Research and implementation of multi-dataset training for image classification with discrepant taxonomies

A master thesis in the field of computer science

by

Björn Buschkämper

1st supervisor: Dr. Petra Bevandic

2nd supervisor: M.Sc. Riza Velioglu

Submitted in the research group "HammerLab"
of the faculty of technology for the degree of
Master of Science
at
Universität Bielefeld

May 3, 2025

Abstract

Scientific documents often use LaTeX for typesetting. While numerous packages and templates exist, it makes sense to create a new one. Just because.

Contents

1 Introduction 1

1.1 Why? . . . 1

1.2 How? . . . 1

1.3 Making this template *yours* . . . 1

1.4 Features . . . 2

1.4.1 Typesetting mathematics . . . 2

1.4.2 Typesetting text . . . 3

1.5 Changing things . . . 4

2 Methodology 5

2.1 Synthetic Taxonomy Generation . . . 5

2.1.1 Generating Domains . . . 6

2.1.2 Cross-Domain Relationship Modeling . . . 8

Acronyms 9

Glossary 11

Bibliography 13

I Introduction

In which the reasons for creating this package are laid bare for the whole world to see and we encounter some usage guidelines.

This package contains a minimal, modern template for writing your thesis. While originally meant to be used for a Ph. D. thesis, you can equally well use it for your honour thesis, bachelor thesis, and so on—some adjustments may be necessary, though.

1.1 Why?

I was not satisfied with the available templates for LaTeX and wanted to heed the style advice given by people such as Robert Bringhurst or Edward R. Tufte. While there *are* some packages out there that attempt to emulate these styles, I found them to be either too bloated, too playful, or too constraining. This template attempts to produce a beautiful look without having to resort to any sort of hacks. I hope you like it.

1.2 How?

The package tries to be easy to use. If you are satisfied with the default settings, just add

```
\documentclass{mimosis}
```

at the beginning of your document. This is sufficient to use the class. It is possible to build your document using either LaTeX|, XeLaTeX, or LuaLaTeX. I personally prefer one of the latter two because they make it easier to select proper fonts.

1.3 Making this template *yours*

Prior to using this template, the first thing you want to do is probably a little bit of customisation. You can achieve quick changes in look and feel by picking your own fonts. With the `fontspec` package loaded and XeLaTeXor LuaLaTeXas your compiler, this is pretty simple:

```
\setmainfont{Your main font}
\setsansfont{Your sans-serif font}
\setmonofont{Your monospaced font}
```

Make sure to select nice combinations of that are pleasing to *your* eyes—this is your document and it should reflect your own style. Make sure to specify font names as they are provided by your system. For instance, you might want to use the following combination:

```
\setmainfont{Libre Baskerville}
\setsansfont[Scale=MatchLowercase]{IBM Plex Sans}
\setmonofont[Scale=MatchLowercase]{IBM Plex Mono}
```

If these fonts exist on your system, your normal text will look a little bit different from the other font used in this example PDF, while your sans-serif font will pair nicely with your `monospaced font`. You can also remove the `Scale` directive, but I find that most fonts pair better if they are adjusted in size a little bit. Experiment with it until you finds a combination that you enjoy.

XeLaTeXand LuaLaTeXalso offer you a way to change the font that is used for mathematical equations. If installed, the `garamond-math` package permits you to choose from different stylistic sets that slightly change how certain mathematical symbols look. For instance, the following command changes 'Fraktur' symbols:

```
\setmathfont{Garamond-Math.otf}[StylisticSet={6}]
```

1.4 Features

The template automatically imports numerous convenience packages that aid in your typesetting process. Table 1.1 lists the most important ones. Let's briefly discuss some examples below. Please refer to the source code for more demonstrations.

1.4.1 Typesetting mathematics

This template uses `amsmath` and `amssymb`, which are the de-facto standard for typesetting mathematics. Use numbered equations using the `equation` environment. If you want to show multiple equations and align them, use the `align` environment:

$$V := \{1, 2, \dots\} \tag{1.1}$$

$$E := \big\{(u, v) \mid \operatorname{dist}\big(p_u, p_v\big) \leq \epsilon\big\} \tag{1.2}$$

Package	Purpose
amsmath	Basic mathematical typography
amsthm	Basic mathematical environments for proofs etc.
babel	Language settings
booktabs	Typographically light rules for tables
bookmarks	Bookmarks in the resulting PDF
csquotes	Language-specific quotation marks
dsfont	Double-stroke font for mathematical concepts
graphicx	Graphics
hyperref	Hyperlinks
multirow	Permits table content to span multiple rows or columns
paralist	Paragraph ('in-line') lists and compact enumerations
scrlayer-scrpage	Page headings
setspace	Line spacing
siunitx	Proper typesetting of units
subcaption	Proper sub-captions for figures

Table 1.1: A list of the most relevant packages required (and automatically imported) by this template.

Define new mathematical operators using `\DeclareMathOperator`. Some operators are already pre-defined by the template, such as the distance between two objects. Please see the template for some examples. Moreover, this template contains a correct differential operator. Use `\diff` to typeset the differential of integrals:

$$f(u) := \int_{v \in \mathbb{D}} \operatorname{dist}(u, v) \, \mathrm{d}v \tag{1.3}$$

You can see that, as a courtesy towards most mathematicians, this template gives you the possibility to refer to the real numbers $\mathbb{R}$ and the domain $\mathbb{D}$ of some function. Take a look at the source for more examples. By the way, the template comes with spacing fixes for the automated placement of brackets.

1.4.2 Typesetting text

Along with the standard environments, this template offers `paralist` for lists within paragraphs. Here's a quick example: The American constitution speaks, among others, of (i) life (ii) liberty (iii) the pursuit of happiness. These should be added in equal measure to your own conduct. To typeset units correctly, use the `siunitx` package. For example, you might want to restrict your daily intake of liberty to 750 mg.

Likewise, as a small pet peeve of mine, I offer specific operators for *ordinals*. Use `\th` to typeset things like July 4th correctly. Or, if you are referring to the 2nd edition of a book, please use `\nd`. Likewise, if you came in 3rd in a marathon, use `\rd`. This is my 1st rule.

If you want to write a text in German and use German hyphenation rules, set the language of your text to german using `\selectlanguage{ngerman}`, or add

```
\PassOptionsToPackage{spanish}{babel}
```

before the `\documentclass` command to load a specific language. The languages `ngerman`, `french`, and `english` are loaded by default, with `english` being selected.

Quotation marks can be typeset using the `\enquote{...}` command from the `csquotes` package, which is preloaded by `latex-mimosis`. Depending on the currently selected language, quotes will look like “this”, „this“, or « this ». One must never use ”ASCII” quotation marks or even 'apostrophe' symbols.

1.5 Changing things

Since this class heavily relies on the `scrbook` class, you can use *their* styling commands in order to change the look of things. For example, if you want to change the text in sections to **bold** you can just use

```
\setkomafont{sectioning}{\normalfont\bfseries}
```

at the end of the document preamble—you don't have to modify the class file for this. Please consult the source code for more information.

2 Methodology

In which the reasons for creating this package are laid bare for the whole world to see and we encounter some usage guidelines.

2.1 Synthetic Taxonomy Generation

To adequately evaluate the performance of our taxonomy generation methods, we need a suitable ground truth to compare against. While there are many real-world datasets available for image classification, there are few that provide clear inter-dataset relationships:

- **ImageNet** [1, 5] is a large-scale dataset with a hierarchical structure that categorises images into trees of classes and sub-classes. The dataset is using the WordNet [2] lexical database of semantic relations to create a taxonomy of classes. However, since this dataset has a perfectly hierarchical structure, it would not suitably represent our use case of connecting multiple datasets with different class structures (where a class might have a perfect match in another dataset, or doesn't match at all with any class).
- **Open Images** [4] is a dataset of ca. 9 million images that are annoted with labels generated using Google's Cloud Vision API[1]. The labels are again based on the WordNet lexical database, resulting in a similar structure to ImageNet. A single image can have multiple labels, which makes it difficult to determine the exact class of an image (required for our evaluation). Additionally, the labels were automatically generated and only the validation and test sets were manually verified, which makes it unsuitable as a ground truth for us.
- **iNaturalist** [3] is a highly specialised dataset of plant and animal species, with a hierarchical structure of fine-grained classes. While it contains a manually created, complex taxonomy, it is highly specific to the domain of biology which deviates from our goal of creating a general-purpose ground truth.

To overcome the limitations of finding human-annoated datasets with verified inter-dataset relationships, we instead propose a method for generating synthetic datasets with a controlled

[1] https://cloud.google.com/vision

taxonomy structure: We define a set of atomic concepts that can be used to define classes, and then generate a set of classes by randomly sampling from the atomic concepts. The resulting classes are disjoint (i.e. no class shares any atomic concepts) and form a single domain. We can repeat this process with the same set of atomic concepts to create multiple domains, and then calculate the inter-domain relationships based on the known, shared atomic concepts.

To now evaluate the performance of our taxonomy generation methods, we can use an existing dataset and use each class as a single atomic concept to generate any number of domains with images from the original dataset.

2.1.1 Generating Domains

Definition

We define:

$$\begin{aligned} \mathcal{U} &= \{1, 2, \dots, n\} \quad \text{with } n \in \mathbb{N}^* \\ \mathcal{C} &\subseteq \mathcal{U} \\ \mathcal{D}_i &= \{\mathcal{C}_1^i, \mathcal{C}_2^i, \dots, \mathcal{C}_k^i\} \quad \text{with } \forall j \neq k : \mathcal{C}_j^i \cap \mathcal{C}_k^i = \emptyset \\ \mathcal{T} &= \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_m\} \end{aligned} \tag{2.1}$$

We first build a set of atomic concepts $\mathcal{U}$, which forms a universe of concepts. A domain $\mathcal{D}_i$ now defines a set of disjoint classes $\mathcal{C}_j^i$, where each class is a subset of the atomic concepts $\mathcal{U}$. A set of domains now forms a synthetic taxonomy $\mathcal{T}$ with m domains, where each domain can have a different number of classes, with each class containing a different number of atomic concepts.

Sampling parameters

We want to generate our synthetic taxonomy $\mathcal{T}$ in a controlled manner, where we are able to specify how many domains we want to generate, how many classes each domain should have as well as how many atomic concepts should be assigned to each class. However, to make the synthetic taxonomy more realistic, we want to sample the number of classes per domain as well as the number of atomic concepts per class from a normal distribution.

Since the normal distribution is unbounded, our sampling might result in negative values or values larger than the number of atomic concepts. To avoid this we use the truncated normal distribution

$$f(x|\mu, \sigma, a, b) = \begin{cases} \frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\sigma\left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right]} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

where ϕ is the standard normal PDF and Φ is the standard normal CDF. This formulation is implemented using the SciPy `truncnorm` module[2], with appropriate standardization:

$$X \sim \text{TruncNorm}(\mu, \sigma^2, a, b)$$

Domain Generation

To now generate a domain we follow the following steps:

1. Sample the number of concepts l to use from the universe $\mathcal{U}$ of length n:

$$l \sim \lfloor \text{TruncNorm}(\mu_{\text{classes}}, \sigma^2_{\text{classes}}, 1, n) \rceil$$

2. Define a pool of available concepts P:

$$a, b, c, ... \sim \text{Uniform}(\mathcal{U}) \quad \text{without replacement}$$
$$P = \{a, b, c, ...\}$$

3. Define our domain $\mathcal{D}_i = \{\}$.

4. While $|P| \neq 0$:

   a) Sample a class size s_i from a truncated normal distribution:

$$s_i \sim \lfloor \text{TruncNorm}(\mu_{\text{class\_size}}, \sigma^2_{\text{class\_size}}, 1, |P|) \rceil$$

   b) Randomly select s_i concepts from the remaining pool to form class $\mathcal{C}^i_j$:

$$c_1, c_2, ..., c_{s_i} \sim \text{Uniform}(P) \quad \text{without replacement}$$
$$\mathcal{C}^i_j = \{c_1, c_2, ..., c_{s_i}\}$$

   c) Remove these concepts from the pool of available concepts:

$$P = P \setminus \mathcal{C}^i_j$$

   d) Add the class to the domain:

$$\mathcal{D}_i = \mathcal{D}_i \cup \{\mathcal{C}^i_j\}$$

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.truncnorm.html

2.1.2 Cross-Domain Relationship Modeling

We have now generated our synthetic taxonomy $\mathcal{T}$, but we still need to define the relationships between different domains.

Our universal taxonomy generation method is based on the assumption that the model classifiers will be able to predict related classes from different domains with a certain probability. It then uses these probabilities to build a universal taxonomy that maps domain classes to universal classes.

To simulate this behavior, we want to generate perfect synthetic probabilities that represent the relationships between classes in different domains. An example of this would be the following:

- We have two domains $\mathcal{D}_A = \{\mathcal{C}_1^A, \mathcal{C}_2^A\}$ and $\mathcal{D}_B = \{\mathcal{C}_1^B, \mathcal{C}_2^B\}$.
- The classes are defined as $\mathcal{C}_1^A = \{1, 2\}$, $\mathcal{C}_2^A = \{3, 4\}$ and $\mathcal{C}_1^B = \{1, 2, 4\}$, $\mathcal{C}_2^B = \{5, 6\}$.
- The relationship from $\mathcal{C}_1^A \rightarrow \mathcal{C}_1^B$ would have a probability of 1, since all atomic concepts in $\mathcal{C}_1^A$ are also present in $\mathcal{C}_1^B$.
- The relationship $\mathcal{C}_2^A \rightarrow \mathcal{C}_1^B$ should have a probability of 0.5, since only half of the atomic concepts in $\mathcal{C}_2^A$ are also present in $\mathcal{C}_1^B$. However, since the synthetic taxonomy should simulate the behavior of a neural network, the other half of the atomic concepts cannot be assigned to any class in $\mathcal{D}_B$ and the neural network would likely randomly choose any class in $\mathcal{D}_B$. Therefore, we evenly distribute the remaining probability mass to all classes in $\mathcal{D}_B$. This means that the relationship $\mathcal{C}_2^A \rightarrow \mathcal{C}_1^B$ would have a probability of $0.5 + \frac{0.5}{2} = 0.75$,

To formalise this, we define the following terms for any two given domains $\mathcal{D}_A$ and $\mathcal{D}_B$ with $\mathcal{D}_A \neq \mathcal{D}_B$:

$$\begin{aligned} \text{NaiveProbability}(i, j) &= \frac{|\mathcal{C}_i^A \cap \mathcal{C}_j^B|}{|\mathcal{C}_i^A|} \quad \text{with } i \in \{1, \dots, |\mathcal{D}_A|\}, j \in \{1, \dots, |\mathcal{D}_B|\} \\ P &\in \mathbb{R}^{|\mathcal{D}_A| \times |\mathcal{D}_B|} \quad \text{with } P_{i,j} \in [0, 1] \\ P_{i,j} &= \text{NaiveProbability}(i, j) + \frac{1 - \text{NaiveProbability}(i, j)}{|\mathcal{D}_B|} \end{aligned} \tag{2.2}$$

Acronyms

CDF	Cumulative Distribution Function
PDF	Probability Density Function

Glossary

LaTeX	A document preparation system
$\mathbb{R}$	The set of real numbers

Bibliography

1. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.
2. C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998. ISBN: 9780262272551. DOI: 10.7551/mitpress/7287.001.0001. URL: https://doi.org/10.7551/mitpress/7287.001.0001.
3. G. V. Horn, O. M. Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. *The iNaturalist Species Classification and Detection Dataset*. 10, 2018. DOI: 10.48550/arXiv.1707.06642. arXiv: 1707.06642[cs]. URL: http://arxiv.org/abs/1707.06642 (visited on 05/03/2025).
4. A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari. "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale". *IJCV*, 2020.
5. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge*. 30, 2015. DOI: 10.48550/arXiv.1409.0575. arXiv: 1409.0575[cs]. URL: http://arxiv.org/abs/1409.0575 (visited on 03/19/2025).