Eingabe der Insektenart". Below the title is a text input field containing the text "Ips amitinus". To the right of the input field is a button labeled "Abschicken".

Abbildung 43: Eingabeseite des Namens der Insektenart

Falls die Insektenart nicht in der Datenbank ist, wird auf dem Server ein Ordner angelegt. Der Ordner heißt dann genauso wie die Insektenart. In diesem Ordner werden die Bilder der Insektenart gespeichert, welche auf der Informationsseite am Ende der Bestimmung angezeigt werden. Es wird nach dem Anlegen des Ordners ebenfalls die *eingabe_insekt_attribute.html*-Seite gerendert, aber nur mit einem Eintrag, und zwar dem wissenschaftlichen Namen der Insektenart. Neben der Eingabe der Daten können hier auch die Bilder für die Insektenart hochgeladen werden. Dies geschieht mit dem in HTML5 integrierten *Fileloader*, zu sehen in Abbildung 44 auf der rechten Seite. Dabei werden die Bilder benannt nach der Insektenart und einer bei 1 beginnenden fortlaufenden Zahl. Gibt es also zwei JPG-Bilder beim *Lucanus cervus*, heißen die Bilder: *Lucanus cervus1.jpg* und *Lucanus cervus2.jpg*.

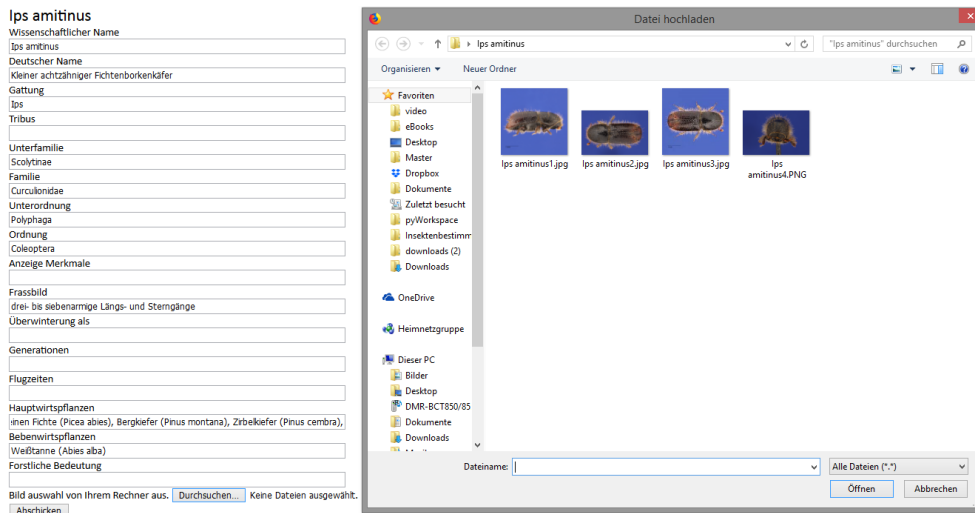


Abbildung 44: Eingabeseite der Merkmale der Insektenart

Nach dem fertigen Anlegen der Insektenart kann diese auf der HTML-Seite *auswahl_insekt* zur Verknüpfung mit den Merkmalen ausgewählt werden. Sollten dafür neue Merkmale benötigt werden, müssen diese ebenfalls neu erstellt werden, dies geschieht auf der HTML-Seite *merkmal_hinzu.html* und muss vor dem Verknüpfen geschehen. Das neue Merkmal wird dabei in ein Eingabefeld geschrieben und mit einem Radio-Button wird eingegeben, ob es ein Bild zu dem Merkmal gibt oder nicht. Falls es eins gibt, wird dies ebenfalls mit dem *Fileloader* hochgeladen. Anschließend ist das Merkmal in der Datenbank und damit auch auf der Verknüpfungsseite vorhanden. Bei den Merkmalen wird auch überprüft, ob es einzigartig ist, damit es keine zwei oder mehr Merkmale mit dem gleichen Text gibt.

Nach der Auswahl der Insektenart wird die Seite *eingabe_merkmale* geladen. Diese Seite listet alle Merkmale der Datenbank auf mit einer *checkbox* pro Merkmal. Erfüllt die Insektenart das Merkmal, hat sie einen Haken in der *checkbox*, zu sehen auf Abbildung 45. Dort hat die Insektenart *Ips amitinus* als Beispiel das Merkmal 2. Sternit ohne Dornfortsatz [71]. Die [71] verweist dabei auf eine Abbildung aus dem Buch „Handbuch zur Bestimmung der europäischen Borkenkäfer“ von Sabine Grüne[Gru79].

- Insekt *Ips amitinus*
- 1,8-2,9 mm
 - 1. Zwischenraum vor dem Absturz mit einer Reihe feiner Körnchen; Absturz schräg, eine flache Mulde bildend
 - 10-12 cm
 - 16 - 29 mm
 - 2,2 - 8,2 mm
 - 2,6 - 4 mm
 - 2. Sternit ohne Dornfortsatz [71]
 - 2. und 3. Zahn an der Basis nicht verschmolzen

Abbildung 45: Verknüpfung der Merkmale mit der Insektenart

Nach der Auswahl der Merkmale wird mit dem Button am Ende der Merkmalsliste die Auswahl an die Datenbank geschickt und dort gespeichert. Anschließend wird ein neuer aktueller Verband gebildet, der die neuen Verknüpfungen beinhaltet. Bei der Verknüpfungsseite ist noch angedacht, dass eine Einteilung der Merkmale nach Kategorien stattfindet, da sie mit den momentan 394 Merkmalen in der Datenbank schon unübersichtlich ist. Als Beispiel dafür gäbe es dann verschiedene Kategorie-Punkte wie Kopf oder Fühler und alle Merkmale, die den Kopf betreffen, befänden sich unter diesen Punkt und alle, die Fühler betreffen, unter Fühler. Momentan ist die Liste noch alphabetisch geordnet. Diese Unterteilung in Punkte hat den Vorteil, dass sie übersichtlicher und damit auch schneller beim Auffinden eines Merkmals ist.

Der Neustart des Servers

Ein Neustarten des Servers ist immer dann nötig, wenn von der in DJANGO voreingebauten Administratoren-Seite eine Änderung an der Datenbank vorgenommen wird. Da der Verband, die Autovervollständigung und andere Funktionen und Methoden aus dem Online-Bestimmungsschlüssel sonst nicht die gleichen Daten enthalten wie die Datenbank. Dies führt zu *key errors* beim Abfragen der Begriffe, die beim Laden der Bestimmungsseiten stattfinden. Da die Administratoren-Seite von DJANGO aber nur im Notfall eingesetzt werden sollte und alle anderen Interaktionen mit der Datenbank über die Eingaben-Seite der Online-Bestimmungssoftware stattfinden, wird das Neustarten des Servers nicht oft von Nöten sein.

4.2.3 Vorschläge zur Weiterentwicklung

In diesen Abschnitt werden Vorschläge gemacht, die der Verbesserung der Online-Bestimmungssoftware dienen und zur Weiterentwicklung anregen sollen.

Bilder der Bestimmung

Eine Verbesserung wäre, Bilder für die Merkmale zu erstellen, die noch keine Bilder haben. Dabei sollte aber noch darauf geachtet werden, dass nicht jedes Merkmal durch ein Bild verbessert würde. Zum Beispiel hat ein Merkmal welches eine Größe angibt, durch ein Bild keinen Mehrgewinn für die Erkennung. Ebenfalls wäre es von Vorteil, wenn die Bilder einheitlich angepasst werden. Momentan stammen die Bilder aus drei Quellen, wobei zwei Quellen sich ähneln. Dies sind die Bücher *Bestimmung wirbelloser Tiere* von Rudolf Bährmann und *Handbuch zur Bestim-*

mung der europäischen Borkenkäfer von Sabine Gruene. Die dritte Quelle sind die von der Forstzoologie in Auftrag gegebenen Bilder, diese stechen durch einen blauen Hintergrund besonders hervor.

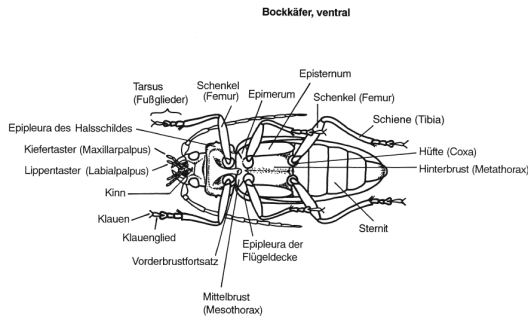
Offline Fähigkeit

Für die Bestimmung im Wald würde sich eine downloadbare Webseite oder App anbieten, da diese beiden Optionen auch ohne Internetverreichbarkeit funktionieren würden und dies in manchen Wäldern doch recht häufig ist. Das Response Design des Online-Bestimmungsschlüssels unterstützt beide Möglichkeiten. Die App wäre aber besser geeignet, da sich diese bei Internetverbindung selber aktualisieren würde. Eventuell besteht die Möglichkeit, bei der Verwendung eines Cross-Compilers neben dem HTML5-Quellcode auch den DJANGO-Quellcode weiter zu benutzen. Falls eine Entwicklung mit nativem ANDROID angestrebt wird, kann zumindest noch die Datenbank wieder verwendet werden, dabei empfiehlt es sich, eine lokale Datenbank auf dem Handy zu nehmen, die sich mit der Datenbank des Online-Bestimmungsschlüssel abgleicht. Das bietet Internet-unabhängigen Zugriff auf die Datenbank, dies dient zum Beispiel der Feldbestimmung fernab des nächsten Sendemastes oder Funkzelle.

Animationen

Die Bestimmung von Insektenarten bietet, wegen der vielen Fachbegriffe und den verschiedenen Morphologien der Arten, eine hohe Einstiegshürde. Dem Einsteiger dies zu erleichtern, dient der Versuch der interaktiven Infografik aus dem Kapitel 3.6. Dabei wurde versucht, die Abbildung möglichst einfach und für viele Insektenarten passend zu gestalten. Eine weitere interaktive Infografik für jede Ordnung wäre ein Verbesserung. Dies ist auch in den Bestimmungsbuch *Bestimmung wirbelloser Tiere* von Rudolf Bährmann so (Abbildung 46). Dort werden die Morphologien der Ordnungen vor der Bestimmung auch erst vorgestellt. Dies ist in Abbildung 46 mit den Coleoptera links und den Diptera rechts zu sehen.

Coleoptera · Käfer



Diptera · Zweiflügler

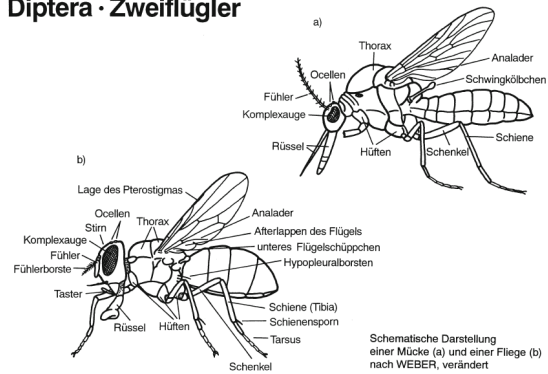


Abbildung 46: Vergleich von ähnlichen Insekten [Bä08]

Eine Überlegung ist auch, die Merkmale der Art auf einem Foto zu animieren. Dabei könnte beispielsweise mit zwei Bildern gearbeitet werden. Das eine ist das Foto der Insektenart und das andere ein ebenso großes Bild mit einer Signalfarbe wie zum Beispiel rot. Es würden dann auf dem unbearbeitetem Foto Bereiche festgelegt werden, die per Aktion, zum Beispiel einen Hovereffekt der Maus, transparent werden. Die beiden Bilder werden dann übereinander gelegt, so dass das Foto der Art über dem Bild mit der Signalfarbe ist. Dann wird bei der Aktion des Nutzers der festgelegte Bereich des oberen Bildes transparent und die Signalfarbe des unteren Bilds füllt den Bereich aus. Als Beispiel wird der Mauszeiger über die Stelle auf dem Foto bewegt wo die Augen sind, und es sehen die Augen daraufhin rot aus, da durch die Transparents des Bereiches es so aussieht, als wären die Augen rot hervorgehoben. Werden die Bereiche mit D3 und SVG erstellt, ist es auch möglich, sie mit Buttons oder anderen HTML Elementen zu verbinden, so wie bei der interaktiven Infografik aus dem Kapitel 3.6. Dann könnte auch beispielsweise der Mauszeiger über das Wort *Ozellen* bewegt werden und die Ozellen leuchten auf dem Foto der Insektenart rot auf. Dies bietet sich besonders bei den Großaufnahmen der Insektenarten an, so wie sie auf der Informationsseite der Insektenart des Online-Bestimmungsschlüssels vorhanden sind.

Literatur

- [ame19] amentsoc.org. amentsoc. <https://www.amentsoc.org/>, 04.10.2019.
- [Bä08] Bährmann, Rudolf. Bestimmung wirbelloser Tiere – Bildtafeln für zoologische Bestimmungsübungen und Exkursionen. Spektrum Akademischer Verlag, 2008.
- [Ban18] Bank, Sebastian. <https://github.com/xflr6/concepts>, Mai 2018.
- [Bas19] Basse, Marvin. Insekten bestimmen: Die besten Apps zum Erkennen von Insekten. <https://www.giga.de/extra/apps/specials/insekten-bestimmen-die-besten-apps-zum-erkennen-von-insekten/>, 04.10.2019.
- [Bet19] Bettin, Daniel Jerome. Projektarbeit Entwurf eines flexiblen Online-Bestimmungsschlüssels für Forstinsekten (auf der Grundlage der formalen Begriffsanalyse). Georg August Universität Göttingen, 2019.
- [Bos19] Boston, Mike. D3. <https://github.com/d3/d3/releases>, 19.03.2019.
- [Cla19] Clark, Alex. PIL. <https://pillow.readthedocs.io/en/stable/>, 19.09.2019.
- [dig19] digital-nature. digital-nature. <https://www.digital-nature.de/>, 04.10.2019.
- [Dja18] Django Girls. Django Girls Tutorial. <https://tutorial.djangogirls.org/de/>, 10.12.2018.
- [dja19a] djangoproject. csrf. <https://docs.djangoproject.com/en/2.2/ref/csrf/>, 21.09.2019.
- [dja19b] djangoproject.com. URL dispatcher. <https://docs.djangoproject.com/en/2.2/topics/http/urls/>, 18.09.2019.
- [doc19a] docker. docker. <https://www.docker.com/>, 01.10.2019.
- [doc19b] docker django. docker django. https://hub.docker.com/_/django, 01.10.2019.
- [doc19c] docker doc. docker doc. <https://docs.docker.com/compose/django/>, 01.10.2019.
- [dpa19] dpa. Mit der App den Insekten auf der Spur. <https://www.hna.de/netzwelt/mit-app-insekten-auf-spur-zr-9784010.html>, 04.10.2019.

- [Ernge] Ernesti, Johannes und Kaiser, Peter. Python 3. Rheinwerk Computing, 2017 (5. Auflage).
- [Gan96] Ganter, Bernhard und Wille, Rudolf. Formale Begriffsanalyse: Mathematische Grundlagen (German Edition). Springer, 1996.
- [Goo18] Google. Google Docs. <https://docs.google.com>, 10.12.2018.
- [Gru79] Gruene, Sabine. Handbuch zur Bestimmung der europäischen Borkenkäfer. M. und H. Schaper, 1979.
- [ins] insektenbox. insektenbox. <http://www.insektenbox.de/index.html>.
- [Jaw19] Jaworski, Piotr. SVG Insekt. https://de.wikipedia.org/wiki/Datei:Insect_anatomy_diagram.svg, 09.09.2019.
- [ker19] kerbtier.de. kerbtier. <https://www.kerbtier.de/>, 04.10.2019.
- [Kle17] Klein, Bernd. Einführung in Python 3. Carl Hanser Verlag, 2017.
- [Lom19] Lompe, Arved. coleonet. <http://coletonet.de>, 04.10.2019.
- [Map18] Map of Life. Map of Life. <https://play.google.com/store/apps/details?id=org.mol.android>, 15.11.2018.
- [McG00] McGavin, George. Insects, spiders, and other terrestrial arthropods. Dorling Kindersley, 2000.
- [NAB19a] NABU Naturschutzbund Deutschland e. V. Beim Insektensommer zählen wir, was zählt. <https://www.nabu.de/tiere-und-pflanzen/aktionen-und-projekte/insektensommer/index.html>, 04.10.2019.
- [NAB19b] NABU Naturschutzbund Deutschland e. V. NABU Insektenwelt. https://play.google.com/store/apps/details?id=com.sunbirdimages.nabu_insects, 04.10.2019.
- [nat18] Offene Naturführer. https://offene-naturfuehrer.de/web/Übersicht_Insekten, 2018.
- [Nenge] Nentwig, Wolfgang und Brandl, Roland und Bacher, Sven . Ökologie kompakt. Spektrum Akademischer Verlag Heidelberg, 2011(3. Auflage).
- [Pet19] Peters, Marcel. Käfer identifizieren - die besten Apps. https://praxistipps.chip.de/kaefer-identifizieren-die-besten-apps_93459, 04.10.2019.
- [Rö97] Röder, Hubert. Strukturanalyse in der Forstökologie. Ludwig-Maximilians-Universität München, 1997.

- [sar18] saragozza. Nature Free - Europa. https://play.google.com/store/apps/details?id=ch.saragozza.nature&hl=de_AT, 15.11.2018.
- [Sch18] Schwägerl, Christian. Was wir über das Insektensterben wissen - und was nicht. <https://www.spektrum.de/wissen/es-gibt-wenig-daten-aber-das-insektensterben-ist-eindeutig\discretionary{-}{-}{-}besorgnis-erregend/1548199>, März 2018.
- [sel19] selfhtml.org. HTTP/Anfragemethoden. <https://wiki.selfhtml.org/wiki/HTTP/Anfragemethoden>, 23.09.2019.
- [Stu05] Stumme, Gerd. Formale Begriffsanalyse. https://www.kde.cs.uni-kassel.de/lehre/ss2005/formale_begriffsanalyse, 2005.
- [TIO19] TIOBE Index for March 2019. <https://www.tiobe.com/tiobe-index/>, 2019.
- [Uni] Universität-Hamburg. Partielle Ordnungen. [https://www.informatik.uni-hamburg.de/TGI/lehre/v1/WS1011/FGI2/sec/FGI2_kap2\(partOrd+zeitstempel\)%20wh.pdf](https://www.informatik.uni-hamburg.de/TGI/lehre/v1/WS1011/FGI2/sec/FGI2_kap2(partOrd+zeitstempel)%20wh.pdf).
- [w3s19] w3schools. How TO - Slideshow. https://www.w3schools.com/howto/howto_js_slideshow.asp, 18.09.2019.
- [Wik19a] Wikipedia. Borkenkäfer. <https://de.wikipedia.org/wiki/Borkenk%C3%A4fer>, 01.10.2019.
- [wik19b] wiki.selfhtml.org. URL-Parameter. <https://wiki.selfhtml.org/wiki/URL-Parameter>, 18.09.2019.

Abbildungsverzeichnis

1	Die Startseite der interaktiven Bestimmung	3
2	Die Ergebnisseite der interaktiven Bestimmung	4
3	Die Auswahl bei Kerbtier.de	5
4	Interaktive Erklärung bei Kerbtier.de	5
5	Interaktive Bestimmung bei Coleonet.de	6
6	Interaktive Bestimmung bei amentsoc	7
7	Übersicht der NABU Insekten	8
8	Vergleich von ähnlichen Insekten	9
9	Vergleich von ähnlichen Insekten	10
10	Auswahl von Merkmalen von Insekten	11
11	Definition des formalen Kontext[Gan96]	13
12	Definition des formalen Begriffs [Gan96]	14
13	Definition der partiellen Ordnung[Uni]	15
14	Beispiel anhand drei Coleopteras	15
15	Begriffe, aus der Tabelle 3 gebildet	16
16	Beispiel anhand drei Coleoptera	17
17	Befehl zum Erstellen der Projektstruktur	20
18	Befehl zum Starten des Servers	21
19	Befehle für die Datenbank	21
20	Media Queries Beispiel	22
21	Beispieleingabe Googles Doc	23
22	Die HTML-Datei der Bestimmung	26
23	Bestimmungseite der Insektenarten	27
24	Der DJANGO-Code der Bestimmung	29

25	Datenbank Klasse der Insekten	30
26	Datenbank Klasse der Merkmale	31
27	Einfaches Beispiel eines HTML Templates	32
28	Einfaches Beispiel einer View-Methode	33
29	urls.py	33
30	Die Auswahlseite der Insektenarten	34
31	Erstellen des Formalen Kontext	35
32	Aufruf und Zuweisungen der Methode	35
33	Die Informationsseite der Insektenart	36
34	Übergabe Parameter per GET	37
35	Die Methode insect	37
36	Die Methode upload_image	38
37	Ein Django-Template Filter	39
38	Aufruf eines Django-Template Filters	40
39	Interaktive Insektenmorphologie Auswahl Mandibel	41
40	Interaktive Insektenmorphologie Auswahl Thorax	41
41	Bestimmung des Hirschkäfers	44
42	Beispiel mit der Eingabe geweihartig	45
43	Eingabeseite des Namens der Insektenart	48
44	Eingabeseite der Merkmale der Insektenart	49
45	Verknüpfung der Merkmale mit der Insektenart	49
46	Vergleich von ähnlichen Insekten [Bä08]	52

Tabellenverzeichnis

1	Auflistung der Unterschiede bei Webseiten	12
2	Auflistung der Unterschiede bei Apps	12
3	Kreuztabelle aus „formale Begriffsanalyse Mathematische Grundlagen“ [Gan96]	14