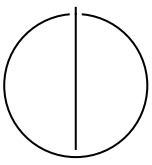
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

How do technical and scientific contributions of Foundation Models for Robotics develop over time compared to those for Language? A large-scale systematic analysis using LLMs

Tobias Geilen



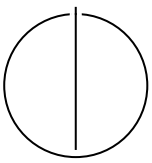
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

How do technical and scientific contributions of Foundation Models for Robotics develop over time compared to those for Language? A large-scale systematic analysis using LLMs

Master's Thesis in Information Systems

Author:	Tobias Geilen
Examiner:	Prof. Dr. Florian Matthes
Supervisor:	Sebastian Sartor, M. Sc.
	Alexandre Mercier, M. Sc
Submission Date:	October 3rd, 2025



I confirm that this master's thesis is my own work, and I have documented all sources and materials used. It is in full compliance with the TUM AI Strategy [Bra24].

Munich, October 3rd, 2025 Tobias Geilen

Acknowledgments

With the greatest gratitude, I wholeheartedly thank all those who have supported, inspired and shaped me on my journey so far.

Abstract

This thesis introduces an automated pipeline leveraging Large Language Models to conduct a large-scale, systematic literature review of Foundation Model development. The objective was to create a comprehensive overview of models and their characteristics by automating the manual paper review process. The developed tool was benchmarked against the EpochAI dataset, achieving a data extraction accuracy of $70\% - 80\%$ while highlighting the challenges posed by LLMs' unreliable structured output. The analysis of the extracted data revealed key trends, including the exponential growth of model parameters, a major shift toward Transformer-based architectures, and the increasing research contribution from large technology companies. Furthermore, a detailed cross-domain comparison with a dataset on robotics models uncovered distinct trends, such as the high degree of multimodality and the field's strong academic-led research output. This work demonstrates the feasibility of using AI for a rigorous, data-driven approach to literature reviews, offering a more holistic understanding of the Foundation Model landscape.

Contents

1. Introduction

1.1. Background & Context

Artificial Intelligence (AI) has become a primary focus of both academic and industry research laboratories. Between 2018 and 2023, the number of AI-related publications increased significantly, rising from approximately 145,000 to over 240,000. [Mas+25]

The development of foundational models, used for example, in large language models (LLMs) such as ChatGPT, has substantially enhanced AI's capabilities and broadened its range of applications. New AI research is published each week across diverse domains, including robotics, medicine, and even Earth observation. Given the rapid expansion of AI research and the overwhelming volume of publications, systematically identifying trends and comparing findings across domains has become increasingly difficult. Comprehensive summaries are, therefore, crucial for guiding future AI advancements and improving the evaluation of model performance.

AI technologies now enable the efficient summarization of vast numbers of research papers within a comparatively short time. Moreover, AI can be leveraged within the scientific process to facilitate large-scale data collection, aggregation, and analysis, thereby enhancing research efficiency and knowledge dissemination.

1.2. Problem Statement

Prior studies have analyzed the history and development of AI, examined its opportunities and risks, and extensively surveyed AI advancements to continuously abstract and summarize findings. These surveys offer a broad overview of AI adoption in specific application domains, structuring findings, and recent developments in the rapidly changing field of AI and foundational models. Additionally, some meta-reviews have been published to compare and provide common benchmark tests and evaluation methods for specific AI technologies across different domains.

The existing surveys are often highly specific in their scope - either focusing on a cer-

tain application domain or a certain model type. Therefore, systematically comparing the adoption of different model types across various application domains is currently methodologically complex due to the fragmented nature of existing studies.

Furthermore, most existing surveys rely on manual reviews of a limited number of publications (typically a few dozen to a few hundred), restricting their ability to capture broader trends in AI adoption. One of the most extensive summaries is provided by the EpochAI research group. Their flagship dataset currently contains information on over 900 notable models. Nevertheless, their methodology is also centered around manual paper review and information retrieval.

1.3. Objective & Research Questions

This thesis aims to bridge these gaps by leveraging AI to conduct a large-scale, systematic literature review of Foundation Model development across various application domains.

The objective is to create a comprehensive overview of models and their characteristics - like the Epoch AI dataset - by developing a software tool that automates the manual paper review and utilizes LLMs for the information retrieval process to extend the currently existing overviews:

1. Build a fully automated system to identify papers introducing Foundation Models
2. Are NLP tools able to extract relevant objective parameters from previously identified papers? What is the quality of the results from the automated solution compared to the Epoch AI dataset?
3. What is the status quo in Foundation Model development for the specific field of Robotics, and how do the models compare in their characteristics to other fields?
4. Are there relevant differences between the robotics domain and other domains (such as language)?

This research is relevant both theoretically and practically. Theoretically, it will provide insights into how well large language models can support and automate the scientific process. Practically, the findings will be valuable for the AI practitioners and organizations to understand how Foundation Models are developed and adjusted for a specific application domain.

2. Conceptual Foundations & Related Work

2.1. Foundation Models

2.1.1. A Brief Introduction to Artificial Intelligence

Artificial Intelligence The term Artificial Intelligence has been used in Mathematics and Computer Science research since the Mid of the 20th century. In his "Proposal for the Dartmouth Summer Research Project on Artificial Intelligence", at the time Assistant Professor in Mathematics at Dartmouth College, John McCarthy describes his interest in researching "some aspects of the artificial intelligence problem", including "How Can a Computer be Programmed to Use a Language", "Neuron Nets" and "Self-Improvement" during the 1956 Dartmouth summer research project. [McC+06]

Since then, the research on Artificial Intelligence has continuously advanced, and many definitions have been introduced by researchers from different fields. In the book "Artificial Intelligence: A Modern Approach" by Russel et al., some of these definitions are categorized, highlighting the broadness of the topic and the various research perspectives (see Table 2.1) [RND10].

More recently, governments and other institutions have also proposed their definitions of Artificial Intelligence, showcasing the current perspective on the topic. In their 2020 National Artificial Intelligence Initiative Act, the U.S. government described AI as "a machine-based system that can, for a given set of human-defined objectives, make predictions, recommendations or decisions influencing real or virtual environments." [Rep20]

The European Union defined an AI system in their EU Artificial Intelligence Act as "a machine-based system that is designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment, and that, for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments". [Eur]

	Humanly	**Rationally**
Thinking	"The exciting new effort to make computers think ... machines with minds, in the full literal sense." (1985) [Hau89] "[The automation of] activities, that we associate with human thinking, activities such as decision-making, problem solving, learning ...." (1978) [Bel78]	"The study of mental faculties through the use of computational models." (1987) [Cha+14] "The study of the computations that make it possible to perceive, reason, and act." (1992) [Win93]
Acting	"The art of creating machines that perform functions that require intelligence when performed by people." (1990) [Kur92] "The study of how to make computers do things at which, at the moment, people are better." (1991) [Ric91]	"Computational Intelligence is the study of the design of intelligent agents." (1998) [PMG98] "AI ... is concerned with intelligent behavior in artifacts." (1998) [Nil98]

Table 2.1.: Definitions of Artificial Intelligence categorized by Russel et al.

It can be seen that the definitions are fairly vague, reflecting the varieties of technologies falling into this category.

Machine Learning Building on McCarthy's early vision on AI and the need for self-improvement, researchers in the 1990s focused on developing systems capable of improving their performance from experience and established *Machine Learning (ML)* as a subdomain of AI research. This field investigates algorithms allowing machines to learn from data without being explicitly programmed. [ISO] These algorithms marked a shift from the way AI systems were previously designed, as rather than specifying how to solve a task, a learning algorithm would induce it based on provided data, and the how emerges by itself during the learning process. [Bom+22]

The learning algorithms used can be categorized into five strategies, which differ in the input data requirements and the way feedback or guidance is provided during the learning process: [Ber20]

1. **Supervised Learning**
   The dataset being used during training has been pre-labeled (e.g. a picture showing a cat was marked as showing a cat and not a dog) and annotated by users or other systems to allow the algorithm to see how accurate its performance is (e.g. how many cat pictures were correctly identified as containing cats and how many pictures showing a dog were falsely identified as showing a dog).

   Over many iterations, the algorithms then optimize their parameters to maximize these performance metrics and minimize their loss function.

2. **Semi-supervised Learning**
   The training data contains a small portion of labeled examples and a large portion of unlabeled examples. Some methods generate pseudo-labels for the unlabeled data and then apply supervised learning to the combined dataset. Other approaches, such as consistency regularization or graph-based label propagation, use the unlabeled data without explicitly assigning labels.

3. **Unsupervised Learning**
   The dataset being used is entirely unlabeled and an algorithm seeks to find an underlying structure - such as clusters, patterns or latent features - without any external guidance or user input.

4. **Self-Supervised Learning**
   Self-Supervised Learning (SSL) is a variant of unsupervised learning where the system automatically generates supervisory signals from the input data itself. For instance, in natural language processing, a model may mask certain words in a sentence and learn to predict them from the surrounding context. Similarly, in computer vision, a model can predict missing parts of an image or the rotation angle of an image patch.

   This strategy enables the model to leverage vast amounts of unlabeled data while learning meaningful representations.

5. **Reinforcement Learning**
   Reinforcement Learning differs from the other learning approaches by usually not relying on a dataset. Instead, a virtual agent interacts with an environment and receives reward signals based on its actions - positive as a reward and negative as a penalty.

Over many iterations, the underlying algorithms learn how to maximize the cumulative reward and thereby also optimize the agent's behavior within the environment guided by the initially defined reward function.

These categories describe overarching learning paradigms. Each can be implemented using a variety of algorithms and model architectures, ranging from decision trees and support vector machines to probabilistic models and modern deep neural networks.

Ultimately, the goal of Machine Learning is to learn patterns from training data, which generalize well to unseen data, enabling robust performance in unknown, real-world scenarios.

Neural Networks Among the diverse algorithms and architectures used in Machine Learning, Artificial Neural Networks have emerged as a particularly powerful class of models, especially in tasks involving high-dimensional and unstructured data such as images, audio, or natural language. Inspired by the structure and functioning of biological neurons, neural networks form the foundation of modern deep learning and have enabled recent breakthroughs in AI capabilities.

From a technical perspective, an artificial neural network is an interconnected group of nodes and edges - a directed graph to be precise - inspired by a simplification of neurons in a brain. Therefore, the nodes are often referred to as neurons. Typically, the neurons are grouped into layers, with the first layer being referred to as *Input Layer* and the last as *Output Layer*. The layers in between are called *Hidden Layers*. If a network has more than one hidden layer, it is typically called *Deep Neural Network* [Bis06]

When data is processed by the neural network, signals travel along the directed edges from the input layer, through the hidden layers to the output layer. Each node receives a signal from the connected nodes in the previous layer and calculates its own output value based on a *Activation Function*, which is usually non-linear, taking all inputs into account. The calculated output value then gets forwarded to all connected nodes in the next layer along the directed edges of the graph. Importantly, the edges have individuals weights w_{ij}, which influence the input signal strength for the receiving node. The individual results of the output layer activation functions, are then interpreted as the model's prediction - for a classification tasks, this often takes the form of a probability distribution over possible output classes.

Training a neural network involves adjusting its weights w_{ij} and biases b_j so that its predictions align with the desired outputs for a given task. This is typically achieved

through *backpropagation*, an algorithm that computes the gradient of a chosen loss function with respect to each weight by applying the chain rule of calculus layer-by-layer in reverse order. These gradients are then used by an optimization algorithm to iteratively update the parameters and minimize the loss / "the error". Over many training iterations, the network learns internal representations in its hidden layers that capture relevant patterns in the data, enabling it to generalize to previously unseen inputs.

In the forward pass, each neuron computes

$$z_j = \sum_i w_{ij}\, a_i + b_j, \quad a_j = \sigma(z_j)$$

where a_i is the activation from the previous layer, w_{ij} is the weight, b_j is the bias, and σ is the activation function.

Given a target output y and prediction $\hat{y}$, a loss function $\mathcal{L}(y, \hat{y})$ quantifies the prediction error.

In backpropagation, the gradients

$$\frac{\partial \mathcal{L}}{\partial w_{ij}}$$

are computed via the chain rule, and the parameters are updated using an optimization rule such as stochastic gradient descent (SGD):

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}}$$

where η is the learning rate.

Deep Learning As previously described, deep learning refers to the the application of neural networks with many layers, enabling them to learn increasingly abstract and complex representations of data. While shallow networks can model simple patterns, deeper architectures can automatically extract hierarchical features. For example, in image recognition, early layers may detect edges, intermediate layers shapes, and later layers entire objects.
[LBH15]

The feasibility of deep learning was unlocked in the early 2010s through three factors:

(i) the availability of massive, labeled datasets

(ii) hardware accelerators, especially GPUs, enabling efficient parallelized training

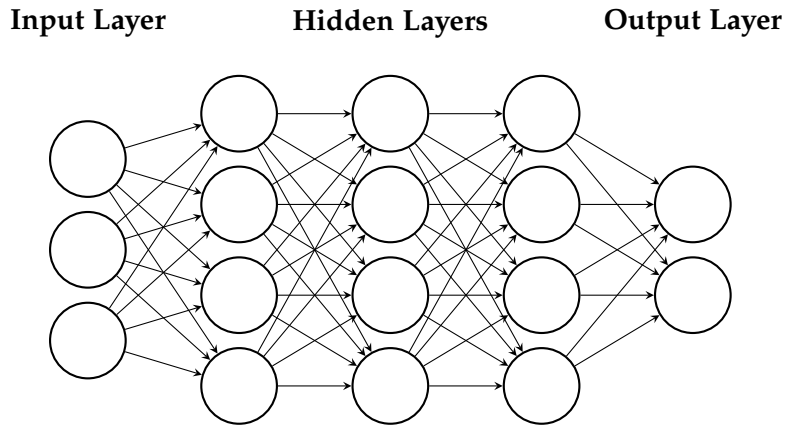


Figure 2.1.: Example of a deep neural network with 3 inputs, 3 hidden layers, and 2 outputs - e.g. a binary classifier

(iii) architectural and optimization advances such as rectified linear units (ReLU) or batch normalization, which improved convergence and mitigated overfitting.

These developments triggered rapid progress in areas such as computer vision, speech recognition and natural language processing. The same scaling principles - larger models, more data and greater compute - now also help facilitate the emergence of foundation models, which extend deep learning architecture to unprecedented sizes and capabilities. [SS18]

2.1.2. What is a Foundational Model?

Defining what is now understood as a Foundation Model is not a simple task.

The term was first used and introduced by Bommasani et al. in August 2021 in their paper titled "On the Opportunities and Risks of Foundation Models", as previously existing terms - such as pretrained model or self-supervised model - "partially captured the technical dimension, but fail to capture the significance of the paradigm shift. [...] In particular, the word *foundation* specifies the roles these models play: a foundational model is itself incomplete but serves as the common basis from which many task-specific models are built via adaptation." [Bom+22]

Since then, many other definitions have emerged, especially from regulators:

The US government defines them as "an AI model that trained on broad data; gener-

ally uses self-supervision: contains at least tens of billions of parameters; is applicable across a wide range of contexts [...]" [Hou23]

The EU AI Act defines foundation models as "AI model that is trained on broad data at scale, is designed for generality of output, and can be adapted to a wide range of distinctive tasks". [Eur]

The United Kingdom defines them as "AI technology [...] trained on vast amounts of data that can be adapted to a wide range of tasks and operations". [UK]

Interestingly, all definitions are broad in scope and emphasize the size of the data used during the training process as a defining characteristic, in addition to applicability across multiple domains.

For this thesis, this notion is followed, and, for simplicity, Foundation Models are regarded as AI models trained on vast amounts of data with applications in multiple domains.

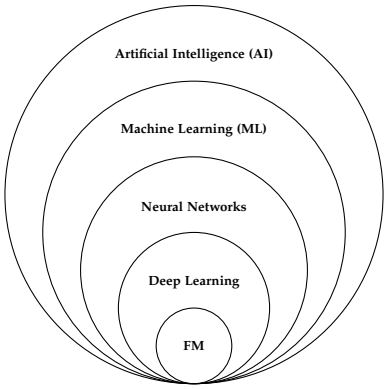


Figure 2.2.: Hierarchical relationship of AI, Machine Learning, Neural Networks, Deep Learning, and Foundation Models (FM) as nested concepts.

2.1.3. LxMs & Application Fields of Foundation Models

A term often used interchangeably with Foundation Models is Large Language Model (LLM). This, however, is not entirely accurate: while many LLMs meet the criteria of foundation models, not all foundation models are language-based, and some LLMs are

trained for highly specific domains or tasks. [Bom+22]

The domain-agnostic nature of Foundation Models implies that the technology can also perform well on non-language-based data types and domains. Recent developments have extended their application fields beyond language. Foundation Models used in specific domains can also vary in their modalities, meaning the data types they can process as input and output.

The following non-exhaustive list provides an overview of some application domains, including exemplary tasks:

- **Language**
  These models are the most well-known, as they are the underlying technology of end-user tools such as the initial release of ChatGPT-3.5, which interact in a conversational manner based on language [Ope24]. Importantly, these models are general-purpose and, therefore, capable of generating written text output usable for a wide variety of tasks specified directly by the user via prompts ranging from code generation to cooking recipes to poetry [Tou+23].

- **Computer Vision**
  The overarching goal of Computer Vision is the retrieval of information from images or videos by a computer. While in the past, algorithms relied on complex logic (e.g., to identify objects), in 2012 AlexNet became the first to leverage deep neural networks to categorize images into 1000 different classes by iteratively learning features during training, instead of hard-coding detection logic by hand [KSH12]. Recent specialized Foundation Models for Computer Vision have greatly extended capabilities, combining information retrieval, image classification, object detection, and action recognition into single models [Yua+21], defining a new era in Computer Vision [Awa+23].

- **Image & Video Generation**
  In contrast to Computer Vision, Image & Video generating models usually receive a text-based input and are able to produce an image or a video as output [Wan+25]. Downstream tasks range from simple personal video generation to morally questionable and malicious practices, including DeepFakes [KM18].

- **Speech**
  Processing speech can be categorized in three fields: audio-to-text (e.g., transcription), text-to-audio (e.g., voice generation), and audio-to-audio (e.g., real-time translation). For the first two fields, state-of-the-art Foundation Models work

with strong accuracy and generalizability [Rad+22]. The latter, however, is currently not yet as mature due to constraints in processing continuous data-streams [WXL25].

- **Medicine**
  Foundational Models demonstrate outstanding success across multiple medicine and healthcare domains. Specialized Clinical Large Language Models support doctors in evaluating symptoms, analyzing medical data, and proposing treatments with human-like performance on dedicated tests. Particularly in Medical Computer Vision models already outperform medical experts in terms of accuracy, e.g., in classifying and detecting abnormalities in X-ray images[Kha+25].

- **Biology**
  Foundational Models are showing how foundational research is being conducted. AlphaFold, a Foundation Model designed to predict the 3D-structure of proteins based on their amino acid sequence, solved this 50-year scientific challenge with near-experimental accuracy, moving the problem from open research to a practical tool. Research can now predict the structure within minutes to hours and at negligible costs, instead of having to spend months actually producing proteins and determining their structures. This democratization of access to protein structures has changed the pace and scale of biological research, enhancing drug discovery, synthetic biology, and evolutionary biology [Jum+21].

- **Earth Science**
  The ability to model and predict the behavior of complex systems has led to the development of Foundation Models trained on diverse geophysical data, including satellite imagery, ocean temperatures, CO_2-levels, and vegetation indices. These models offer the possibility to predict air quality, ocean waves, tropical cyclones, as well as high-resolution weather [Bod+25], and are being adopted to understand extreme weather and climate events [Cam+25].

- **Robotics**
  Foundational Models for Robotics are highly complex and often combine multiple skill-specific models (vision models, insertion models, grasping models, movement models, etc.) connected by another centralized controlling model for skill selection and input processing. Due to the complexity and interaction with the physical world are still less advanced than other application fields. [Jog+24] The significance of the field, however, is widely agreed upon. Leading companies and strategists see Physical AI as the next wave of AI, with the ability to revolutionize manufacturing. [NVI25]

Foundation Models are being adopted across many domains and enable more accurate, more efficient, and entirely new ways to process data and make predictions.

2.2. Scaling Laws & Performance Influencing Factors

The rapid advancements in the capabilities and the accuracy of Foundation Models are driven by multiple factors. This chapter explores these factors and their impact on model performance.

In this context, scaling laws describe empirically observed trends showing how a model's test loss—interpreted as its error—changes according to power-law relationships when key factors such as model size, dataset size, or compute are scaled.

Following early investigation by Hestness et al. [Hes+17], Kaplan et al. have famously shown, that for Transformer Language Models "performance depends strongly on scale, weakly on model shape [with scale consisting of] the number of model parameters N (excluding embeddings), the size of the dataset D, and the amount of compute C used for training" [Kap+20]. According to their research, "performance has a power-law relationship with each of the three scale factors N, D, C, when not bottlenecked by the other two. [Observing] no signs of deviation from these trends on the upper end, though performance must flatten out eventually before reaching zero loss."

These scaling laws can also be identified in other domains, such as image & video generation or mathematical problem solving [Hen+20]. Interestingly, the scaling laws' accuracy allows to use them for both forecasting the needed scale to achieve any given reducible loss, and also predicting a model's performance given its scale [Hes+17].

Figure 2.3 visualizes the individual scaling laws for the Compute, Dataset Size, and Parameters. Due to the logarithmic scale of the chart, the exponential changes appear in a near-linear line.

2.2.1. Model Architectures & Parameter Size

Despite Kaplan et al.'s findings that a model's performance "only very weakly [depends on] architectural hyperparameters such as depth vs. width" [Kap+20], it is fundamental to understand modern foundation model architecture, before discussing parameter count as influencing factor on the models performance.

In 2017, the new Transformer architecture has revolutionized AI, particularly in natural language processing, by introducing a fundamentally new way of handling

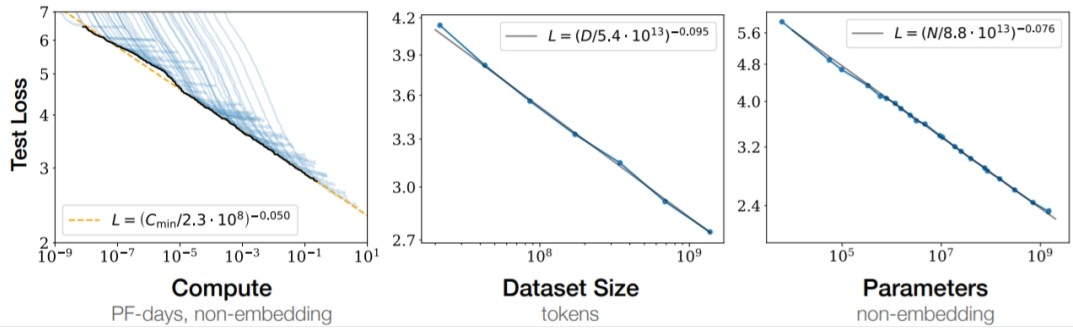


Figure 2.3.: Scaling laws: Test loss decreases exponentially as compute, dataset size, or model parameters increase (Kaplan et al.).

sequential data [Vas+23]. Thereby, four key innovations stand out, also enabling scaling:

- **Self-Attention Mechanism**
  Unlike earlier models that processed data sequentially, Transformers use self-attention to simultaneously consider all parts of the input sequence. This allows the model to dynamically weigh the importance of each word relative to others in a sentence, capturing complex dependencies and contextual relationships more effectively.

- **Parallel Processing**
  Transformers can process entire input sequences at once, rather than interpreting them step-by-step. This parallelization has drastically increased training speeds, making it viable to train on much larger datasets.

- **Scalability**
  The architecture is separated into encoder and decoder stacks, each composed of identical layers with multi-head self-attention and feed-forward neural networks connecting them. Model size can relatively easily be increased by extending this chain of attention layers and feed-forward neural networks.

- **Long-Range Context**
  Transformers overcome limitations of previous architectures, which struggled with "remembering" long sequences, by processing all positions of the input together, enabling better understanding of context across large spans of text and data.

In general, all Transformer-based models consist of the same main components: Tokenizers, Embedding layers, Transformer layers and Un-embedding layer.

Tokenizers While language models in their original form process text as input modality, their ability to learn relies on calculating derivatives of continuous functions. Hence, text input must be converted into numerical representations (tokens).

A tokenizer maps raw text into a sequence of integer token IDs. It does so using a predefined vocabulary of size V, typically consisting of characters, subword units, or short character sequences. Each token in the vocabulary is assigned a unique integer index $t \in \{0, 1, \ldots, V-1\}$.

For example, the tokenizer published by Mistral AI, specifically developed for their models, has a vocabulary size of 130,000 plus an additional 1,000 control tokens representing special instructions to the model [Mis25b].

Embedding Layer Once the text is tokenized into integer IDs, each token t is represented as a *one-hot vector*:

$$\mathbf{e}_t \in \mathbb{R}^V, \quad (\mathbf{e}_t)_i = \begin{cases} 1 & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

The embedding layer then maps this sparse representation into a dense d_{model} - dimensional vector using the embedding matrix $M \in \mathbb{R}^{V \times d_{\text{model}}}$. The embedding is computed as:

$$\text{Embed}(t) = \mathbf{e}_t^\top M,$$

which is equivalent to selecting the t-th row of M.

Transformer-based language models, unlike recurrent or convolutional architectures, do not have any inherent notion of token order. Since Transformer models and their attention mechanisms treat all tokens in the input sequence as a set rather than a sequence, additional information is required to let the model distinguish between different positions.

This is achieved through positional encodings, which assign each position p in the input sequence a unique vector $\mathbf{p}_p \in \mathbb{R}^{d_{\text{model}}}$. These vectors are added element-wise to the corresponding token embeddings before being fed into the Transformer layers:

$$\mathbf{x}_p = \text{Embed}(t_p) + \mathbf{p}_p.$$

Two main types of positional encodings are commonly used:

- **Fixed (Sinusoidal) Encoding:** Introduced by Vaswani et al. [Vas+23], these encodings use sine and cosine functions of different frequencies:

$$\mathbf{p}_{p,2i} = \sin\left(\frac{p}{10000^{2i/d_{\text{model}}}}\right), \quad \mathbf{p}_{p,2i+1} = \cos\left(\frac{p}{10000^{2i/d_{\text{model}}}}\right),$$

where i indexes the embedding dimension. This allows the model to generalize to sequence lengths longer than those seen during training.

- **Learned Positional Embeddings:** Instead of using a fixed function, the positional vectors $\mathbf{p}_p$ are learned during training, similar to token embeddings. This can improve performance when the maximum context length is fixed and known in advance.

In both cases, positional encoding injects information about token order into the model without changing the overall dimensionality, enabling the self-attention mechanism to take sequence structure into account. [Dev+19]

Transformer Layer The sequence of token embeddings, augmented with positional encodings, forms the input $\mathbf{X} \in \mathbb{R}^{n \times d_{\text{model}}}$ to the first Transformer layer, where n is the sequence length and d_{model} is the embedding dimension.

Each Transformer layer consists of two main sub-layers: a *multi-head self-attention mechanism* and a *position-wise feed-forward neural network* (FFN), each wrapped with residual connections and layer normalization [Vas+23].

In the self-attention mechanism, the input $\mathbf{X}$ is linearly projected into three different representations — queries (Q), keys (K), and values (V) — using learnable weight matrices:

$$Q = \mathbf{X}W_Q, \quad K = \mathbf{X}W_K, \quad V = \mathbf{X}W_V,$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and d_k is the dimensionality of each attention head's query/key vectors. For multi-head attention with h heads, these projections are computed independently per head, resulting in h separate sets of (W_Q, W_K, W_V). The outputs of all heads are concatenated and projected back to the model dimension using $W_O \in \mathbb{R}^{h \cdot d_k \times d_{\text{model}}}$.

Following attention, the FFN applies two fully connected layers to each token embedding independently:

$$\text{FFN}(\mathbf{x}) = \sigma(\mathbf{x}W_1 + b_1)W_2 + b_2,$$

where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ and $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$. Here, d_{ff} — typically several times larger than d_{model} — makes the FFN the dominant contributor to each layer's parameters.

For a stack of L Transformer layers, the total parameter count scales approximately as:

$$\mathcal{O}\big(L \cdot (\underbrace{4 \cdot d_{\text{model}} \cdot d_k \cdot h}_{\text{attention projections}} + \underbrace{2 \cdot d_{\text{model}} \cdot d_{\text{ff}}}_{\text{feed-forward layers}})\big),$$

plus the embedding and output layers. The first term corresponds to the Q, K, V, and output projection matrices in the attention mechanism, while the second term accounts for the two dense layers in the FFN.

In practice, d_k is often chosen as d_{model}/h. Since h typically grows much more slowly than d_{model}, the attention term scales as $\mathcal{O}(4 \cdot d_{\text{model}}^2)$.

In nearly all transformer architectures, $d_{\text{ff}} = s \cdot d_{\text{model}}$ where s is a small constant (often $s = 4$). Hence, the FFN term scales as $\mathcal{O}(8 \cdot d_{\text{model}}^2)$.

Combining the attention term with the FFN term, this means that for common configurations, the total parameter count grows approximately as:

$$\mathcal{O}(12 \cdot L \cdot d_{\text{model}}^2).$$

The transformer-based GPT-3 model by OpenAI [Bro+20]., consists of $L = 96$ layers with $d_{\text{model}} = 12288$ leading to an estimated parameter size

$$\hat{n}_{\text{model}} \approx 12 \cdot 96 \cdot 12288^2 = 173.946.175.488$$

Officially, the GPT-3 model has an official rounded parameter size, including additional parameters from the embedding of

$$n_{\text{model}} = 175.000.000.000$$

Un-embedding Layer After the final Transformer layer, the model's output is a sequence of dense vectors, one for each position, where each vector has a dimension of d_{model}. The un-embedding layer's purpose is to convert these dense vectors back into a probability distribution over the model's entire vocabulary of size V.

This is achieved by using a linear projection layer, often referred to as the output head. This layer applies a weight matrix $W_o \in \mathbb{R}^{d_{\text{model}} \times V}$ to the final output vector $\mathbf{h} \in \mathbb{R}^{d_{\text{model}}}$ from the Transformer stack:

$$\textbf{logits} = \mathbf{h}^\top W_o + \mathbf{b}_o,$$

where $\textbf{logits} \in \mathbb{R}^V$ is the unnormalized log-probability for each token in the vocabulary and $\mathbf{b}_o$ is a bias vector.

Finally, the logits are passed through a softmax function to obtain a probability distribution **P** over the vocabulary, where each element P_j represents the likelihood of the j-th token being the next predicted token:

$$\mathbf{P} = \text{softmax}(\textbf{logits}).$$

This final probability distribution is used during training to calculate the loss and during inference to select the next token in the sequence. [PW17]

While architectural details of these individual components only weakly affect performance, increasing the number of transformer layer L or d_{model} scales the parameter count n_{model}, which in turn influences performance according to scaling laws when training compute and dataset size are sufficient. Kaplan et al. have shown this, by fixing n_{model} and testing the performance of models with varying numbers of layer L and d_{model} still multiplying to n_{model}. As the trend is especially well noticeable, when the embedding parameters are excluded, they argue the embedding matrix can be made smaller without impacting performance, as also shown by Lan et al. [Lan+20].

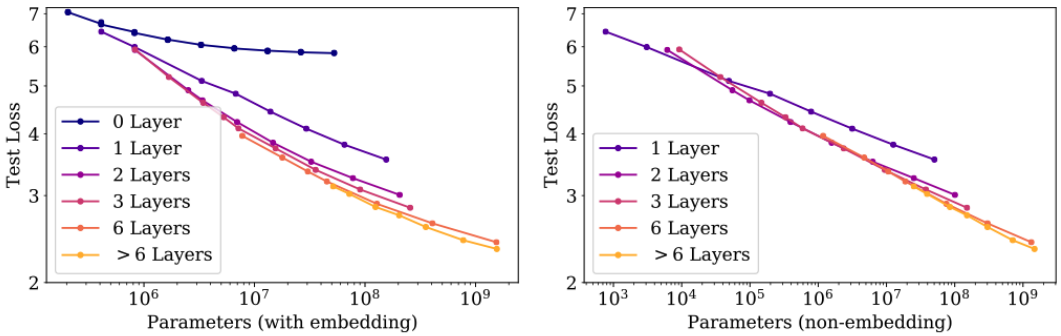


Figure 2.4.: Parameter Scaling: Model performance increases with Parameter count and is only very weakly influenced by varying architectural configurations (Kaplan et al.)

Transformer models are not able to process infinite sequences of tokens. The maximum context size, or context window, defines the longest sequence of tokens the model can process in a single forward pass. Architecturally, it determines the dimensions of the self-attention matrix $(n_{\text{max}} \times n_{\text{max}})$ and the length of the positional encodings, independent of the embedding dimension d_{model}, attention head size d_k, or number of heads h. Choosing a larger context window increases memory and compute requirements

quadratically, and thus represents a key design trade-off in transformer architectures.

2.2.2. Compute

Similar to the previously explained neural networks, Foundation Models learn by iteratively adjusting parameter values to minimize loss on a training dataset. Each training step requires a forward and a backward pass through the model.

While there are multiple and more complex approaches to training models, such as contrastive or diffusion learning for image generation [HJA20], the following assumes a simple next-token predicting language model to illustrate the compute demand of model development and training.

In such a model, input text is tokenized, embedded, passed through the network's layers according to the current weights and activation functions, and finally transformed into a probability distribution over the vocabulary. During inference, the token with the highest likelihood is selected; during training, the full distribution is used to compute the loss.

At the start of training, all parameter weights and biases are randomly initialized, so predictions are essentially random. In each step, the model processes a batch of text from the training dataset and produces an output distribution. Since the correct next token is known, the prediction error can be quantified using a loss function such as cross-entropy. The backward pass then updates the model's parameters using gradient descent. Over many iterations, the parameters converge toward values where further improvements become minimal.

The computational cost of this process is typically expressed in floating-point operations (FLOPs), the number of arithmetic operations involving decimal values performed during training. FLOPs per second (FLOP/s) measures the speed of the hardware executing these operations. Larger models, longer sequences, and bigger batch sizes all increase the FLOPs per training step. In the chart below by Kaplan et al., the performed computations are given in PF-days, meaning PetaFLOPs/s-days. This is equivalent to the number of operations performed with a throughput of 10^{15} FLOPs/s over one full day, or $8.64 \cdot 10^{19}$ operations in total. For comparison, an Apple M1 Pro chip has a computation speed of roughly 8 TeraFLOPs/s, meaning it could perform one PetaFLOPs/s-day of work in about 125 days.

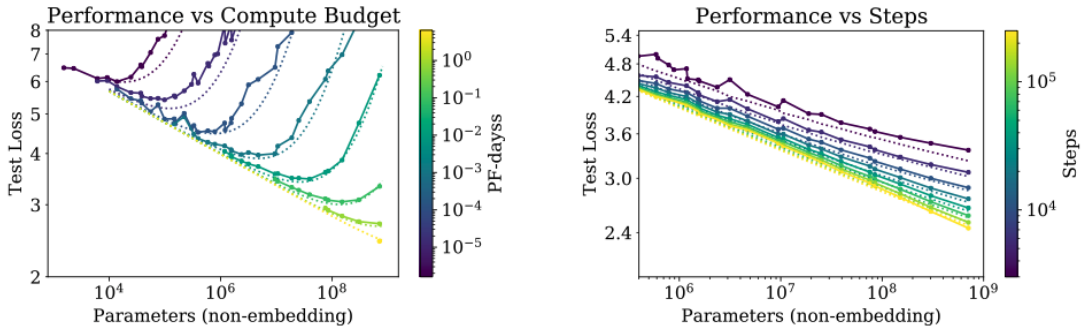


Figure 2.5.: Compute Scaling: Given the parameter count & desired model performance, required compute can be estimated based on power laws (Kaplan et al.)

Kaplan et al. demonstrated that, assuming sufficient training data, model performance (as measured by test loss) follows a universal power law in relation to model size and the total compute used for training. This insight makes it possible to estimate the compute required for a given parameter count and target performance before starting training, given an expected batch size and number of training steps.

As model sizes increase exponentially, so do the compute requirements to achieve competitive performance. For example, the Grok 3 model, published in February 2025 by xAI, has a parameter size of $395 \cdot 10^9$ and was trained in two phases. First, 100,000 of the most powerful NVIDIA H100 GPUs synchronously trained the model for 122 days. In the second phase, the model was trained for another 92 days on 200,000 H100 GPUs, totaling 30.6 million GPU-days. The estimated total compute amounts to roughly $3.5 \cdot 10^{26}$ FLOPs. [Bun25] On a single Apple M1 Pro chip, this computation would take around 1.3 million years.

Despite advances in scaling laws and hardware performance, compute availability remains a major bottleneck for training state-of-the-art Foundation Models. Access to large clusters of high-end GPUs or TPUs is restricted to a small number of well-funded organizations, and global supply constraints for advanced chips such as NVIDIA's H100 further exacerbate the issue. In addition to hardware shortages, the financial cost of large-scale training runs can reach hundreds of millions of dollars, making compute one of the key limiting factors in model development. As a result, many research groups focus on fine-tuning or distillation of existing models rather than full-scale training from scratch.

2.2.3. Data

Alongside model architecture and compute, the quality and quantity of training data is a decisive factor in the performance of Foundation Models. Without sufficient and diverse data, large models risk underfitting, developing narrow domain expertise, or inheriting systematic biases. Broad and representative datasets allow the model to capture statistical patterns across languages, domains, and modalities, enabling robust zero-shot and few-shot capabilities. [Bha+24]

Depending on the target modality, training data may consist of text, images, audio, video, or combinations thereof. Language models are commonly trained on large-scale text corpora including web crawls such as Common Crawl, Wikipedia, digitized books, news articles, and academic papers. Vision-language models rely on paired image–text datasets such as LAION-5B or COCO [Lin+15], while code models are trained on collections of source code from public repositories such as The Stack [Koc+22]. Speech models use corpora like LibriSpeech [Pra+20] or Common Voice [Ard+20], and in some cases, synthetic data [BTS24] generated by other models is incorporated to augment scarce or domain-specific datasets.

Several large, publicly available datasets have become benchmarks in model development. The Colossal Clean Crawled Corpus (C4) [Dod21] is widely used for language pre-training and is derived from a filtered and deduplicated version of Common Crawl. Wikipedia and BooksCorpus provide curated, high-quality text in specific domains. LAION-5B offers over five billion image–text pairs for multimodal pre-training [Sch+22], and LibriSpeech is a standard speech recognition dataset containing approximately 1,000 hours of read English speech. While the exact data composition of proprietary models like GPT-4 or Claude is undisclosed, open models such as LLaMA 2 report training on mixtures of publicly available and academically licensed datasets, filtered for quality and deduplication [Tou+23].

The role of data in scaling was initially underappreciated in early work by Kaplan et al., who assumed effectively infinite training data and focused on scaling laws relating compute, model size, and performance [Kap+20]. Hoffmann et al., in the Chinchilla paper, showed that for a fixed compute budget, model performance is maximized when the number of training tokens is proportional to the number of model parameters [Hof+22]. This result implies that many large models before Chinchilla were undertrained, wasting potential performance by using insufficient data. Today, training strategies are increasingly guided by balancing three factors: parameter count, dataset size in tokens, and available compute.

Despite these advances, significant challenges remain in assembling training datasets. Large-scale web data often contains misinformation, offensive content, and low-quality text, which may be memorized and reproduced by the model. Language and cultural representation is heavily skewed toward English and Western contexts, limiting performance for underrepresented languages and domains.

The legal status of web-crawled datasets is contested, with lawsuits ongoing regarding the use of copyrighted works for AI training. Finally, many high-quality datasets, particularly in specialized fields such as medicine or law, are proprietary and inaccessible to most researchers. These constraints have driven interest in synthetic data generation, transparent dataset documentation, and collaborative efforts to curate legally compliant, high-quality corpora at scale.

2.2.4. Summary

In summary, the performance and capabilities of Foundation Models emerge from the interplay of architecture, compute, and data.

Model architecture defines the number of parameters and the internal mechanisms, such as self-attention and feed-forward layers, that process and store knowledge.

Compute determines how efficiently these parameters can be optimized during training, with scaling laws providing guidance on the relationship between model size, dataset size, and performance.

Finally, the quality, diversity, and quantity of training data shape the patterns the model can learn, influencing generalization, robustness, and potential biases.

Together, these components form a tightly coupled system: increasing parameter count without sufficient data or compute may underutilize potential capacity, while ample data and compute without an appropriate architecture can limit performance gains.

Understanding these interdependencies is essential for designing, training, and evaluating modern Foundation Models.

2.3. Existing Surveys on Foundation Model Development

The number and diversity of Foundation Models under development has increased sharply in recent years, driven by advancements in compute, data availability, and research investment. Dozens of new models are released annually, spanning a wide range of modalities, architectures, and application domains. This rapid growth makes it increasingly challenging for researchers, policymakers, and industry stakeholders to maintain a comprehensive overview of the field.

Meta-surveys and tracking initiatives, such as those conducted by Epoch AI and the Stanford AI Index, address this need by systematically collecting, aggregating, and analyzing data on model characteristics, performance, and broader deployment trends. These reports provide high-level perspectives on the evolution of AI capabilities while also documenting technical and infrastructural developments over time.

In the context of this thesis, which aims to develop a tool for automating literature reviews and identifying key characteristics of Foundation Models, such surveys serve both as reference material and methodological inspiration. They illustrate established practices for compiling large-scale, heterogeneous data sources into structured and accessible outputs. The following sections examines two prominent examples: the empirical tracking work of Epoch AI and the comprehensive annual analyses of the Stanford AI Index Report.

2.3.1. Stanford AI Index Report

The Stanford AI Index Report is an annual publication that tracks, analyzes, and visualizes global trends in artificial intelligence across research, industry, policy, and societal impact [Mas+25]. Compiled by the Stanford Institute for Human-Centered Artificial Intelligence (HAI) in collaboration with academic, corporate, and governmental partners, the report synthesizes hundreds of data sources into a single, accessible reference.

Its scope extends beyond foundation models, covering topics such as research publication output, AI benchmark performance, investment trends, regulatory developments, public opinion, and the geographic distribution of AI talent. For technical advances, the report aggregates results from widely used benchmarks in natural language processing, computer vision, robotics, and multimodal tasks. By including both quantitative metrics and expert commentary, it provides context for the pace and direction of progress in AI.

A dedicated section on large-scale models highlights developments in model size, training data volume, and compute usage, often referencing datasets such as those maintained by Epoch AI. While the AI Index Report does not attempt exhaustive tracking of every model, it offers a broad view of AI's evolution within the global socio-technical landscape. As such, it serves as a complementary resource to more specialized repositories, situating foundation model trends within the wider AI ecosystem.

2.3.2. Epoch AI

Epoch AI is a multidisciplinary research institute investigating the trajectory of Artificial Intelligence (AI), with a particular focus on the development and deployment of large-scale Foundation Models [Epo25b].

In response to the rapidly growing number of foundation models under development, Epoch AI compiles and maintains one of the most comprehensive public datasets on model characteristics. This includes parameter counts, training compute estimates, dataset sizes, release dates, and in some cases energy consumption and carbon footprint.

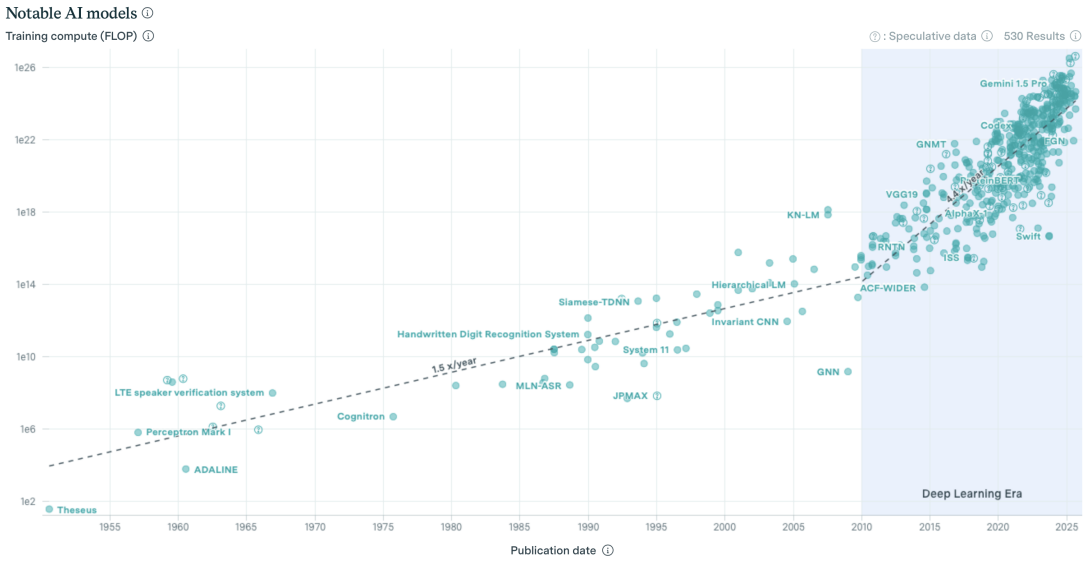


Figure 2.6.: Overview of the Notable AI Model dataset by Epoch AI, showing the development of Training compute over time

The data is aggregated from public announcements, academic papers, technical blogs, and indirect estimation methods for models whose specifications are not fully disclosed. While proprietary models often omit key details, Epoch AI employs inference techniques based on scaling laws and hardware performance benchmarks to estimate missing values. These efforts are documented alongside confidence levels, allowing users to gauge the reliability of individual data points.

Over time, Epoch AI has manually cumulated a comprehensive database of over 2700 models, tracking key factor driving machine learning progress. The dataset divides the models into three potentially overlapping groups:

i Notable AI models are those, either providing state-of-the-art improvement on a recognized Foundation Model benchmark, being highly cited with over 1.000 citations, high historical relevance for the advancement of Artificial Intelligence research, or seeing significant adoption by over a million monthly users per month.

ii Frontier models are models that were in the top 10 by training compute at the time of their release.

iii Large-scale models, which were trained with over 10^{23} FLOPs of compute, stemming from legislative definitions classifying models by their FLOPs

Epoch AI's tracking has become a reference point for AI governance researchers, policymakers, and industry analysts. By quantifying trends in model scaling and release patterns, their work enables meta-analyses of the AI ecosystem and supports empirical studies on the economic, environmental, and societal impacts of advanced AI systems. [Epo25a]

2.3.3. Further publications

Next to these large, regularly occurring reports, many researchers have performed individual and smaller scope meta-surveys on developments in the field of AI & Foundation Models.

Sevilla et al. conducted a study on 123 milestone Machine Learning systems to better understand compute trends of Machine Learning. In their research, they identified three eras with significantly difference compute growth rates. In the Pre Deep Learning Era compute closely followed Moore's law roughly doubling every 20 months. Starting in the early 2010s during the Deep Learning Era the doubling increased to approximately every 6 months. In late 2015, the Large-Scale Era emerged with firms developing with

10 to 100-fold larger compute requirements [Sev+22].

In a new study, Kumar and Manning explore Foundation Models categorized as general-purpose AI with systemic risk - a term introduced by the EU AI Act describing models with using greater than 10^{25} FLOPs. In their work, they first evaluated the development of Foundation Model compute requirements over the last decade to then model multiple scenarios on the future computational development and number of newly released models. [KM25].

In addition to these compute focused meta-studies on general-purpose Foundation Models, there also exist many publications on the development of Artificial Intelligence in specific fields.

Takita et al. have conducted a systematic review and meta-analysis of diagnostic performance comparisons between generative AI and physicians. Analyzing 83 studies, they were able to benchmark multiple models against each other and real doctor performance to create a broader overview on the capabilities of Artificial Intelligence in Medicine [Tak+25].

In the field of Robotics, a large survey and meta-analysis by a team around Hu, Xie and Jain from Carnegie Mellon University summarizes current research methodologies of Foundation Models for Robotics. Besides unifying information on methods used, the survey provides common robotics dataset and highlights current challenges seen across the research field [Hu+24].

As AI development accelerates, the role of broad, regularly updated surveys, complemented by targeted domain-specific analyses, will become increasingly important for navigating the rapidly expanding landscape of Foundation Models.

2.4. AI-Based Tools for Literature Review Automation

The literature review process is a cornerstone of academic research, yet it is often time-consuming and labor-intensive. With increasing capabilities of Large Language Models, AI-powered tools have emerged to streamline various stages of this process, from source discovery to citation management. These tools provide enhanced efficiency, accuracy, and depth in literature reviews.

2.4.1. Overview of AI Tools for Literature Reviews

AI tools for literature reviews can be categorized based on the specific tasks they assist with:

- **Source Discovery:** Tools that help researchers find relevant academic papers quickly.
- **Summarization:** AI systems that condense lengthy papers into concise summaries.
- **Evidence-Based Search:** Tools that provide direct, evidence-supported answers to specific research questions.
- **Research Mapping:** Applications that visualize connections between studies and authors.
- **Conversational Analysis:** AI systems that facilitate interactive exploration of academic papers.
- **Data summaries:** AI-based system that automatically extract certain data points from many papers

Several tools have emerged that implement one or more of these functionalities.

2.4.2. Selection of Commercial AI Tools for Literature Reviews

The market for literature review AI tools is broad and applications offer similar functionalities, without any domination player yet emerging. Following, a non-exhaustive list of these products is provided including a short description of their capabilities:

Sourcely is an AI-powered academic search assistant that offers access to over 200 million research papers. It combines advanced search tools, smart organization features, and automated summarization to assist researchers in locating, assessing, and

organizing academic sources efficiently [Sou25].

Consensus is an AI-powered search engine that provides evidence-based answers across various academic fields, including economics, medicine, and social policy. It delivers direct, evidence-supported answers to specific research questions, saving time by quickly identifying crucial papers relevant to the query [Con25].

ChatPDF is an AI-powered tool that enables conversational document analysis. Researchers can upload PDFs and interactively query the content, facilitating a deeper understanding of individual papers without extensive manual reading [Cha25].

Scholarcy is an AI tool that breaks down complex papers into concise summaries. It extracts key points, figures, and references, helping researchers quickly grasp the essence of a paper and decide on its relevance to their work [Sch25].

Noticeably and in contrast to the topic of this thesis, most tools focus on summarizing literature and quickly highlighting the main findings. The ability to structurally extract data points from hundreds of automatically selected papers, is not a common feature.

2.4.3. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a technique that combines the generative capabilities of Large Language Models with an external retrieval system. Instead of relying solely on the model's internal parameters, RAG augments the generation process with documents retrieved from a dedicated knowledge source, thereby grounding outputs in verifiable evidence [Lew+21].

Technically, a RAG system operates in two stages. First, a retriever component selects the most relevant passages from an external knowledge base, often using dense vector embeddings and similarity search. Second, these retrieved passages are appended to the user query and passed to the language model, which conditions its generation on both the query and the retrieved context. This architecture allows the model to dynamically access up-to-date or domain-specific information without retraining.

For the domain of literature review automation, RAG offers some advantages. It enables tools to provide answers that are both contextually relevant and supported by citations to original sources. This reduces the risk of hallucinations, increases transparency, and ensures that generated content can be traced back to peer-reviewed research. Consequently, many of the tools introduced above, build their functionality

on retrieval-augmented methods.

Despite these potential advantages, RAG was not applied in the context of this thesis. The research objective here is not to provide conversational answers to arbitrary queries, but rather to extract structured data points from a predefined corpus of academic publications. Since the dataset is curated in advance, the benefits of retrieval are limited, while the added architectural complexity of a RAG system would not align with the goal of efficient large-scale data extraction.

2.4.4. Implications for Literature Review Automation

The integration of AI tools into the literature review process offers several advantages:

- **Efficiency:** Automates time-consuming tasks, allowing researchers to focus on analysis and synthesis.
- **Depth:** Facilitates comprehensive exploration of literature across disciplines.
- **Organization:** Enhances the management and structuring of research materials.

These tools not only expedite the literature review process but also enhance the quality and scope of academic research. As AI continues to evolve, its role in academic research is expected to expand, offering even more sophisticated capabilities for literature review automation.

3. Methodology

This thesis conducts a large-scale, systematic literature review of Foundation Model development across various application domains. To achieve this, a software tool has been developed that automates the manual paper review and utilizes LLMs for the information retrieval process to extend the currently existing overviews.

3.1. Research Design

Conducting a manual literature review across hundreds of papers is both time-consuming and labor-intensive. To address this, the data pipeline developed in this thesis automates the process, enabling the retrieval of specific data points from large volumes of literature efficiently and accurately.

At a high level, the pipeline consists of four main stages:

1. **Data Extraction**
   The pipeline can connect to various data sources via dedicated APIs or, where necessary, potentially web scrapers. After performing keyword-based searches, the title, abstract, and additional metadata of candidate papers are evaluated using an LLM-based relevance filter. Papers deemed relevant have their metadata and full PDFs stored locally for subsequent processing.

2. **Information Retrieval**
   To process the full content of stored papers, PDFs are converted into machine-readable, text-based markdown using an external Optical Character Recognition (OCR) service. The resulting files are then optimized and submitted to an externally hosted LLM with additional instructions for structured information extraction.

3. **Robust Storage**
   Extracted information is handled and stored in a database designed for maximum flexibility, enabling easy adaptation to new extraction targets and fast querying.

4. **Insight Generation**
   Finally, the database supports efficient queries to generate insights quickly, which can be visualized using charts and other analytical tools.

Each stage of the pipeline will be described in detail in the following chapter.

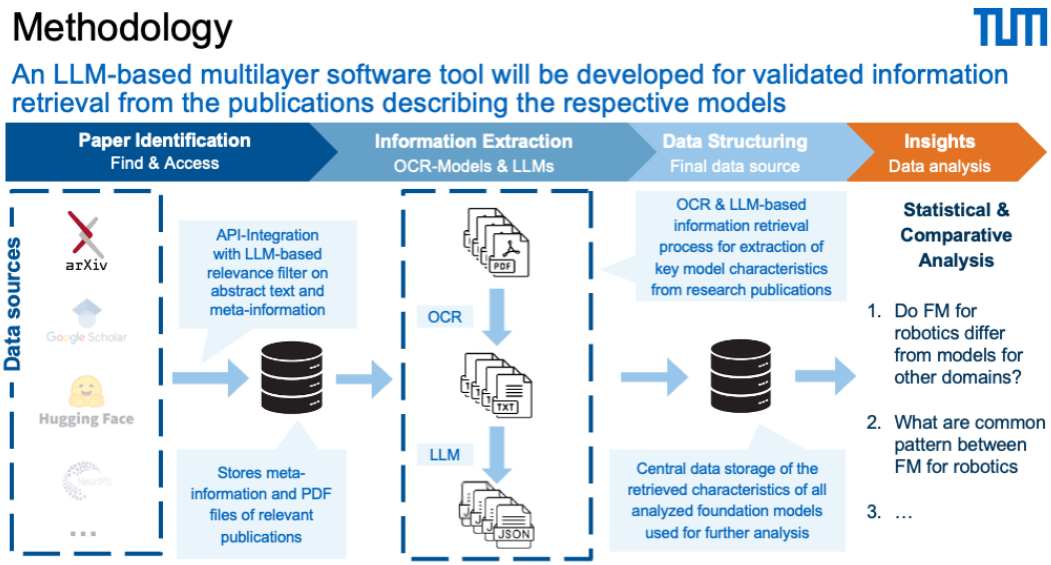


Figure 3.1.: Architecture of the data pipeline enabling efficient and accurate automated data extraction.

3.2. Data Pipeline

3.2.1. Paper Identification

The goal of the data pipeline is to automatically extract specific data points from a vast number of publications. Hence, the ability to find, identify, and select relevant publications is a requisite for high-quality output data and autonomy.

Relevant publications can be published in a multitude of formats and are also accessible through many platforms. To limit the scope of this thesis, the current implementation is designed to only access publications available on arXiv, a curated research platform open to anyone.

However, the architecture of the data pipeline is fully modular and can be easily extended. Publications can already be added manually, and integrating further publication repositories may only involve connecting to their API to retrieve some meta-information like Titles, Abstracts, and Publication dates, as well as the actual publication in a PDF format.

3.2.1.1. arXiv Integration

arXiv offers public API access in order to maximize its openness and interoperability.

Using the API, publications can be searched based on given search terms. In addition, more complex queries can also be executed to further specify filters on for example publication dates, or research areas, similar to the advanced search on the arXiv website.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry>
    <id>http://arxiv.org/abs/cond-mat/0102536v1</id>
    <updated>2001-02-28T20:12:09Z</updated>
    <published>2001-02-28T20:12:09Z</published>
    <title>Impact of Electron-Electron Cusp on Configuration Interaction Energy</title>
    <summary> The effect of the electron-electron cusp on the convergence of
        configurationinteraction (CI) wave functions is examined. By analogy with
        thepseudopotential approach for electron-ion interactions, an effective electron-
        electron interaction is developed which closely reproduces thescattering of the
        Coulomb interaction but is smooth and finite at zeroelectron-electron separation.
         We perform CI and quantum Monte Carlo calculations for He and Be atoms, both
        with the Coulomb electron-electron interaction and with the smooth effective
        electron-electron interaction. We find that convergence of the CI expansion of
        the wave function for the smooth electron-electron interaction is not
        significantly improved compared with that for the divergent Coulomb interaction
        for energy differences on the order of 1 m Hartree. This shows that, contrary to
         popular belief, description of the electron-electron cusp is not a limiting
        factor, to within chemical accuracy, for CI calculations.
    </summary>
    <author>
      <name>David Prendergast</name>
      <arxiv:affiliation xmlns:arxiv="http://arxiv.org/schemas/atom">Department of
          Physics</arxiv:affiliation>
    </author>
</entry>
```

Listing 3.1: Examplatory XML response of a request to the arXiv search API

The response of the arXiv search API contains all relevant meta-information needed for classifying the relevance of a given publication. In particular, it returns the Title as

well as a Summary, which is in most cases simply the abstract of the publications. In addition, the response also includes the internal arXiv ID, which is needed at a later stage in the pipeline, when accessing the full publication in PDF format.

3.2.1.2. LLM-based Relevance Filter

At the time of writing, arXiv hosts just over 2.8 million scholarly articles and publications.

Querying all these publications often leads to high numbers of matches on the search terms, especially when also allowing search term existence in the publication's summary.

In general, this is good as it means the data extraction can be performed on many publications and lead to broad and relevant insights.

When actually extracting the underlying information, however, this can lead to excessive computing, high cost, and poor data quality, in the case where publications are falsely selected based on the keyword search, although in reality they are not relevant for the desired analysis.

To avoid this and reduce the number of false positives from the keyword search, the developed data pipeline utilizes a second verification mechanism for selected arXiv publications.

The underlying idea of the second filter is to let a Large Language Model process the Title and Summary of a publication, together with a structured, standardized prompt giving instructions on the exact desired specifications of publications. Based on the provided information, the LLM shall then classify whether the specific publication is relevant for further processing in the data pipeline.

Below, a full prompt for classifying publications is provided. In this particular case, the prompt is for evaluation, whether a given publication introduces a new Foundation Model in the Robotics domain.

`{title}` and `{abstract}` serve as placeholders, which get automatically replaced with the actual Title and Summary of the publication, before making the API request to the LLM. The prompt provides exact criteria for the LLM to decide on, and also examples to clarify behavior in potential edge cases. This is commonly known as Few-Shot Prompting, which has been shown to boost the accuracy of responses [IV+21]. Lastly, the desired response format is specified to enable automated processing by the

next steps in the pipeline. In this specific case, the `is_foundational_robotics_model` key value pair in the JSON-style response will be used to neglect irrelevant publications.

```
FOUNDATIONAL_MODEL_CLASSIFIER_PROMPT = """
Analyze this paper title and abstract to determine if it introduces a new foundational
    model in robotics.

Title: {title}

Abstract: {abstract}

Criteria for foundational model in robotics:
1. Introduces a NEW model/architecture (not just applications of existing models)
2. Specifically targets robotics applications or robot learning
3. Claims to be general-purpose, foundational, or broadly applicable
4. Shows applicability across multiple robot tasks or domains
5. Focuses on model architecture, training methodology, or core capabilities

Examples of what QUALIFIES:
- New transformer architectures for robot learning
- Foundation models specifically trained for robotics
- General-purpose robot policies or world models
- Novel multi-modal models for robot perception and control

Examples of what does NOT qualify:
- Applications of existing models to specific robot tasks
- Task-specific solutions (single-purpose controllers, planners)
- Datasets, benchmarks, or evaluation frameworks
- Hardware or mechanical innovations
- Simulation environments or tools

IMPORTANT: Respond with ONLY valid JSON. No Markdown formatting. Use this exact format:
{{
  "is_foundational_robotics_model": boolean,
  "confidence": float between 0.0 and 1.0,
  "reasoning": "brief explanation",
  "model_name": "extracted model name or null",
  "robotics_domains": ["list of relevant domains"]
}}
"""
```

Listing 3.2: Prompt used to classify the relevance of publication for an analysis on Foundation Models in the rovotics domain

The LLM-based publication filter significantly reduces the amount of qualifying papers and thereby enhances the final data quality.

3.2.2. Information Extraction

The first stage of the developed data pipeline focuses on identifying, qualifying, and selecting publications that are relevant to the research question or planned analysis. Once identified, this stage extracts the desired data points from the full text of these publications.

At a high level, the process proceeds as follows: the pipeline requests the PDF of a publication from arXiv, converts the PDF into a machine-readable format, and subsequently submits the full text to a Large Language Model (LLM) for information extraction.

3.2.2.1. Publication Procurement & Conversion

The extraction process begins with obtaining access to the full publication. The arXiv repository provides a dedicated endpoint to download PDFs directly, given a publication's unique arXiv ID. These IDs, which are included in the initial search response, enable automated download and local storage of all publications that were previously marked as relevant by the filtering stage of the pipeline.

To ensure robustness, it is important to account for technical constraints: the arXiv API enforces rate limits to prevent excessive or malicious requests. As a result, publications must be retrieved sequentially with enforced waiting times between requests. When processing a large number of papers, this limitation can noticeably increase total runtime. To mitigate this, a caching mechanism is implemented in the pipeline: before requesting a file from arXiv, the system first checks whether the publication already exists in local storage and only fetches missing ones.

After the PDFs have been procured, they must be transformed into a text-based representation that can be processed by an LLM. This is achieved using Optical Character Recognition (OCR), a technology that converts printed, handwritten, or otherwise embedded text into machine-readable form. With the advent of advanced computer vision models, OCR has become highly accurate and widely accessible.

In this pipeline, the `Mistral OCR 2503` model is employed to convert PDFs into Markdown files. Markdown is particularly suitable for this task, as it preserves hierarchical structures such as headings and subheadings, and supports the representation of diverse content types such as tables, figures, and code snippets in a structured yet lightweight text format.

The conversion step is performed via a third-party API provided by Mistral AI, which exposes a dedicated OCR endpoint. Pricing is usage-based, with a cost of approximately 1 USD per 1,000 pages. Performance-wise, up to 2,000 pages can be processed per minute in a single request, and processing can be parallelized to a certain extent, making it scalable to large document collections.

Model	Overall	Math	Multilingual	Scanned	Tables
Google Document AI	83.42	80.29	86.42	92.77	78.16
Azure OCR	89.52	85.72	87.52	94.65	89.52
Gemini-1.5-Flash-002	90.23	89.11	86.76	94.87	90.48
Gemini-1.5-Pro-002	89.92	88.48	86.33	96.15	89.71
Gemini-2.0-Flash-001	88.69	84.18	85.80	95.11	91.46
GPT-4o-2024-11-20	89.77	87.55	86.00	94.58	91.70
Mistral OCR 2503	**94.89**	**94.29**	**89.55**	**98.96**	**96.12**

Table 3.1.: Performance comparison of OCR models across multiple tasks. [Mis25a]

Similar to the caching mechanism used for PDF files, the converted Markdown files are also stored locally. Before triggering an OCR conversion, the pipeline verifies whether a Markdown version of the publication already exists. This approach ensures both cost efficiency and improved overall performance.

3.2.2.2. LLM-based Data Extraction

Once the publications have been converted and stored as Markdown files, the next step is to prepare and execute the LLM-based information extraction. The process begins by constructing a detailed prompt that guides the model in retrieving the desired information.

The prompt itself is comparatively long and structured into several components. First, it defines the *role* of the model and clearly outlines the task to be performed. Next, it specifies the exact data fields to be extracted. For illustration purposes, the following visualization omits or shortens some of these fields compared to the full prompt used in practice.

```
You are a PHD-level computer scientist and research analyst. Your task is to extract key information from a scientific publication provided in Markdown format.

The publication content will appear after the marker <<<PubStart>>>.

Your extraction goals are to identify and extract technical details for **each distinct model variant** described in the paper (e.g., ModelName-Mini, ModelName-Large, ModelName-Pro). Each model variant must be represented as a separate JSON object.

For each model variant, extract the following items (if present):
1. Model name
2. Domain Select all fitting domains from the following standardized list (do not invent or paraphrase):
["3D modeling",
  "Audio",
    ...
  "Video",
  "Vision"]
...
6. Parameters (model size in parameters)  For Mixture of Experts models, report only the TOTAL parameter count, not activated parameters (e.g., if you see "389B total, 52B activated", report "389B"). When a range is given (e.g., "50-70B parameters"), always select the LARGEST value ("70B"). Format as numeric values or with M/B suffixes (e.g., "4.5M", "273.9B") - never use commas, scientific notation strings, or other formats.
7. Training dataset
...
16. Input Modality  one or more of: "text", "image", "audio", "video", "multimodal"
17. Output Modality  one or more of: "text", "image", "audio", "video", "multimodal"
18. Architecture  Select **one** from the following standardized list. Only use "Other - {High-Level Architecture Name}" when truly novel architectures don't fit any category:
["Transformer (Encoder-only)",
   ...
  "Multi-architecture Ensemble",
  "Other - {High-Level Architecture Name}"]
**Architecture Selection Guidelines:**
...
- For hybrid models  choose the DOMINANT architecture type or specific hybrid category
- **For novel architectures:** Use "Other - {High-Level Architecture Name}" format (e.g., "Other - Neural ODE", "Other - Capsule Network", "Other - Memory Network")

19. Task  Specific tasks or applications the model was designed for (e.g., "text classification", "image generation", "speech recognition", "question answering")
```

Listing 3.3: First section of the data extraction prompt specifing role of the model

The next section of the prompt defines the required output format for the LLM's response. A structured and, most importantly, consistent response is crucial for reliable downstream processing. However, due to the probabilistic next-token prediction at the core of these models, responses are not always generated as valid JSON. To address this limitation and ensure robustness of the pipeline, a custom parser has been implemented. This parser pre-processes the model output, correcting common inconsistencies and formatting errors before the data is stored and analyzed.

```
## OUTPUT FORMAT REQUIREMENTS

**CRITICAL**: Your response must be a **valid JSON array** containing one or more model
    objects. Follow this exact structure:

### Single Model Example:
```
[
  {
    "model_name": {
      "value": "GPT-4",
      "confidence": 100,
      "references": "Abstract, Section 1"
    },
    "domain": {
      "value": ["Language"],
      "confidence": 100,
      "references": "Abstract"
    },
    "parameters": {
      "value": "1.7T",
      "confidence": 85,
      "references": "Section 3.2"
    },
    "task": {
      "value": ["language modeling/generation", "chat", "question answering"],
      "confidence": 90,
      "references": "Abstract, Section 1"
    }
  }
]
```
```

Listing 3.4: Specifing the output requirements using few-shot prompting (1/2)

Furthermore, the prompt applies few-shot prompting to explicitly define the expected output format, covering both cases: when a publication describes a single model and when it describes multiple models.

````
### Multiple Models Example:
```
[
  {
    "model_name": {
      "value": "BERT-Base",
      "confidence": 100,
      "references": "Table 1"
    },
    "parameters": {
      "value": "110M",
      "confidence": 100,
      "references": "Table 1"
    },
    "task": {
      "value": ["language modeling/generation"],
      "confidence": 95,
      "references": "Section 2"
    }
  },
  {
    "model_name": {
      "value": "BERT-Large",
      "confidence": 100,
      "references": "Table 1"
    },
    "parameters": {
      "value": "340M",
      "confidence": 100,
      "references": "Table 1"
    },
    "task": {
      "value": ["language modeling/generation"],
      "confidence": 95,
      "references": "Section 2"
    }
  }
]
```
### Key Format Rules:
- **Always start with `[` and end with `]`** - this creates a JSON array
- **Each model is a JSON object `{...}` inside the array**
- **Separate multiple models with commas**: `[{model1}, {model2}]`
````

Listing 3.5: Specifing the output requirements using few-shot prompting (2/2)

Then, another example of a correct output that includes all data fields is provided. Lastly, some additional instructions are added to the prompt, followed by a final reminder to ensure a correct JSON output formatting.

```
---
The result must be returned as a **JSON array**. Each element in the array must be a JSON object corresponding to a **distinct model variant**.

For each model, follow this structure:
[
  {
    "model_name": {
      "value": "<<value or n/a>>",
      "confidence": <<integer 0100>>,
      "references": "<<section name or quote>>"
    },
    ...,
    "task": {
      "value": [<<list of tasks or "n/a">>],
      "confidence": <<integer 0100>>,
      "references": "<<section name or quote>>"
    }
  }
]
---
Additional Instructions:

- If multiple model variants are described (e.g., Mini, Base, Large), extract one object per variant.
...
- Do not infer from external knowledge or model naming alone - rely strictly on the publication content.
---
Confidence Scoring Guidelines (0100):
- **100**: Direct, unambiguous quote (e.g., "The model has 70B parameters.")
- **9099**: Explicit but slightly indirect (e.g., "Parameters: 70B" in a table)
...
- **5069**: Weak implication (e.g., "Trained on 1M samples" in a related section)
- **049**: Unreliable or speculative
---
**FINAL REMINDER**: Only output the JSON array. No Markdown code blocks (no ‘‘‘json‘‘‘), no commentary, no explanation. Start directly with ‘[‘ and end with ‘]‘.

<<PubStart>>
"""
```

Listing 3.6: Providing a full output example and addtional instructions (shortened)

Unshortened and configured to extract around 20 unique datapoints from a publication, the total prompt length is just over 300 lines.

Similar to the LLM-based Filter, `«PubStart»` is used as a placeholder and gets replaced with the actual Markdown representation of the publication.

Unfortunately, as mentioned before, Large Language Models do not have an infinite input size. Therefore, larger publications, if simply appended in full, can cause the processing to fail. To avoid this undesired behavior, the developed data pipeline includes a sophisticated mechanism optimizing the total request size:

If the chunking is activated, the pipeline uses a tokenizer to estimate the total Markdown file size. Depending on the size and the specified maximal chunk size, the request is either made with the full file or the chunking process starts.
In general, the goal of the chunking logic is to preserve the original document structure and provide as much context to the Large Language Model as possible within a single chunk, to ensure high compute efficiency with adequate output data quality. To achieve this, the inherent structure of Markdown files is used. Regular expressions allow slicing the file by the Markdown headings (#, ##, ###, ####) to identify the sections of the original publication.

The algorithm concatenates complete sections (defined by Markdown headings) until adding the next section would exceed the set token limit. In that case, the section that could not be added is split into its constituent paragraphs, which are then individually appended to chunks. If a paragraph itself exceeds the token limit, this process continues recursively to the sentence level, where extraordinarily long sentences are truncated at the token level as a final fallback mechanism.

To give the LLM some additional context, for each chunk, an overlap of the previous chunk's content is added as a prefix, and when the processing request is made, the LLM's output of the previous chunk is added to the prompt as a reference.

The response of the last chunk's request is then used as the final response of the publication and saved to the database.

3.2.3. Data Structuring

After the last chunk's response is parsed into a valid JSON format, the extracted results need to be stored persistently for further analysis. For this purpose, a relational

database schema was designed and implemented in PostgreSQL. The overarching design goal is to ensure *maximal flexibility* with respect to the types of data that can be extracted, while also maintaining efficiency and consistency during subsequent analysis.

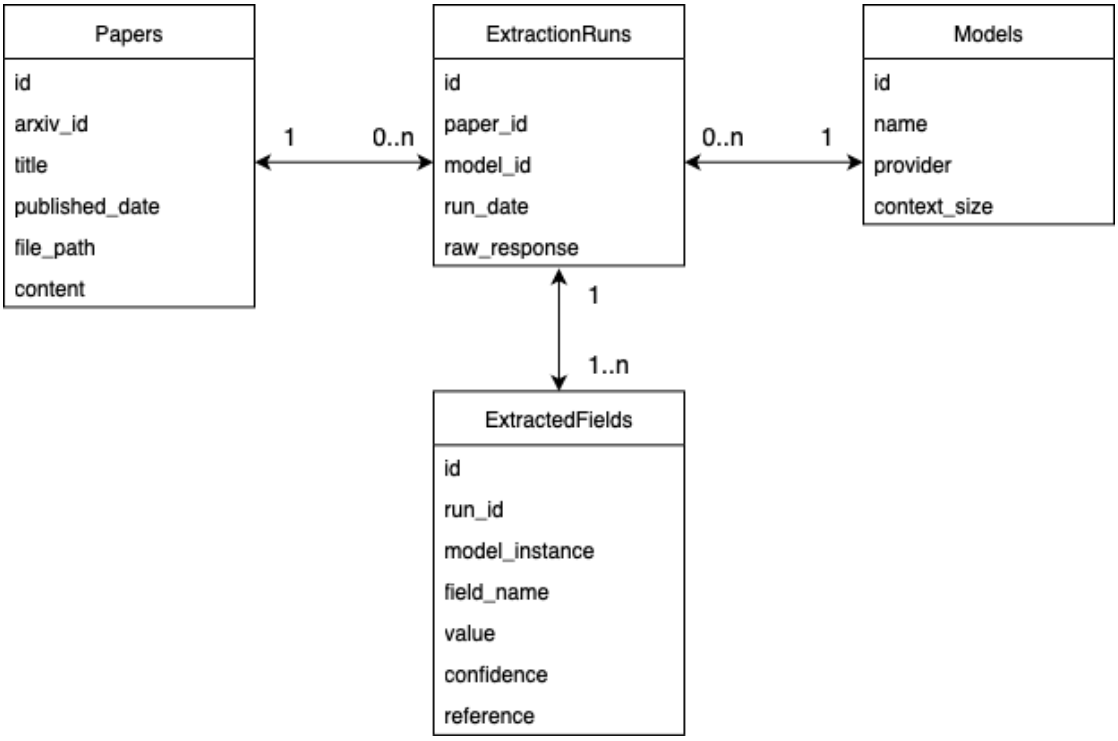


Figure 3.2.: The database structure is designed to maintain maximal flexibility regarding the extractable values.

The schema consists of four main tables:

1. **Papers**
   Stores metadata for all processed papers. In addition to identifiers such as the arXiv ID, the table holds the full machine-readable content. This enables repeated extractions from the same papers without reprocessing. However, storing full content may become a scalability concern if the number of papers increases significantly.

2. **Models**
   Contains metadata about the Large Language Models used during extraction,

including their name, provider, context size, and version. This allows precise tracking of which model produced a given extraction result.

3. **ExtractionRuns**
   Represents a single execution of an extraction for a specific paper–model combination. Each entry stores metadata such as the execution date, model parameters (e.g., temperature), and the raw LLM response. This table serves as the central link between papers, models, and extracted values.

4. **ExtractedFields**
   Contains the individual key–value pairs produced by each extraction run. Each entry records the extracted field name, its value, confidence score, and optional supporting references. In cases where a paper describes multiple versions of a model, the `model_instance_id` field differentiates them. By not fixing extractable values in the schema, the design achieves maximum flexibility: different runs can target different fields without modifying the database structure.

This schema balances simplicity and adaptability, enabling the pipeline to handle evolving extraction requirements while keeping data analysis efficient.

4. Results

4.1. Pipeline Performance Benchmark & Evaluation

The entire approach of this master's thesis has been centered around not only building an LLM-based data extraction tool, but also having the ability to correctly benchmark its performance on large datasets.

Hence, the following sections detail the extracted data quality, which is used as a reference for the overall pipeline performance.

4.1.1. Evaluation of the LLM-based Filter

Prior to analyzing the final output data quality, the accuracy of the LLM-based filter used to triage relevant publications is investigated.

To evaluate the publication filter, the underlying prompt has been slightly adjusted to be application domain agnostic compared to the robotic-specific one used in the production pipeline, and allows for a benchmark based on the EpochAI dataset as ground truth.

In the analysis, 253 well-documented papers introducing new foundational models were selected as positive samples from the EpochAI dataset, and an additional 228 random papers from various scientific fields were published on arXiv as negative samples. Using the adjusted prompt, the papers have then been analyzed based on their Title and Abstract by the Google Gemini model.

The confusion matrix (Figure 4.1) visualizes the results: In general, the LLM-based filter showed High Precision and rather Low Recall values across multiple configurations.

In the specific case, the *Precision* was 0.990, the *Recall* 0.426 resulting in a *F1-Score* of 0.596 with an *Accuracy* of 0.701.

It needs to be noted that the benchmark comes with some inaccuracies and is not fully reliable, as the exact inclusion criteria utilized by the EpochAI researchers are

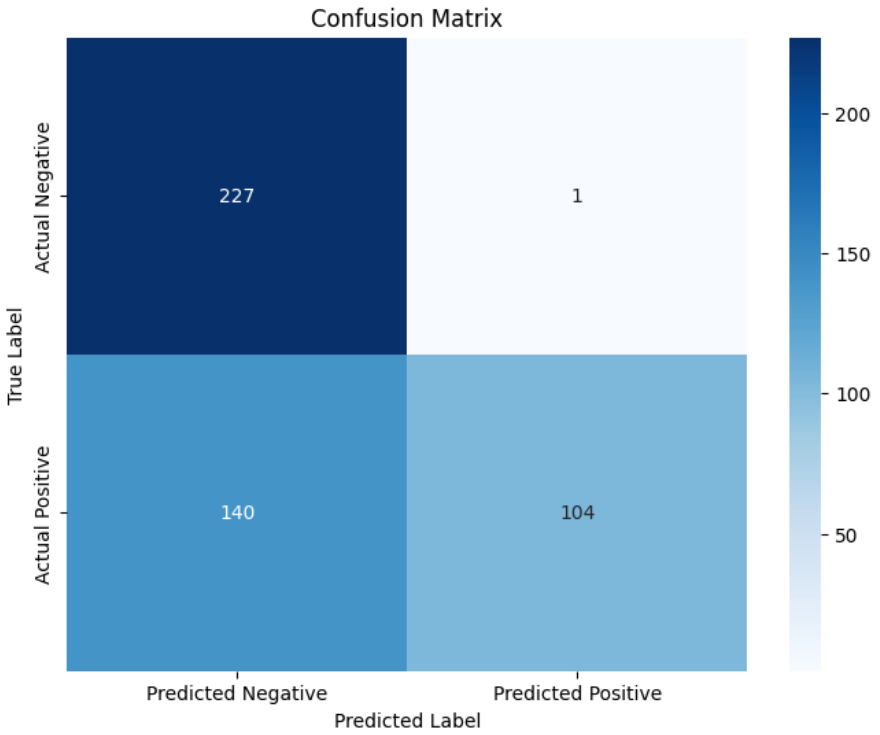


Figure 4.1.: Confusion Matrix for the LLM-based Filter benchmark on selected paper from the EpochAI dataset

not fully public and clear. Hence, it is possible that the dataset used as the benchmark includes some publications that should not qualify as positives according to the criteria defined for the filter, leading to lower recall scores and a lower F1-score.

4.1.2. Evaluation of the Data Extraction

To evaluate the performance of the developed data extraction pipeline, it was used to replicate Epochs AI datasets. More specifically, the dataset includes the arXiv IDs of 479 of the 950 models in the dataset, which were then selected as the ground-truth for the benchmark.

To enhance the accuracy and reduce the potential impact of confounding variables, the extraction has been performed multiple times with varying maximal chunk sizes on two different Large Language Models. Explicitly, the *DeepSeek-V3-0324* and *Google Gemini-2.0-flash-001* models were used via their dedicated API with set chunk sizes of

4.096 (4*k*), 8.192 (8*k*), 16.384 (16*k*), 32.768 (32*k*), 65.536 (64*k*), and 131.072 (128*k*) tokens. As the maximal input size of the DeepSeek model is only 128k tokens, and some additional tokens are added to the prompt in the API request, the analysis with a 128k chunk size could not be made. The used Google model has a maximal chunk size of one million tokens.

The benchmark performed to evaluate the extraction pipeline's result is comparing the result to the corresponding entries in the EpochAI data set, and calculating the share of correctly extracted values. To do this, the correctness of an extracted value had to be defined. For the special case of *Parameter Count* of a model, the following options count as correct:

i **Exact Numeric Value**
If the extracted numeric value exactly matches the numeric value in the EpochAI dataset, the extraction counts as correct

ii **Matching n/a Value**
If the extraction pipeline returned *n/a* since it could not find any information, and the EpochAI dataset also contained no information, the extraction counts as correct

iii **Correct Order of Magnitude** In some cases, the extracted values slightly differ from the EpochAI values, but have the same Order of Magnitude (e.g., 705 billion vs. 700 billion parameters). This can, for example, be caused by varying values in the source paper's abstract and more detailed tables. To account for this error, all values with the matching first digit and the same rounded log_{10} value are seen as correct (e.g. $7,05 \cdot 10^{11} \approx 7,00 \cdot 10^{11}$, but $7,05 \cdot 10^{11} \not\approx 8,00 \cdot 10^{11}$ and $7,05 \cdot 10^{11} \not\approx 7,05 \cdot 10^{12}$).

Importantly, an extraction is marked as correct, if at least one of the extracted model versions matches the original data according to the definition above. This is needed, as some publications introduce entire families of models with multiple versions and varying characteristics. While the original dataset only selected one in most cases, the developed pipeline is designed to create a holistic overview including all model versions.

Figure 4.2 highlights the pipeline's accuracy increase with scaling the maximal chunk size. The underlying values refer to the total number of correctly extracted values according to the definitions stated above. The increase in accuracy is similar for both models: first sharply increasing, and then a plateau after 32*k* tokens.

Interestingly, the average token length per paper is 19.798 with 88% of the papers having less than 32*k* tokens. This suggests that with further scaling of chunk

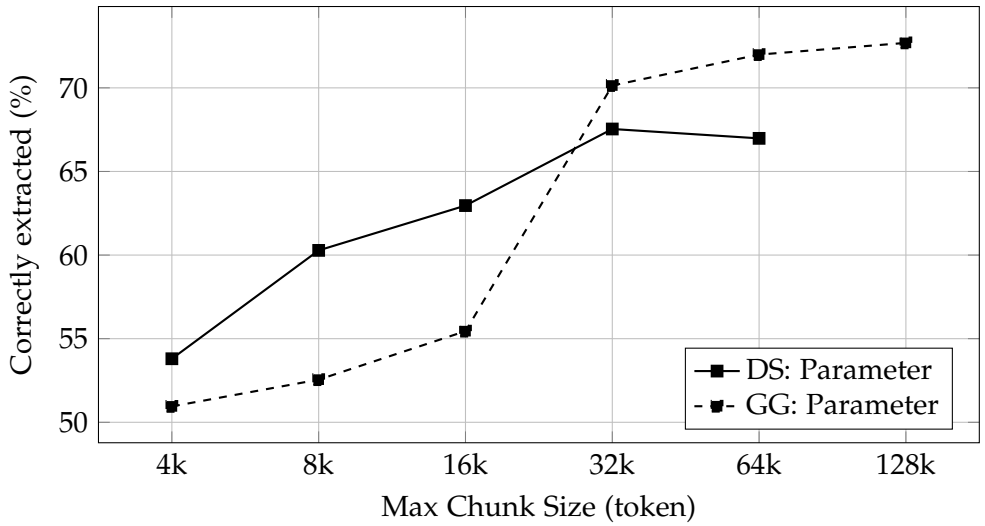


Figure 4.2.: Performance Benchmark on the Parameter value against the EpochAI dataset for DeepSeek (DS) & Google Gemini (GG)

sizes, accuracy would likely not increase much. Out of the 479 papers used in the benchmark, only one paper was more than 128*k* tokens long. Hence, with a chunk size of 128*k* tokens, virtually all papers were already being processed in a single request.

When parsing the extraction results, the developed pipeline is able to transform some non-numeric formats into numeric values, such as parsing *300B* to 300.000.000.000. However, the LLMs often returned results in formats not processable by the developed parser, as they were case-specific or did not match any specified rules.

As the percentage of such *unclear* values is fairly high, the accuracy of the developed pipeline can likely be increased by developing an even more sophisticated parser, which either covers more possible cases or potentially uses another LLM request solely for parsing the received values into a numeric format.

In comparison, the result for domain extraction is different. Here, instead of having to extract a specific numerical value, the prompt specified possible application domains of the Foundation Models, and the LLM had to select the most fitting.

For this case, *correctness* of an extraction is measured using the Jaccard index $J(A, B)$. It is a statistical concept used to evaluate the similarity of two sets, which are, in this case, the application domains of a specific model from the EpochAI dataset and the list

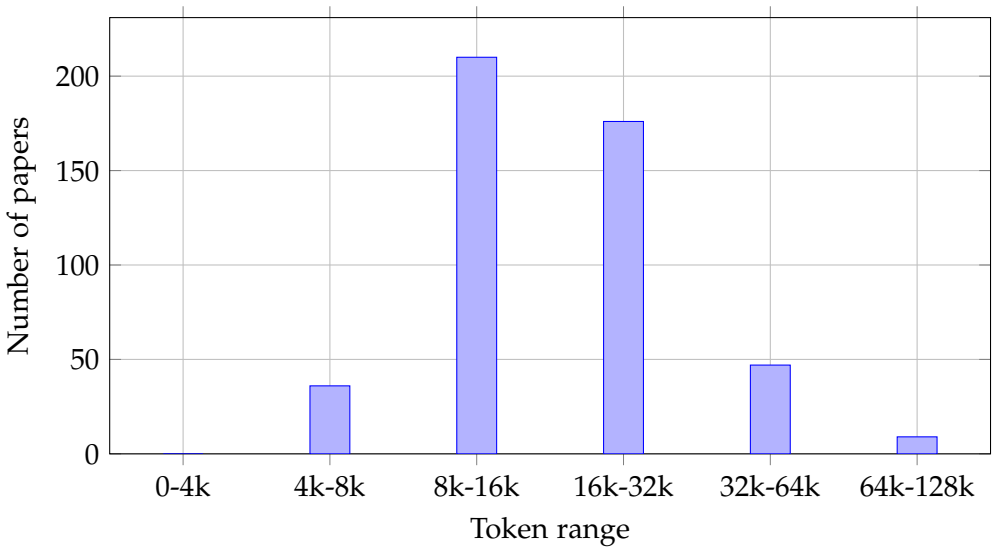
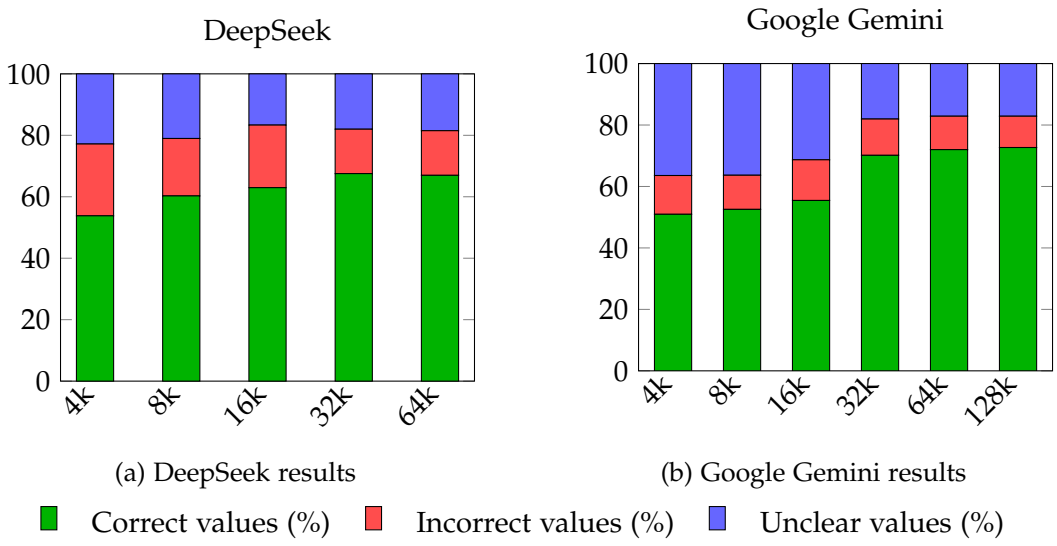


Figure 4.3.: Histogram of publication lengths measured by the token number grouped in ranges



(a) DeepSeek results

(b) Google Gemini results

Figure 4.4.: Comparison of extraction correctness between DeepSeek and Google Gemini across different chunk sizes.

of extracted application domains for the same model.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

More specifically, the average Jaccard index for a specific extraction run with a set max chunk size.

$$\bar{J} = \frac{1}{n} \sum_{i=1}^{n} J(A_i, B_i)$$

Figure 4.5 visualizes these average Jaccard indices again for the DeepSeek and Google Gemini models with varying chunk sizes. Notably, for all extraction runs independent of the model and chunk size, the median Jaccard index was 1, meaning perfect overlap.

Neglecting some noise likely introduced by the non-determinism of the LLMs, the Jaccard indices are rather constant with no noticeable growth trends, unlike the increasing Parameter accuracy with scaling chunk sizes.

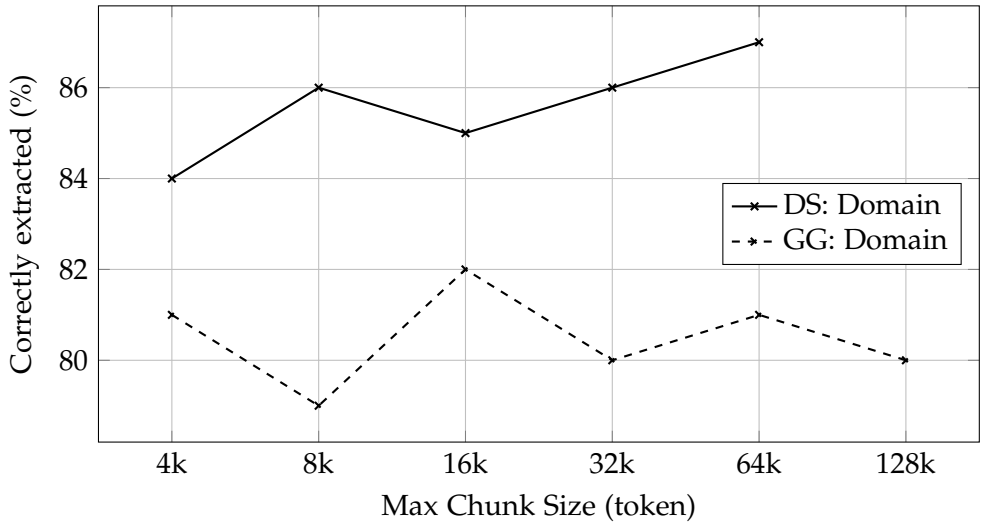


Figure 4.5.: Performance Benchmark on the Application Domain against the EpochAI dataset for DeepSeek (DS) & Google Gemini (GG) using the Jaccard index

4.1.3. Evaluation of the Pipeline's Economics

Besides the output quality of the developed pipeline, economic factors also play a role in the evaluation. To account for them, execution costs and runtimes have been analyzed.

The execution costs consist of two factors:

1) **OCR Processing**
   In the first stage of the pipeline, a publication's PDF file is transformed into a Markdown file using an external OCR service hosted by Mistral AI. The conversion costs are usage-based, with 1000 pages costing $1. As converted publications are cached and stored in Markdown format, OCR costs only apply for new publications.

2) **LLM Analysis**
   Two different Large Language Models are used for the actual data extraction, which are also hosted by their respective developers DeepSeek and Google. In general, the pricing for LLMs is divided in costs per million input tokens and costs per million output tokens. In addition, DeepSeek differentiates between prices in Standard Hours (UTC 00:30-16:30) and Discount Hours (UTC 16:30-00:30).

Table 4.1 breaks down the exact costs related to executing the pipeline and provides an example for a single average extraction of a 20-page publication. Depending on the model, this analysis costs between 2 and 4 cents USD.

Pipeline Cost Overview			
Step 1: OCR Processing	**Mistral AI OCR**		
	$1 / 1000 pages		
Step 2: LLM Analysis	**DeepSeek (Standard)**	**DeepSeek (Discount)**	**Gemini**
Input tokens[1] (per 1M)	$0.17	$0.085	$0.30
Output tokens (per 1M)	$1.10	$0.55	$2.50
Average Cost per Paper (20 pages, 20k input, 3.5k output)			
Mistral OCR cost	$0.02000	$0.02000	$0.02000
LLM cost	$0.00725	$0.00363	$0.01475
Total pipeline cost	**$0.02725**	**$0.02363**	**$0.03475**

Table 4.1.: Pipeline costs including sample calculation for an average paper (20 pages, 20,000 input tokens, 3,500 output tokens).

Expectantly, the average extraction time for a single publication reduces exponentially with experientially growing chunk sizes.

[1] Assuming an equal amount of cache hits and cache misses

As described, the Large Language Models used to extract the information process all input tokens simultaneously at a speed defined by the compute power of the deployment infrastructure. Hence, the factor influencing the extraction time is not the number of tokens in a chunk, but the number of chunks needed to process a full paper. For smaller chunk sizes, publications need to be split resulting in multiple requests to the LLMS, which take a fix amount of time.

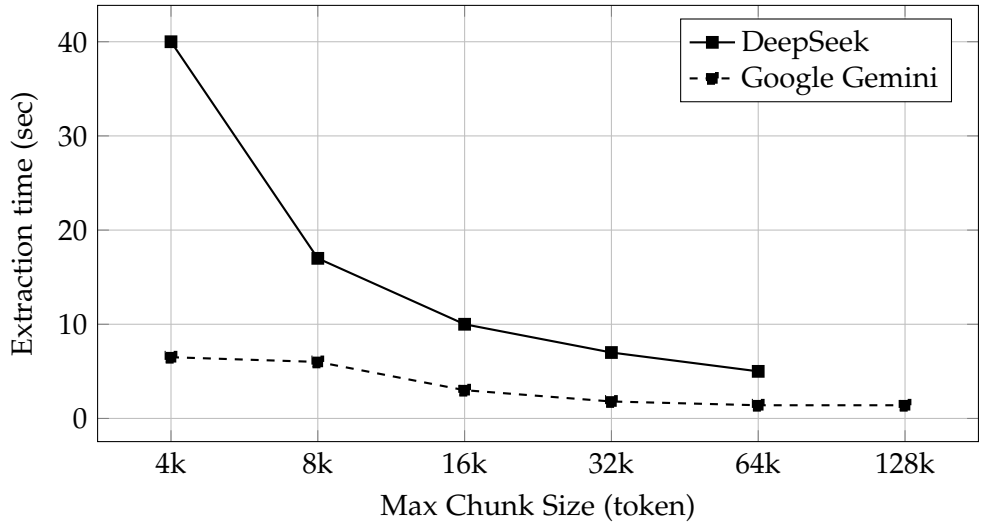


Figure 4.6.: Averaged extraction times of an average, single publication for DeepSeek & Google Gemini

4.2. EpochAI Dataset Replication

In the previous chapter, the quality of the extracted data was discussed in detail based on two different benchmarks.

Especially for the *parameter* benchmark, visualizing the extracted data reveals additional characteristics compared to the original EpochAI dataset.

Figure 4.7 displays this visualization, plotting the parameter count of models over time. The extracted values are color-coded into three groups: Original EpochAI data (Green), extracted pipeline data (Blue), and extracted data on Robotics models (Orange).

Although the pipeline ran on the same publications investigated by EpochAI, the number of extracted models is noticeably higher, with 1591 found model versions compared to the 328 models tracked in the original dataset.

The calculated parameter growth rates for the EpochAI dataset and replication are almost identical, with the difference, that the average value of the replication data is smaller. Hence, the regression result shows almost parallel lines, with the line of the replication sitting below the EpochAI line.

Noteworthily, the robotics-specific Foundation Models have significantly fewer parameters, but a similar growth rate.

Importantly, the y-axis of the chart is on a logarithmic scale. Hence, the linear growth trend would be exponential on a linear scale.

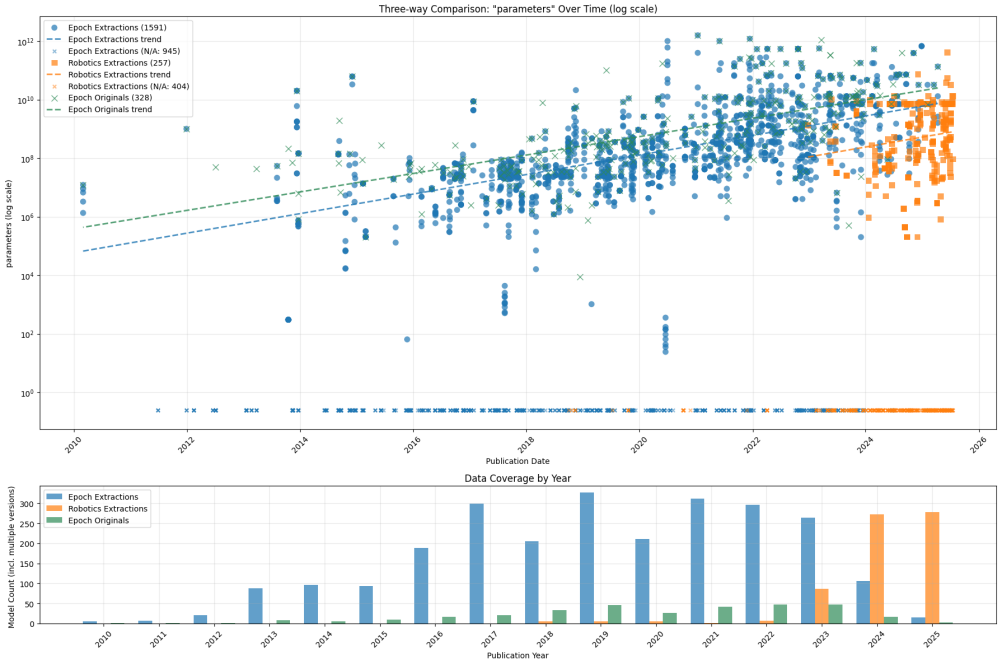


Figure 4.7.: Three-way comparison of the "Parameter size" values of the original EpochAI dataset (Green), the replication created by the developed pipeline (Blue), and newly analyzed models on Robotics (Orange) (see full-size version in Appendix A.1)

4.3. Foundation Model Landscape

The developed data extraction pipeline enables performing broad meta-analysis on vast amounts of publications. In addition, the extraction is not limited to only certain data points, but the extractable values can be adjusted and specified with every extraction run.

Table 4.2 shows the data points that have been extracted for the analysis of the Foundational Model Landscape. The *"% of N/A"* column can be used as an indicator for the data quality and availability of the extracted values. Lower values mean more values available and therefore better data quality.

It becomes clear that the created dataset is broad in scope and allows for many analyses to be performed on it. In the following chapter, selected insights will be shared. However, the highlighted analyses are not exhaustive, and the dataset allows for many other analyses.

Notably, for the following figures, charts, and analyses, the dataset has been extracted based on the publications analyzed in the EpochAIs dataset. The number of publications per year varies, and especially for the early years, as well as recent years, there are comparatively few publications. In some analyses, these years are therefore excluded. The exact number of publications per year can also be seen in Figure 4.7.

4.3.1. Organizations

Before deep-diving into the technicalities of Foundation Models, understanding the developing research institutions provides a broad overview of the leading players in Foundation Model research.

For the charts in this section, the extracted organizations have been cleaned, grouped, and then automatically enriched with additional characteristics such as country of origin or type (i.e., academic research institute, or private sector lab) using AI.

The development of Foundation Models takes place predominantly in North America. The extracted dataset shows that, of the most relevant Foundation Model publications according to EpochAI, in almost every year 40% of the publications were authored or published by researchers from American institutions.

In the mid-2010s, European and Chinese institutions published similar amounts of new Foundation Models, followed by some publications from APAC countries. How-

Data Point	Type	% of N/A	Comment / Examples
Model name	Text	1%	
Domain	Categorical	1%	Medicine, Robotics, etc.
Organization	Text	25%	
Authors	Text	11%	
Publication date	*Date*	*82%*	*Extracted from Arxiv instead*
Parameters	Numeric	39%	
Training dataset	Text	11%	
Training dataset size	Numeric	58%	
Epochs	Numeric	74%	
Training time	Numeric	83%	
Training hardware	Text	52%	
Hardware quantity	Numeric	64%	
Hardware utilization	Numeric	98%	
Base model	Text	58%	
Batch size	Numeric	50%	
Input modality	Categorical	1%	Text, Image, Audio, etc.
Output modality	Categorical	6%	Text, Image, Audio, etc.
Architecture	Categorical	1%	Vision Transformer, LSTM, etc.
Task	Text	1%	Image gen, classification, etc.
Robot Models only			
Robot Type	Categorical	23%	Humanoid, Industry arm, etc.
Sensor Modalities	Categorical	21%	RGB Camera, Lidar, tactile, etc.
Control Type	Categorical	28%	Low-level, planning, E2E, etc.
Environment Types	Categorical	17%	Simulated, controlled, outdoor

Table 4.2.: Overview of dataset fields, their types, percentage of missing values (N/A), and example values.

ever, in the 2020s, the share of Chinese models has vastly increased and is becoming more similar to the share of American publications. The research output from countries in the Middle East has noticeably increased in recent years.

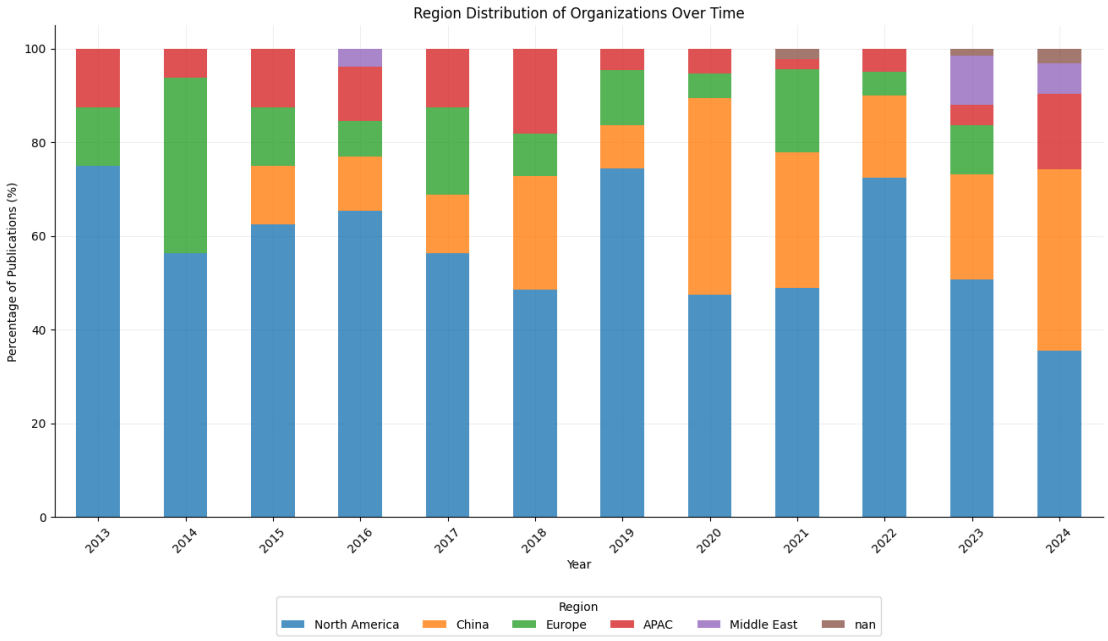


Figure 4.8.: Distribution of model development by geographical regions over time (see full-size version in Appendix A.2)

Looking more detailed at the top 10 publishing countries, underlines this trend. Each year, China and the United States dominate the number of published Foundational Models, with the other countries varying between the years. In Europe, France, Germany, and the United Kingdom produce the most research output, while in the APAC countries (excluding China) South Korea, Japan, and Singapore dominate.

The drill-down on the organizational level shows expected results: The American "Big-Tech" companies *Alphabet* ($n = 573$), *Meta* ($n = 228$), *Microsoft* ($n = 72$), and *OpenAI* ($n = 57$) dominate the number of publications, followed by a mix of American and Chinese universities, in addition to some further companies from the US and China.

It is important for this chart to remember that all model variances from a model family were counted as individual models. Hence, the data can be biased towards organizations publishing multiple smaller versions in addition to their main models.

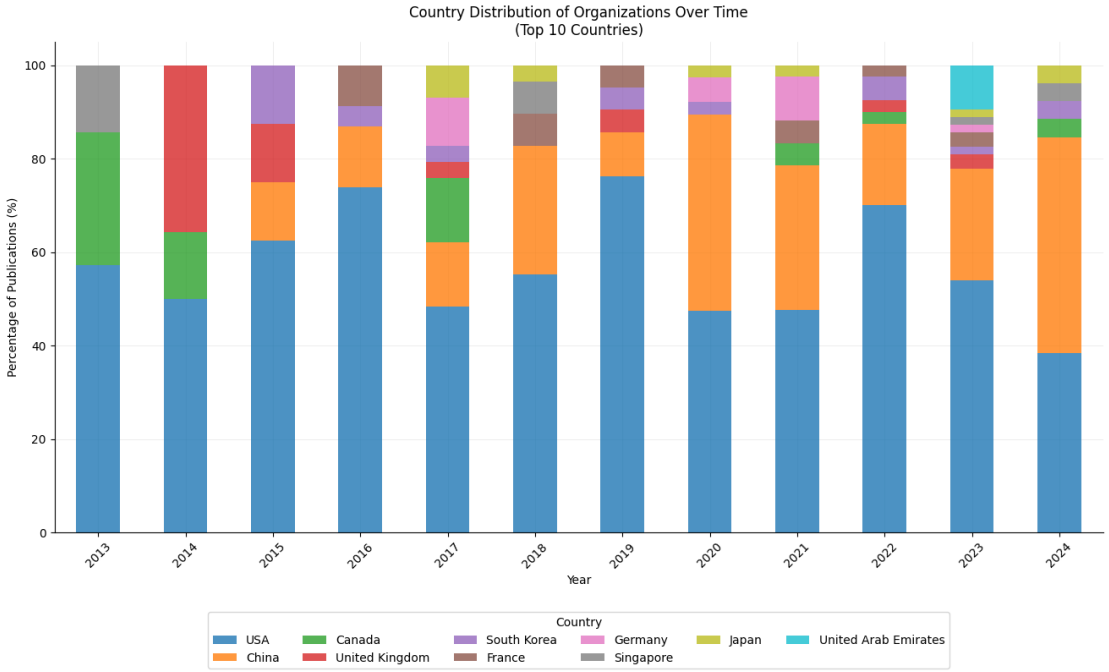


Figure 4.9.: Distribution of model development by country over time (Top 10 countries per year) (see full-size version in Appendix A.3)

4.3.2. Training Datasets

Besides parameter scaling, the type and size of datasets used for training are relevant for predicting the capabilities of Foundation Models according to the Scaling Laws.

For Figure 4.11, the extracted training datasets have been grouped by publications, meaning if multiple model versions were introduced in one paper, which have potentially been trained on overlapping datasets, the dataset was only counted once.

The data shows that *ImageNet* ($n = 76$) and *CIFAR-10* ($n = 48$) are the most used datasets, before *Wikipedia* ($n = 32$), and *Penn Treebank* ($n = 25$), with MNIST ($n = 22$) following as fifth.

In total, 1243 unique dataset names have been extracted. 74 dataset names have been extracted from more than 3 papers.

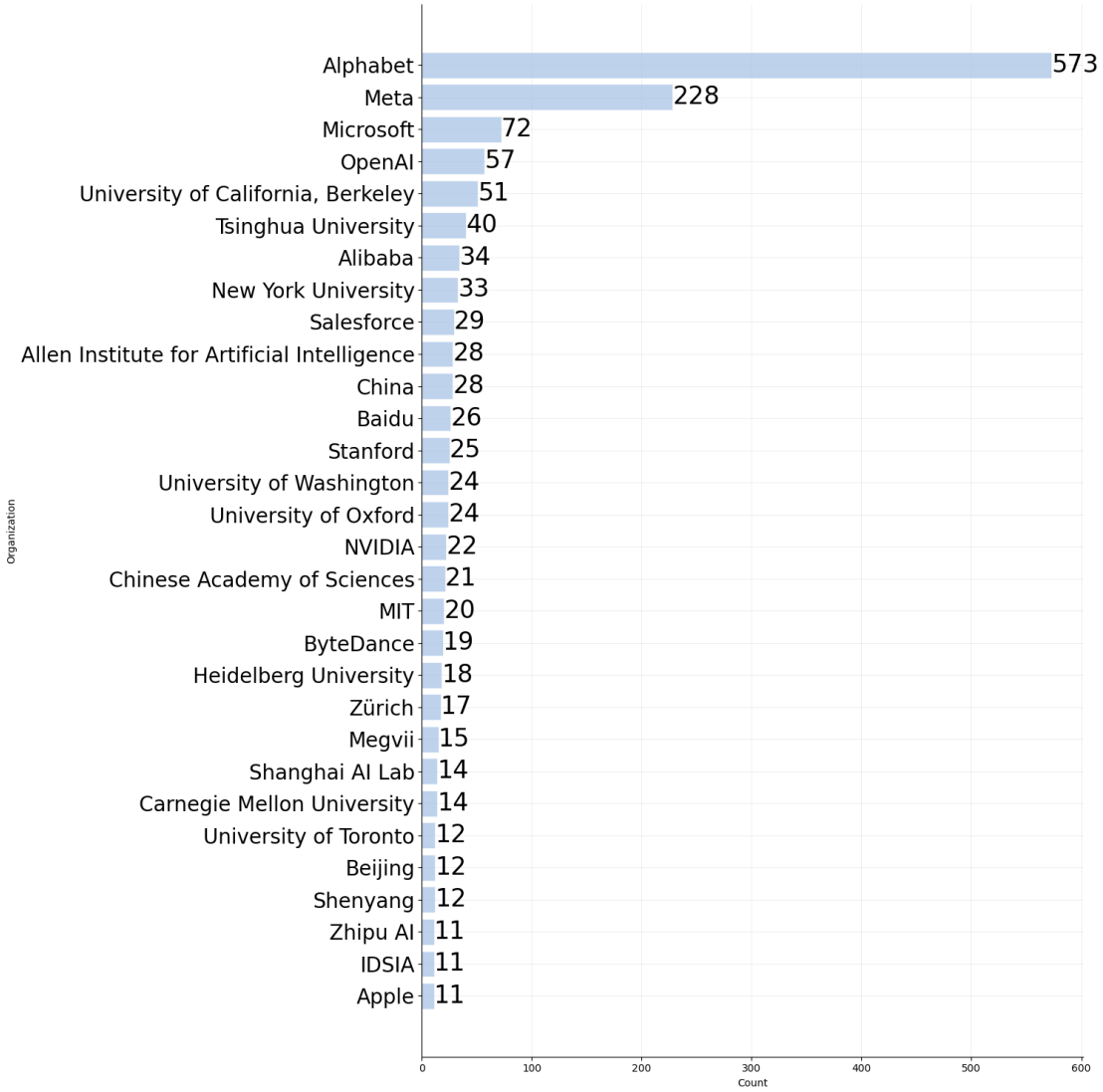


Figure 4.10.: Number of Foundation Models Publications of the 30 most active organizations (see full-size version in Appendix A.4)

4.3.3. Parameter Development

Figure 4.7 already provides a high-level overview of the exponential parameter count scaling, which, according to previously described Scaling Laws, enables capability enhancements of the analyzed Foundational Models.

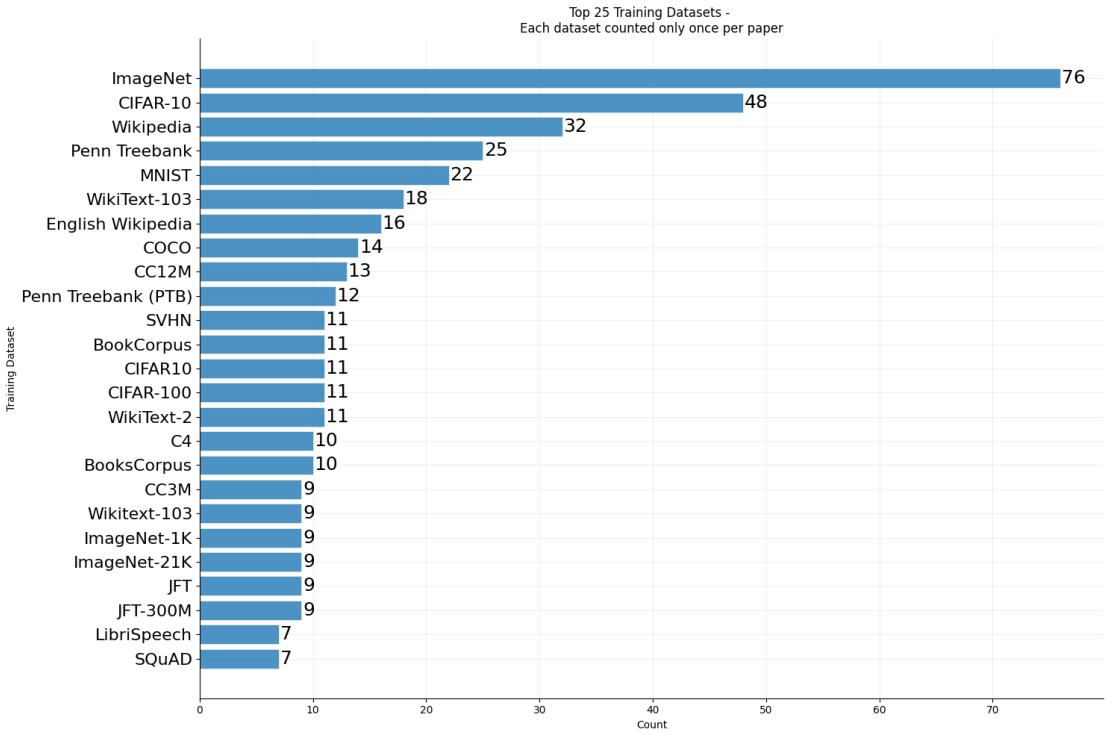


Figure 4.11.: Most used datasets for training Foundation Models (see full-size version in Appendix A.5)

Grouping the models in the extracted dataset by their underlying architecture reveals additional insights into model scale and growth.

Table 4.3 highlights the differences in size of the most frequently used model architectures. The values have been calculated with only the largest and most capable version of a modern family to focus on the state-of-the-art model development. However, the R^2 values are rather low, indicating irregular growth and sudden leaps.

In addition, the dataset has been broken down to better understand the adoption of model architectures over time. Figure 4.12 spotlights the adoption of the ten most used architectures on the parameter scaling graph.

Noticeably, with the emergence of Transformer-based architectures, almost all other architectures have lost their relevance and are only used by very few models, while in the 2010s multiple architectures were used without a clearly predominant one.

Architecture	Data Points	Avg Parameters	Annual Rate	R^2
Mixture of Expert	11	$4.24 \cdot 10^{11}$	0.180	0.097
Transformer (Decoder-only)	71	$8.06 \cdot 10^{10}$	0.198	0.087
Transformer (Encoder-Decoder)	40	$7.86 \cdot 10^{10}$	0.290	0.133
Multimodal Transformer	25	$4.17 \cdot 10^{10}$	0.394	0.206
Transformer (Encoder-only)	19	$1.17 \cdot 10^{10}$	0.293	0.222
CNN	26	$9.63 \cdot 10^{8}$	0.126	0.088
Hybrid CNN-Transformer	16	$9.41 \cdot 10^{8}$	0.225	0.273
ResNet	17	$3.03 \cdot 10^{8}$	0.382	0.475
LSTM	30	$7.96 \cdot 10^{7}$	0.105	0.116

Table 4.3.: Average size and growth rate of the largest models per family grouped by underlying architecture.

Since the introduction of the Transformer in 2017, the parameter counts have also risen significantly. The prior used *Long Short-Term Memory (LSTM)* models, *Convolutional Neural Networks (CNN)*, and *Recurrent Neural Networks (RNN)* show maximal sizes of roughly 10^8 parameters in the years until 2017 and only limited growth in the following years.

In contrast, even the smallest Transformer-based models have more than 10^8 parameters, with only downsized versions in a model family having fewer. On average, the largest model of a Transformer-based model family has over 10^{10} parameters – 100 times the size of the non-Transformer models.

Mixture of Experts models are on average the largest according to the extracted data. Within the Transformer-based architectures, there are also recognizable trends towards *Decoder-Only* and *Multimodal* Transformer.

In addition to the most frequently used architectures highlighted in both Table 4.3 and Figure 4.12, a total of 42 individual architectures have been extracted, with some being adaptations of higher-level architectures.

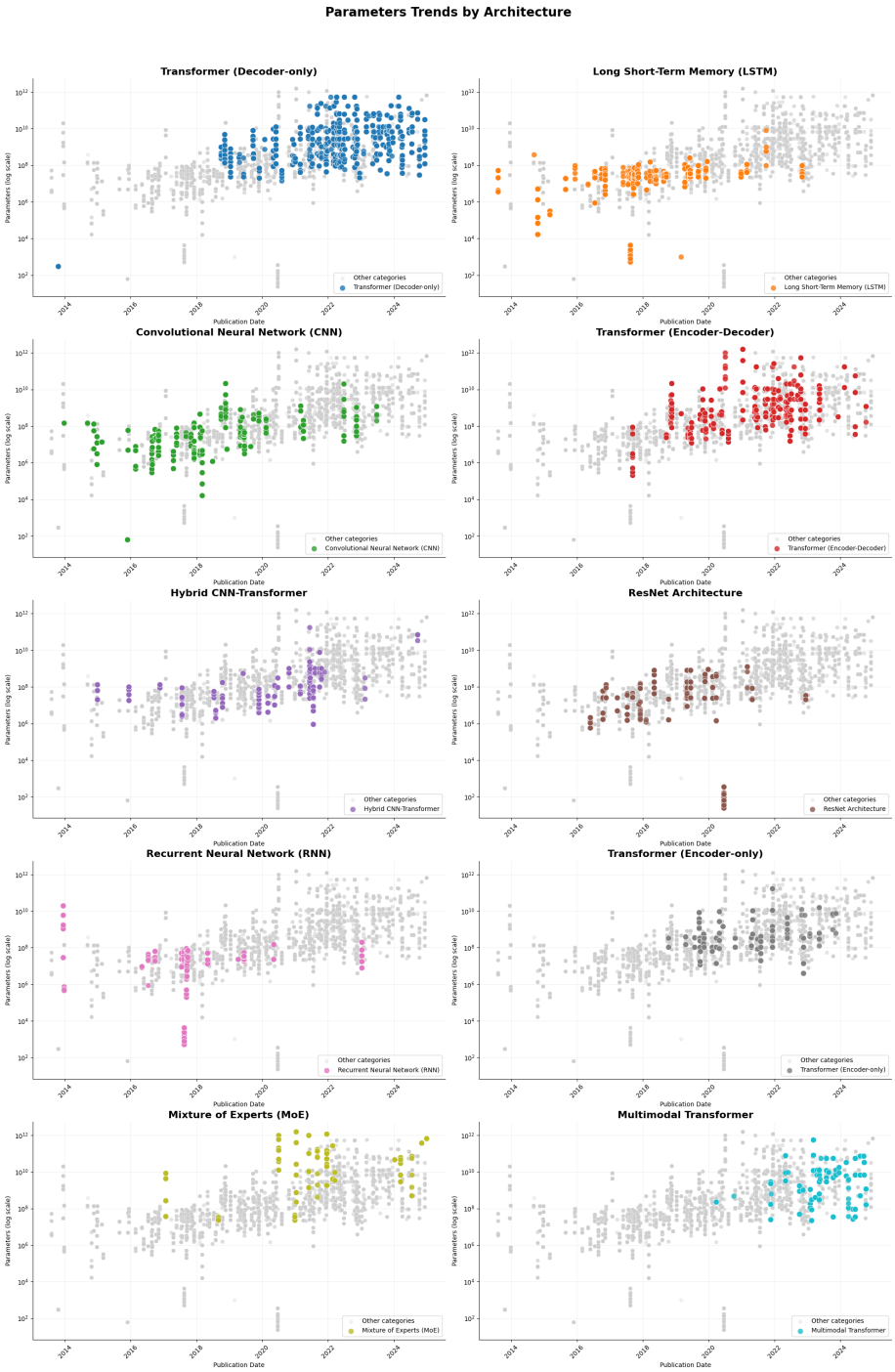


Figure 4.12.: Architecture popularity trends over time, with each panel spotlighting one architecture (colored)

4.4. Robotic Models

Looking in more detail at the data created from a dedicated extraction run on robotics papers allows to better understand the Foundation Model landscape in the Robotics domain.

Again, similar to the replicated EpochAI dataset, the robotics-specific dataset contains 23 data categories and is therefore too large to fully analyze in this thesis. Hence, only selected analyses are highlighted in this section.

The used search terms in the arXiv paper discovery were `(Robotics AND Foundation Model) OR (Robotics AND Foundational Model)`. These needed to be included in the publication's title or abstract for it to be further evaluated by the LLM-based filter.

The filter marked 223 publications as relevant, with publication dates between 2018 and July 2025. Importantly, there was no date limit during the paper search and filtering process, indicating no relevant papers prior to 2018 according to the developed filter.

Figure ?? visualizes the yearly distribution of publications and models. In the years 2018 to 2022, only 11 publications have been marked as relevant, corresponding to 24 models in said time-frame. For 2023 32 publications have been analyzed with a total of 87 models. For 2024 273 models have been extracted from 88 publications. For the first half of 2025 (Jan. to mid-July), around the same number of publications were analyzed ($n = 91$) and 278 models extracted.

4.4.1. Organizations

Analogous to the Epoch-based dataset, the publishing organizations have also been extracted, grouped, and counted with the same methodology for the robotics-focused publications.

The 223 analyzed publications were authored and co-authored by researchers from 175 unique main organizations. *Tsinghua University* in Peking published the most models ($n = 47$), with the *Shanghai AI Lab* ($n = 29$), *MIT* ($n = 25$), and *NVIDIA* ($n = 21$) following. Both *Peking University* and the *National University of Singapore* have published 20 models.

Together, these six organizations have been involved in roughly 20% of all published

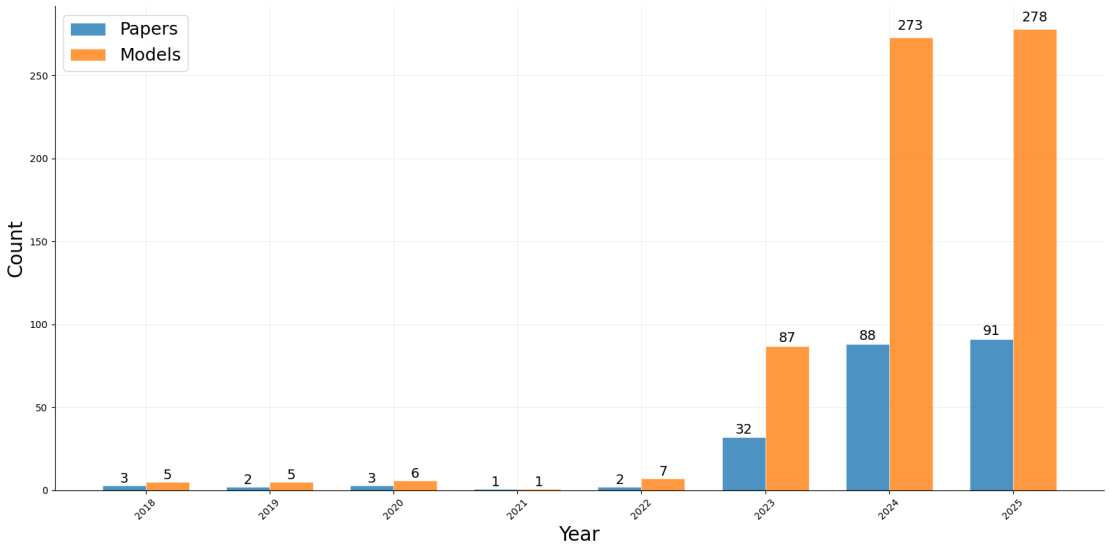


Figure 4.13.: Number of relevant Publications (Blue) and Extracted Models (Orange) per year for the Robotics domain (see full-size version in Appendix A.6)

models. A total of 56 organizations have been involved in the development of four or more models, and 106 have published more than one model.

Figure 4.14 shows the 15 organizations with the highest involvement in Foundational Model development for Robotics based on the published models.

Figure 4.15 provides additional information on the top publishing organizations in the Robotics field, by comparing the number of developed and published models to the number of publications. In the figure, 15 organizations with the most publications are shown.

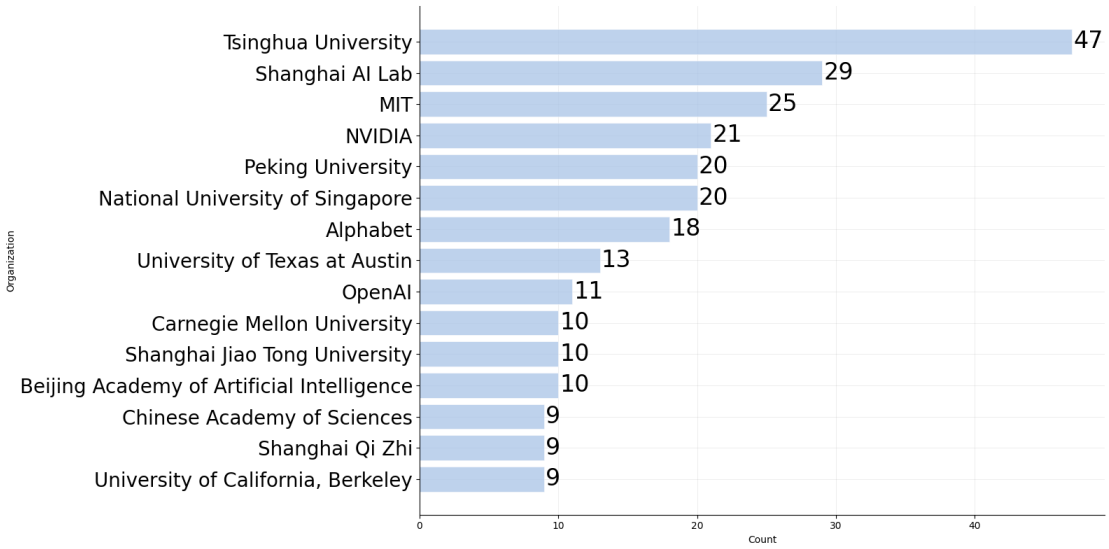


Figure 4.14.: Number of Foundation Models Publications of the 15 most active Organizations in the Robotics domain (see full-size version in Appendix A.7)

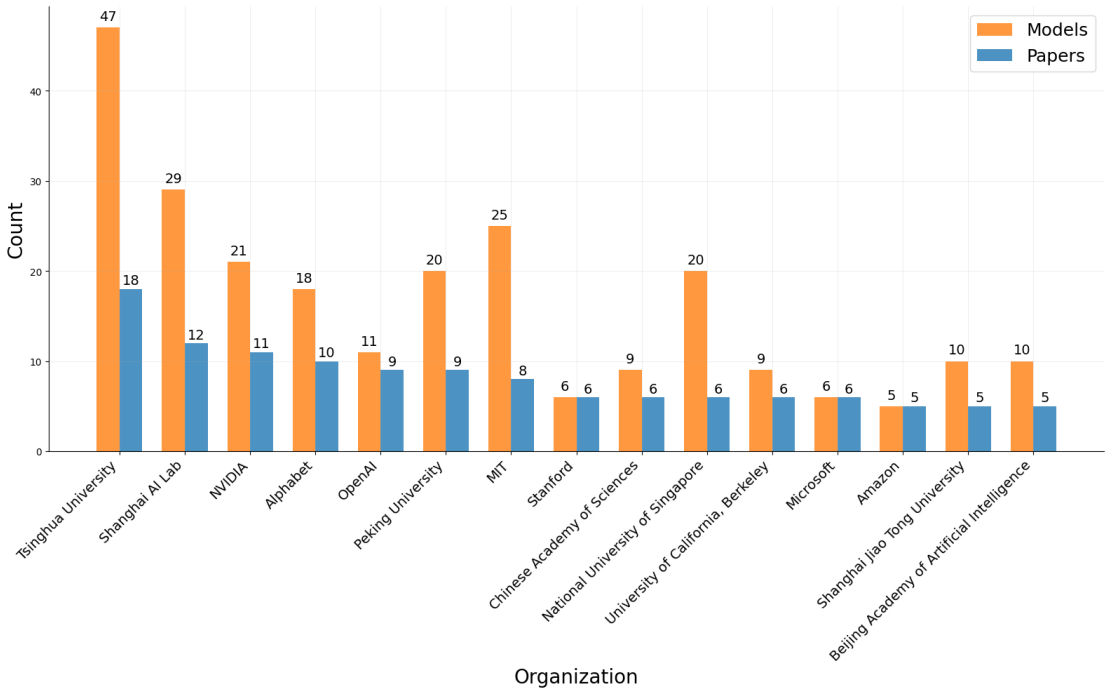


Figure 4.15.: Full-size version of the number of relevant Publications (Blue) and Extracted Models (Orange) in the Robotics domain by the Top 15 Organizations (see full-size version in Appendix A.8)

4.4.2. Robot Types

To gain an understanding of the investigated Foundation Models for Robotic applications, it is essential to understand which types of robots have been investigated.

The extracted data shows six different main robot types:

1. **Industrial Arms** ($n = 114$) are the most common underlying robot type in the dataset. These robots are usually used in factories to perform specified tasks, such as grabbing and placing objects or painting car bodies in automotive factories.
2. For **Mobile Robots** ($n = 86$), the second most Foundation Models have been developed. This area is rather broad, including models for self-driving cars, but also, for example, robots delivering plates to a table in a restaurant.
3. **Humanoid Robots** ($n = 26$) place third with roughly a third of the Mobile Robot model count. These robots have a human-like appearance, which helps them to learn complex tasks by mirroring real human behavior and movements.
4. **Quadruped Robots** ($n = 27$) follow closely with one model less. These robots are defined by having four "legs", often having a similar shape and size to a dog.
5. **Custom platforms** ($n = 13$) are very broad in scope and can be seen as a "Other" category. A custom platform can, for example, be a soft robot, which moves by varying air pressure.
6. Lastly, **Drones** ($n = 10$) form the least frequent robot type in the extracted dataset.

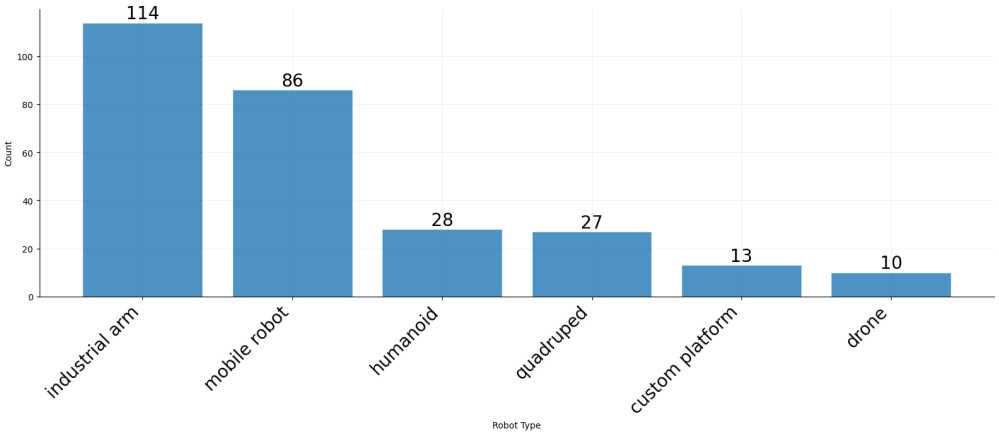


Figure 4.16.: Different Robot Types underlying the investigated Foundation Models

4.4.3. Control Types

There are multiple ways in which a Foundation Model can interact with a robot.

To better understand the integration depth, the control-level and therefore the interaction capabilities of the models have been categorized into 4 groups:

1. **High-level planning:** Focuses on strategic, long-term decision-making. Examples include selecting lanes, determining routes, or deciding which intersection to take. These planners do not output direct control commands but guide overall behavior.
2. **Mid-level planning:** Bridges high-level strategy and low-level control by generating actionable plans over a shorter horizon. Examples include lane-change maneuvers, merging into traffic, or producing trajectories for a robot arm.
3. **Low-level control:** Handles precise execution of actions at a high frequency, such as steering, acceleration, or joint torque commands. It ensures stability, accuracy, and responsiveness to real-time feedback.
4. **End-to-end policy:** Maps raw inputs (e.g., sensor data or images) directly to low-level control outputs without necessarily explicit intermediate planning steps. It combines perception, planning, and control in a single model.

Figure 4.17 visualizes the control capabilities of the analyzed model. Importantly, a model can be placed into multiple categories if the required planning and control capabilities are specified in its publication. This means that most *End-to-End* models are also counted in the other 3 categories.

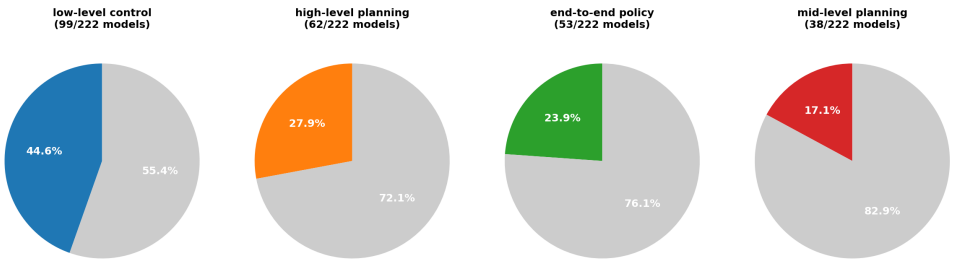


Figure 4.17.: Share of Control Types in Robotics Models

The data shows a fairly equal share of models with *end-to-end*, *high-level*, or *mid-level* planning capabilities. The share of models with direct *low-level* controls is roughly twice as high.

4.4.4. Control Types of Robot Types

Combining the previous two analyses enables understanding differences in the robot control types by the underlying robot type. Figure 4.18 details for each robot type the share of the four control types applied by the respective Foundation Models.

Noticeably, both the *Mobile Robots* and the *Drones* have a much higher share of Foundation Models performing high-level planning tasks.

In contrast, the Foundation Models for *Industrial Arms*, *Humanoids*, and *Quadrupeds* have a stronger focus on low-level controls.

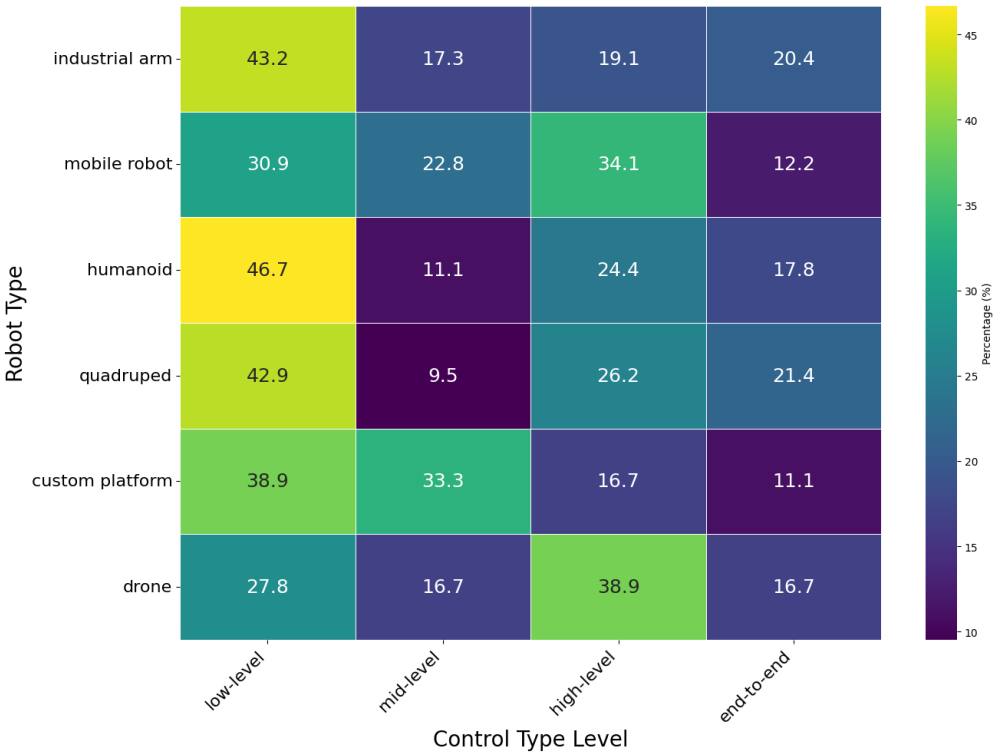


Figure 4.18.: Distribution of the Control Types for each Robot Type

4.5. Cross-Domain Comparison

Having the two complementary datasets enables to also perform cross-domain comparison of the robot-specific Foundation Models, to the broad overview of the Foundational Models from the EpochAI-based dataset.

4.5.1. Modality Comparison

Figure 4.19 is an example of such a comparison, visualizing the distribution of the number of modalities for each dataset.

Almost 80% of all models in the EpochAI-based dataset only process one modality. The median number of modalities is 1 and the average 1.25. Less than 5% of the models have three or more input modalities.

For the Robotics dataset, the average number of modalities is higher, with an average of 2.57 and a median of 2. Additionally, the modality count distribution for the Robotics Foundation Models has a long tail with over 35% having three or more, and over 15% of the models having four or more input and sensor modalities.

The models with apparently zero indicate incomplete results for some of the extractions.

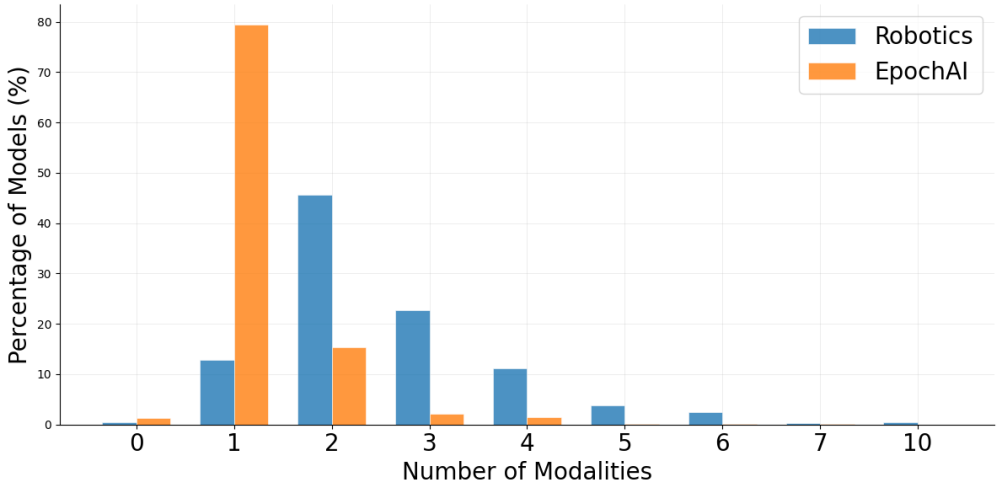


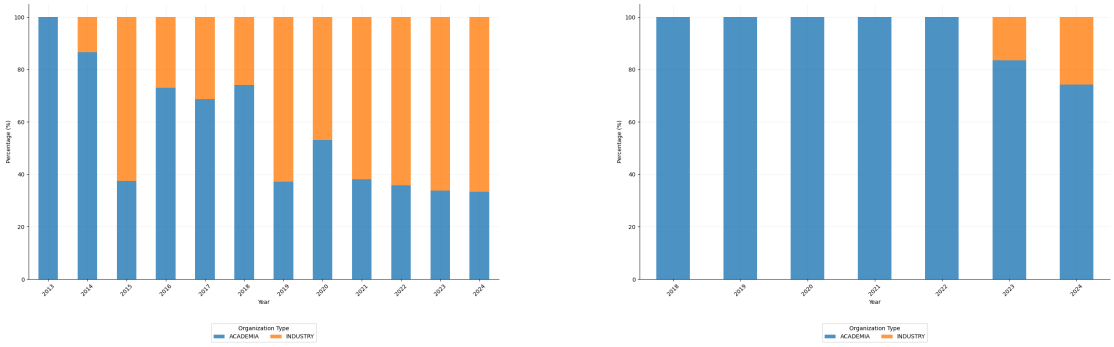
Figure 4.19.: Distribution of Number of Modalities for Robotics Models and All Models (excl. Robotic Models)

4.5.2. Research Institutions

Figure 4.20 compares the year distribution of research published by academic institutions with industry research labs. The percentages are visualized in individual charts for both the Epoch-based and robotics-focused extractions. The epoch values range from 2013 to 2024, while the robotics dataset only includes data for models starting in 2018.

Apart from two outliers in 2015 and 2019, where the share of industry publications was around 60%, and the initial year of 2013, where all investigated publications came from academia, the share of publications from industry has risen linearly from around 10% in 2014 to over 60% in 2021. Since then, the share of publications from industry compared to academia has only seen minor growth, with around 65% of publications coming from industry in 2024.

In the robotics domain, academia dominates research output. All investigated publications from 2018 to 2022 stem from academia. 2023 around 17% comes from industry, and in 2024, roughly 25%.



(a) Epoch-based Extraction

(b) Robotics-focused Extraction

Figure 4.20.: Share of yearly Publications by Academia (Blue) vs Industry (Orange) for the extracted Epoch-based and Robotics Datasets (see full-size version in Appendix A.9)

5. Discussion

The objective of this thesis consists of two individual tasks:

1. Developing a fully automated system, capable of sourcing relevant scientific publications, and extracting specified datapoints from them.
2. Replicating and extending the EpochAI dataset with Robotics models to create a more holistic Foundation Model landscape.

The following sections discuss the previously described results and reflect on the task performance.

5.1. Automated Literature Review Tool

In general, it is possible to develop a tool around Large Language Models, capable of extracting key characteristics from scientific publications and therefore helping to perform large-scale literature reviews and meta-studies.

Depending on the utilized benchmark, the developed extraction pipeline achieves an accuracy between 70% and 80% of fully correct values – a data quality sufficient for identifying development trends.

Looking at the developed pipeline in more detail, some additional characteristics are noteworthy:

The first stage of the pipeline includes the automatic sourcing and filtering of publications from arXiv.

With over 2.8 million publications, the arXiv repository is extraordinarily large, and keyword-based searches return a high number of potentially relevant publications. The developed filter has purposely been designed to be restrictive, leading to high precision (0.990) with rather low recall (0.426) in the used benchmark (see Fig 4.1). Although this behavior was acceptable for this thesis to limit the needed computing costs during the processing stage, and the F1-score benchmark additionally coming with some inaccuracies, a less restrictive filter with higher recall scores can be desirable to create

even more exhaustive literature reviews and meta-analysis on a larger underlying dataset.

When increasing the recall score, however, retaining high accuracy should remain of high priority to ensure relevant datapoints.

According to a study from 2020, the error rate of human reviewers during abstract screening in systemic literature reviews varies between 5.71% and 21.11%, depending on the study area and question type [Wan+20].

In the extraction stage, especially the LLM's response structure is noteworthy.

Unfortunately, even when explicitly prompted and instructed to only return a correctly formatted *JSON* result, the tested Large Language Models are not capable of doing so reliably.

Hence, a rather advanced custom parser needed to be developed to ensure processable results, based on the most common formatting issues. Nevertheless, some extraction results were not processable, and the information was therefore not available in the dataset.

The newest Large Language Models have started to address this, with additional API-endpoint parameters allowing to specify a structured output, which is supposed to ensure correct *JSON* formatting.

Besides the response formatting, the content of the responses also does not reliably follow the formatting instructions given in the prompt.

Especially for the analysis of the extracted values, this is of high impact. As shown in Figure 4.4, the share of not processable values was as high as almost 40%. Even in the best performing scenario, 20% of the extracted values were still unclear.

Assuming that the share of correct and incorrect extractions is equal to the processable data, being able to correctly process the unclear data would boost the accuracy of the extraction by almost 15% points to just under 90% of fully correctly extracted values, according to the Parameter-based benchmark.

In addition to ensuring the correct formatting of the extracted values, sophisticated and context-aware standardization of results could also significantly boost the data quality and correctness.

This can, for example, be seen in Figure 4.11, showing the histogram of the most used datasets in the extraction based on the publications aggregated by EpochAI.

The *Penn Treebank* dataset is most often extracted as *Penn Treebank*, but in an additional 12 cases as *Penn Treebank (PTB)*, although they refer to the same dataset.

In contrast, the *ImageNet-1K* and *ImageNet-21k* are two versions of the *ImageNet* dataset with varying sizes and are therefore correctly counted individually.

A sophisticated parser would need to be context aware to identify values that should be grouped and not counted individually – a task that was out of scope for the regular expression-based parser in the developed pipeline.

Lastly, Figure 4.7 highlights another challenge of extracting high-quality data from scientific publications.

More precisely, the groups of outlier values in 2017 and 2020. In this specific case, the 2020 values stem from a publication, in which the parameter count is provided in a table with a column titled *Parameters [M]*. [Che+20]

In cases like this, the pipeline sometimes does not correctly link the additional meta-information provided in column headers, labels, or captions to the content of the cells and only extracts the data from the table without the important information on the order of magnitude, leading to significantly smaller values.

A large-scale study performed to understand the validity of data extractions in pairwise meta-analysis found that in over 85% of the 291 investigated meta-analyses, the data extraction had errors [Xu+22]. This highlights that, in the vast majority of cases, even human reviewers are not able to perform a fully correct data extraction in literature reviews.

Overall, other extraction tools show slightly lower to similar performances on complex extractions compared to the one developed for this thesis.

A tool evaluated based on medical data in comparison to a human reviewer achieved between 65% and 100% accuracy depending on the data type [Dar+25].

Similar results are seen in a performance comparison of two AI tools against human reviewers. On highly standardized extraction fields, such as population characteristics of a medical trial and categorical values for study designs, the tools had an accuracy of 85% to even 100%. When extracting more complex data points and review-specific variables, the accuracy of correct extraction dropped to just over 70% [Hel+25].

Another study found their extraction results to be between 62% and 72% consistent with the human extraction [SJZ25].

5.2. Foundation Model Landscape

As stated multiple times in Chapter 4, the extracted dataset is broad in scope and allows for many analyses, deep dives, and insights. This section details and discusses

some selected results.

The decisive difference between the dataset manually aggregated by EpochAI and the extraction based on the corresponding publications becomes nicely visible in Figure 4.7:

The EpochAI dataset tracks, potentially due to the associated effort, only the largest version in a model family. The extracted dataset, however, can provide a much more detailed picture of the Foundation Model landscape by also containing information on the smaller versions in a model family.

Interestingly, although the extracted dataset contains many more models, with only smaller parameter sizes than the main models of a publication, the growth trends calculated from the datasets are almost identical. The only impact of the higher number of included models is a lower average parameter count per year.

This indicates that with the scaling of the main models, the smaller versions scale accordingly and do not remain at equal sizes over the years. In coherence with the discussed scaling laws, the developing organizations also need to increase the parameter count of the smaller model versions to provide additional capabilities when introducing a new model family.

The organizational trend with the predominance of large, American technology companies, highlighted in Figure 4.10, does not surprise, as these are the players having sufficient resources in terms of compute infrastructure (i.e., enormous datacenter with the newest, most efficient chips), liquidity to finance the operations and energy costs, and access to valuable exclusive training. Research by EpochAI sees historic costs of training growth by 2 to 3% per year, and expects that by 2027 the development costs for a frontier model surpasses 1 billion dollars [Cot+25].

Interestingly, the research output from countries in the Middle East has increased noticeably in recent years, which could be related to increased investments into innovative technologies such as Artificial Intelligence by countries like Saudi-Arabia or the United Arab Emirates to decrease their dependence on fossil resources, including oil, and diversify into new, more future-proof sectors. In 2019, the then newly established *Saudi Data and Artifical Intelligence Authority* set its vision to position Saudi Arabia as a global leader in the elite league of data-driven economies [Mem+21]. Backed by enormous investment from their sovereign wealth fund into data centers and research institutions, the country is successfully implementing this vision and gaining relevance according to the extracted dataset.

Figure 4.11 shows that *ImageNet* and *CIFAR-10* are the most used datasets, with MNIST following as fifth.

This is especially interesting as these datasets contain annotated images and are

usually used for computer vision or image-generating models.

Their high ranking surprises, as most of the investigated Foundation Models are not for vision or image tasks, but text-based ones. Besides potential inaccuracies of the extracted data, this could be caused by a bias in the models.

If text-based models do not share the used training data as frequently in their belonging publications, they do not get extracted and hence appear fewer, compared to the relatively standardized and named training datasets for vision tasks.

There are multiple reasons why the underlying datasets of Language models may not be shared as often. On the one hand, this could simply be caused by fewer available named datasets, leading to unstructured text scraping from various sources. On the other hand, this could also indicate the conscious decision not to state the used training data, as some parts, for example, were used illegally, infringing copyright and intellectual property regulations. The later is not unlikely, as recently multiple companies have been sued and partially agreed to settlements related to copyright infringements during the AI training [Sca25] [Kni25a] [Kni25b] [Pre25].

Additionally, it is notable that the dataset analysis has an extremely long tail, with over 1150 datasets being mentioned by less than three publications. Under the assumption that Foundation Model developers have access to generally similar datasets and try to maximize the training dataset size (cf. Scaling Laws), this is surprising, as most datasets should be used frequently. One potential cause for this could be vague data descriptions and the use of datasets with non-standardized names. The second case, again, seems unlikely, as existing datasets usually have a specific name, simply to be referenced and findable. Potentially, a context-aware parser could also group datasets with slightly varying extracted names to shorten the long tail.

Figure 4.12 showcases multiple complex relationships between model architectures, parameter sizes, and research trends. As stated, *Mixture of Experts* models are, on average, the largest according to the extracted data. This is plausible since such architectures are composed of multiple specialized sub-models, or *experts*, that are trained in parallel. A separate gating network decides which experts to activate for a given input, meaning that only a fraction of the overall model is used during inference. This design allows *Mixture of Experts* models to scale to very large parameter counts while maintaining efficiency, as computation is distributed across experts rather than applied uniformly to the entire network.

The recognizable trend towards *Decoder-Only* and *Multimodal* Transformer within the Transformer-based architectures is also explainable:

Decoder-only models are predominantly employed for sequence generation tasks, where the model predicts the next token conditioned on all previously generated

tokens. This architecture is widely used in large language models for applications such as text completion, summarization, question answering, code generation, and other tasks that can be framed as sequence continuation.

Multimodal models are designed to process and integrate information from multiple data modalities, such as text, images, audio, and video. By learning representations that capture cross-modal relationships, these models enable tasks including image captioning, visual question answering, speech-to-text generation, and multimodal content understanding, providing richer and more context-aware outputs than single-modality models.

With the rise of LLM-Chatbots such as ChatGPT or Google Gemini, this trend seems reasonable, as *Decoder-only* and *Multimodal* models are needed to provide their desired capabilities, and most of the models are developed by the Chatbot providers (see Fig. 4.10).

Figure 4.18 visualizes the relationships between robot types and the model's integration depth. Noticeably, both the *Mobile Robots* and the *Drones* have a much higher share of Foundation Models performing high-level planning tasks.

This makes sense, as, following the definitions in Chapter 4, for these robot types, a high-quality strategic movement plan is necessary to perform the desired movements in the correct order and at the correct time, allowing them to move freely in space.

Additionally, these robots and the belonging models are often built upon existing infrastructure, such as regular cars, for example, Foundation Models for Self-Driving. Their movements can often be programmable using established technologies, allowing the Foundation Models to perform more high-level tasks.

Staying in the context of Self-Driving, cars have been able to accelerate, decelerate, and steer by themselves for a long time, with cruise-control and lane-assist technologies being broadly available. Recognizing the entire surrounding environment, including all other cars, pedestrians, and street signs, to correctly and safely plan the next turns, is, however, a much more complex task requiring Foundation Models.

In contrast, the Foundation Models for *Industrial Arms*, *Humanoids*, and *Quadrupeds* have a stronger focus on low-level controls.

This also seems reasonable, as precise and controlled movements are required for these robot types to successfully perform their desired tasks. To achieve this, the models need to ensure stable control over the robot's movable parts in all degrees of freedom, taking additional factors such as varying weight distributions into account, which require real-time adaptation of standardized movements.

6. Conclusion

6.1. Key Findings

The previously thoroughly described and discussed can be summarized in the following key findings:

1. **Automated Tool Performance**
   The developed pipeline can automate large-scale literature reviews with an extraction accuracy of $70\% - 80\%$ depending on the benchmark and its settings. Noteably, the chunk size during the extraction positively impacts the resulting data quality. The filter for identifying relevant papers was designed for high precision (0.990) at the cost of recall (0.426) to manage computational costs.

2. **Foundation Model Growth**
   The parameter size of Foundation Models is growing exponentially. Including smaller model versions from a model family does not significantly change the growth rate compared to the EpochAI dataset, which investigates only the largest model versions.

3. **Architectural Trends**
   The Transformer architecture has become the dominant paradigm since 2017, with a recognizable trend towards Decoder-Only and Multimodal models. Mixture of Experts models are, on average, the largest.

4. **Research Dominance**
   The development of Foundation Models is concentrated in large American technology companies. However, a significant increase in research output from China and, more recently, the Middle East is visible and noteworthy. European institutions play no major role.

5. **Robotics Domain Trends**
   The field of Foundation Models for Robotics is rapidly accelerating, with a recent surge in publications and models. Academia is the primary driver of research, and robotics models, on average, integrate a higher number of modalities while having fewer parameters.

6. **Model Control Types in Robotics**
   A correlation exists between the robot type and its control level. Mobile Robots and Drones primarily use models for high-level planning, while Industrial Arms, Humanoids, and Quadrupeds rely more on models for low-level control.

6.2. Implications

The above summarized key findings lead to the following implications:

1. **Feasibility of Automation**
   The results demonstrate that Large Language Models can effectively automate the high-effort process of manual literature reviews, enabling researchers to conduct large-scale meta-analyses and identify trends much more efficiently.

2. **Impact of Model Size Growth**
   The exponential growth in model parameters, as discussed in the context of Scaling Laws, directly implies a corresponding exponential increase in computational resources and energy required for model development and training.

3. **Strategic Investments**
   The data on geographic trends highlights how strategic investments in AI, particular in nations like Saudi Arabia and the UAE, are successfully impacting the global research landscape.

4. **Bias in Data Reporting**
   The prevalence of image datasets in the training data analysis implies a significant bias in how researcher report their training data, particularly for language models, which could be related to intellectual property returns.

6.3. Limitations

As partiality already stated and discussed, the extraction pipeline and its results show some limitations:

1. **Ground-truth Reliability**
   As stated and shown by other research, over 85% of meta-analyses performed by researchers manually extracting data contained errors in some of the extraction. The same is likely true for the EpochAI dataset used as the benchmark during the pipeline evaluation. Hence, the actual accuracy of the developed pipeline might potentially slightly differ.

2. **LLM Reliability**
   The models used did not reliably follow prompt instructions for structured output, making a complex, custom-built parser to process the results necessary. This led to a significant percentage of "unclear" or unprocessable values.

3. **Filter Trade-off**
   The filter's design, favoring high precision over recall, means the dataset is not fully exhaustive and may have missed some relevant publications. In addition, the search terms used were specifically focused on *Foundation(al) Models* and *Robotics*. Publications not including these keywords in their title and abstract were therefore never regarded.

4. **Extraction Challenges**
   The pipeline occasionally did not correctly interpret crucial metadata (e.g., units like "M" for million parameters), leading to inaccuracies in some extracted values.

5. **Data Standardization**
   The rule-based parser could not perform contextual standardization (e.g., grouping "Penn Treebank" and "Penn Treebank (PTB)"), resulting in a long tail of unique datasets impacting the result quality.

6.4. Future Work

Future work on the thesis and its objective could focus on two key areas. Firstly, the automated literature review pipeline can be enhanced, and secondly, the scope, extent, and domain of the data analysis can be extended.

6.4.1. Enhancing the Extraction Pipeline

The primary area of future work should involve a significant focus on improving the robustness and accuracy of the developed tool itself by addressing the current pipeline's limitations. This mainly includes the following improvement potentials:

1. **Robust Data Extraction**
   Develop a more sophisticated, context-aware parser to handle the unreliable and incorrectly formatted output from LLMs. This could, for example, include leveraging newer LLM APIs with structured output parameters.

2. **Data Contextualization**
   Enhance the pipeline's ability to interpret extraction results and integrate metadata

from a publication. This would allow it to correctly link data points in tables with their corresponding units (e.g., "M" for million), detect potentially incorrect values, and additionally standardize values that have multiple variants (e.g., "Penn Treebank" and "Penn Treebank (PTB)").

3. **Balanced Filtering**
   Adjust the LLM-based publication filter to prioritize a higher recall score. While this would increase computational costs, it would enable the creation of a more exhaustive dataset for meta-analysis, providing a more complete picture of the Foundation Model landscape.

6.4.2. Extending the Scope of the Analysis

The scope of the extracted data analyses can be extended both in breadth and depth to provide a more comprehensive overview of the Foundation Model landscape. In detail, this could involve the following topics:

1. **Holistic Analysis**
   Leverage the vast and broad scope of the existing dataset to perform additional, more detailed analyses beyond the ones presented in this thesis, such as further investigating the relationship between model architectures and respective tasks.

2. **New Application Domains**
   Apply the developed methodology to other emerging fields beyond robotics, such as genomics, materials science, or finance, to identify domain-specific trends in Foundation Model development and compare them to the general landscape.

3. **Multi-Source Data Aggregation** Integrate data from sources beyond arXiv, such as project websites, GitHub repositories, and model cards. This would help create a richer dataset, as some valuable information may not be detailed in the main publication.

6.5. Thesis Conclusion

This thesis successfully demonstrated the feasibility of developing an automated, LLM-based software tool for conducting large-scale literature reviews and meta-analyses.

By applying this tool, a comprehensive dataset was created that not only replicated but also extended the existing EpochAI dataset to include a detailed analysis of the robotics domain.

The analysis of this data provided key insights into the exponential growth and architectural shifts in Foundation Model development, while also highlighting unique trends in robotics, such as a higher reliance on multimodal inputs and a strong prevalence of academic-led research.

A. Full-size Figures

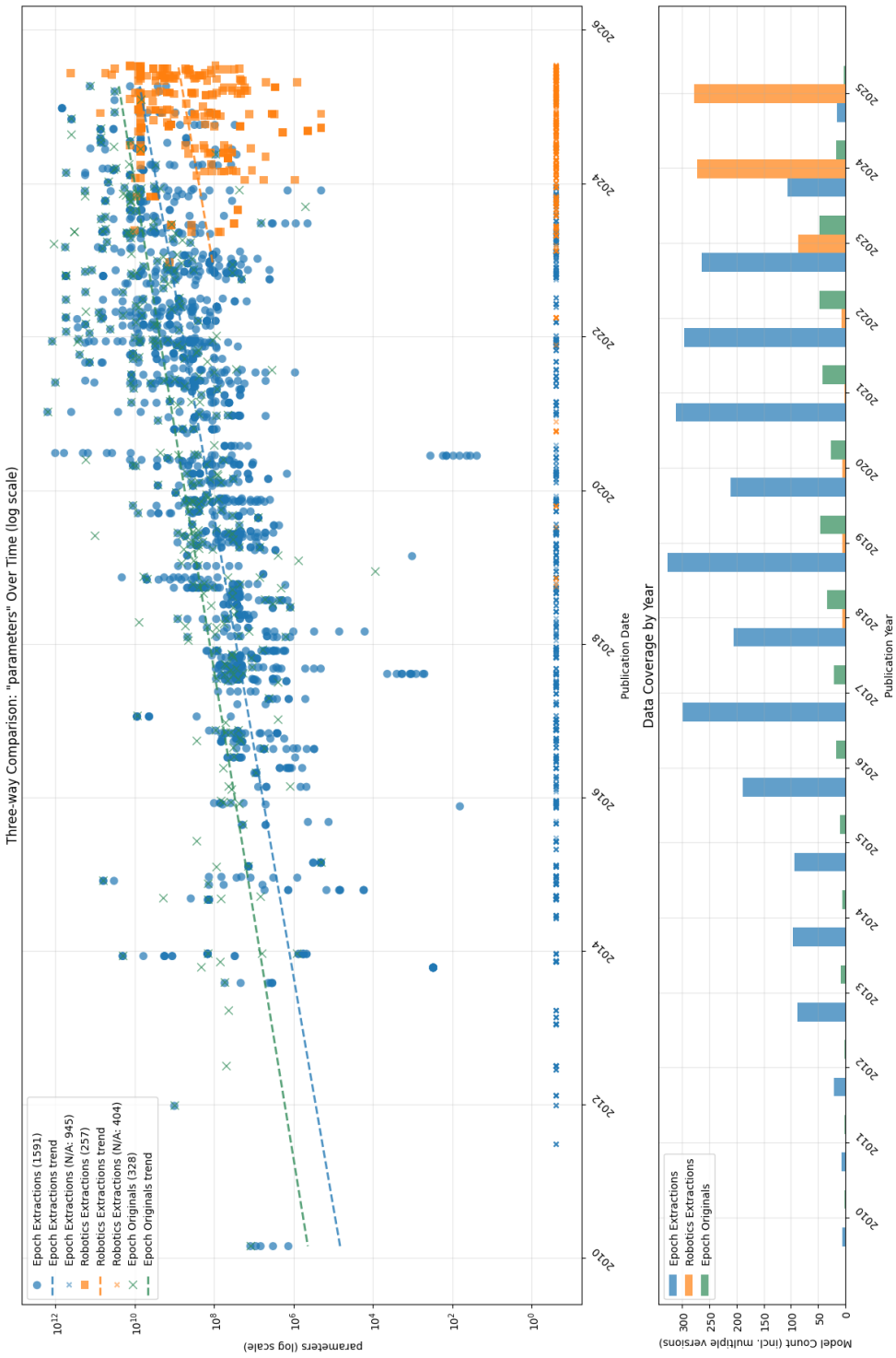


Figure A.1.: Full-size version of the comparison of the "parameter" values of the original EpochAI dataset, the replication created by the developed pipeline, and newly analyzed models on Robotics.

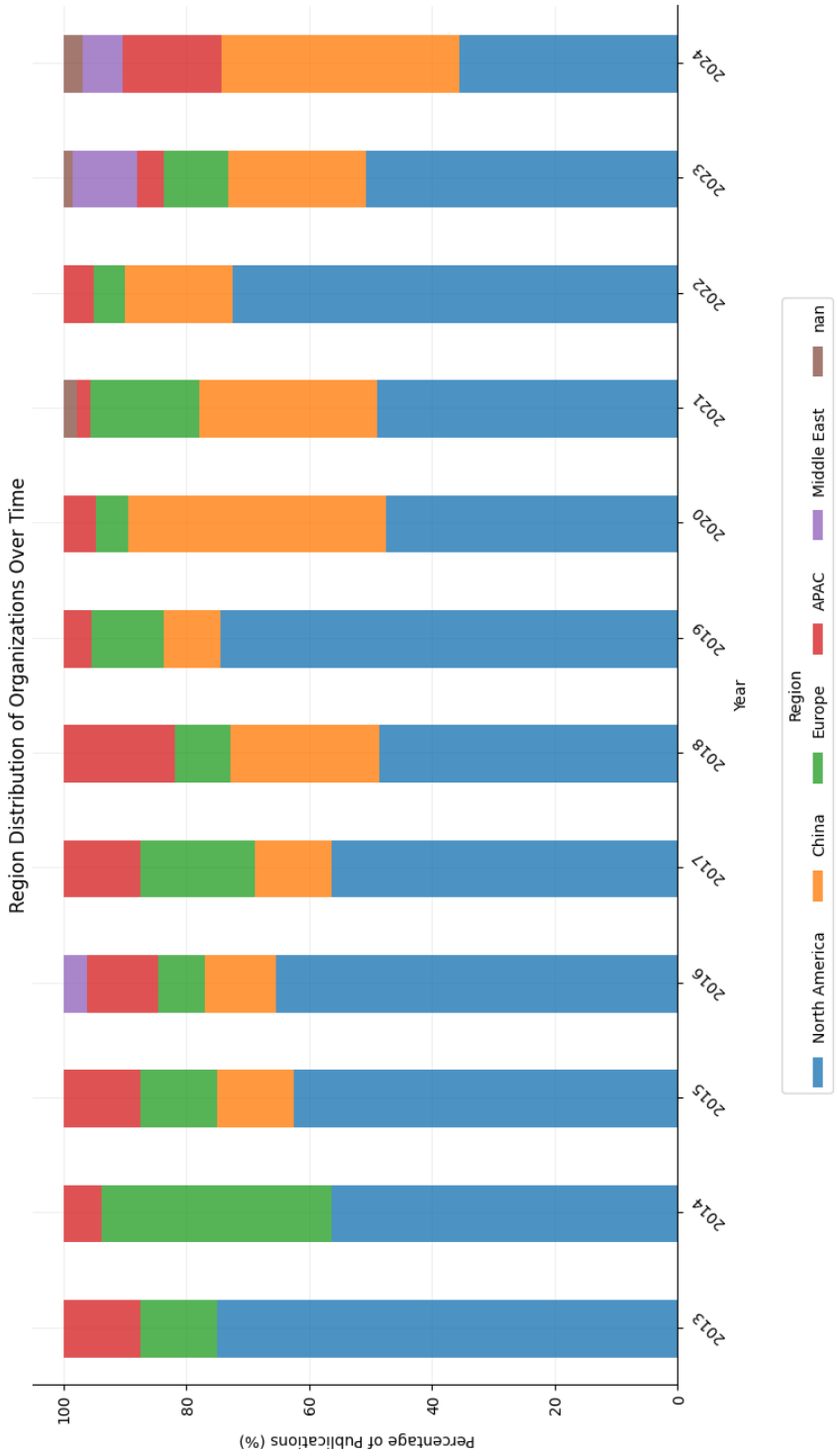


Figure A.2.: Full-size version of the distribution of model development by geographical regions over time

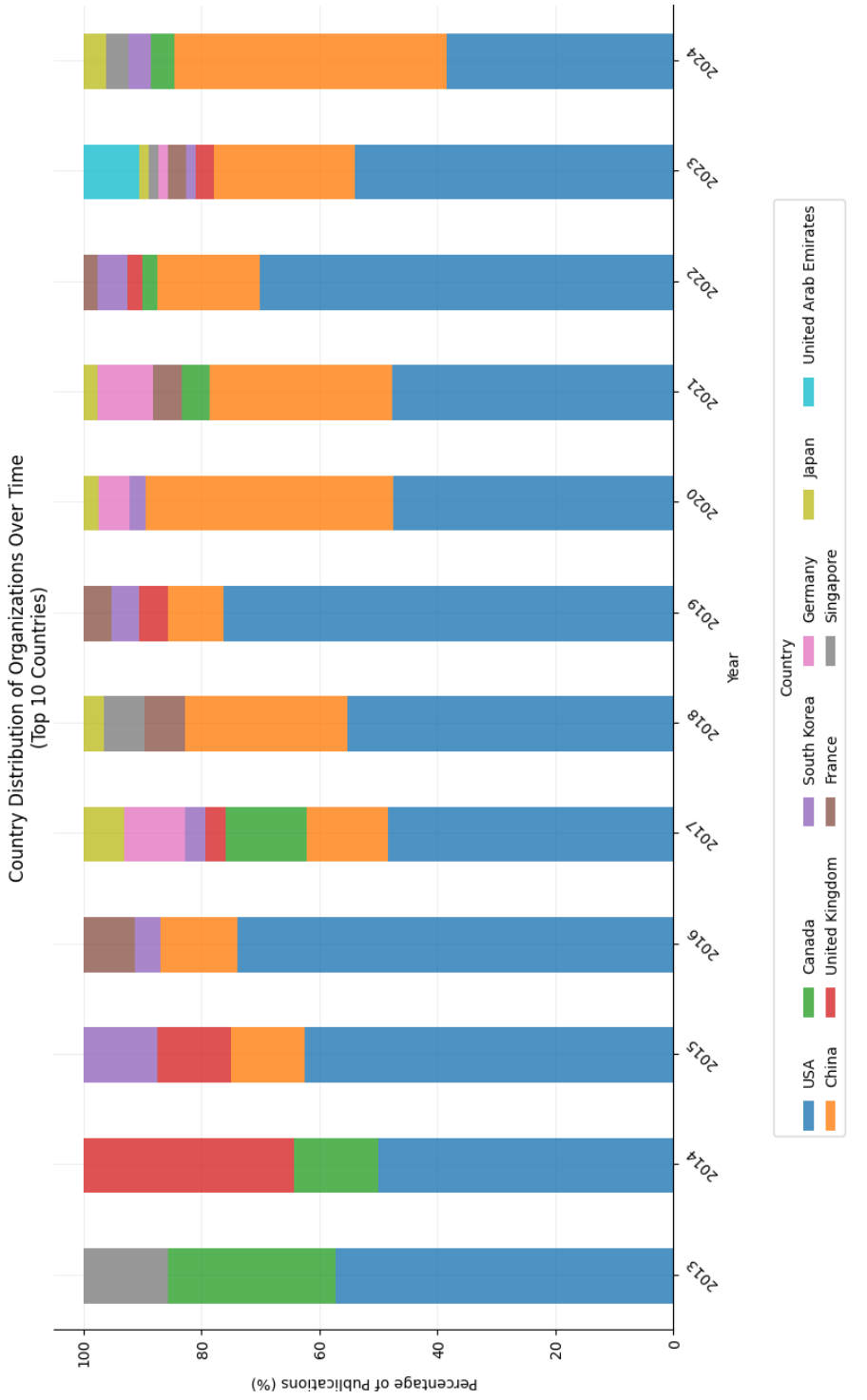


Figure A.3.: Full-size version of the Distribution of model development by country over time (Top 10 countries per year)

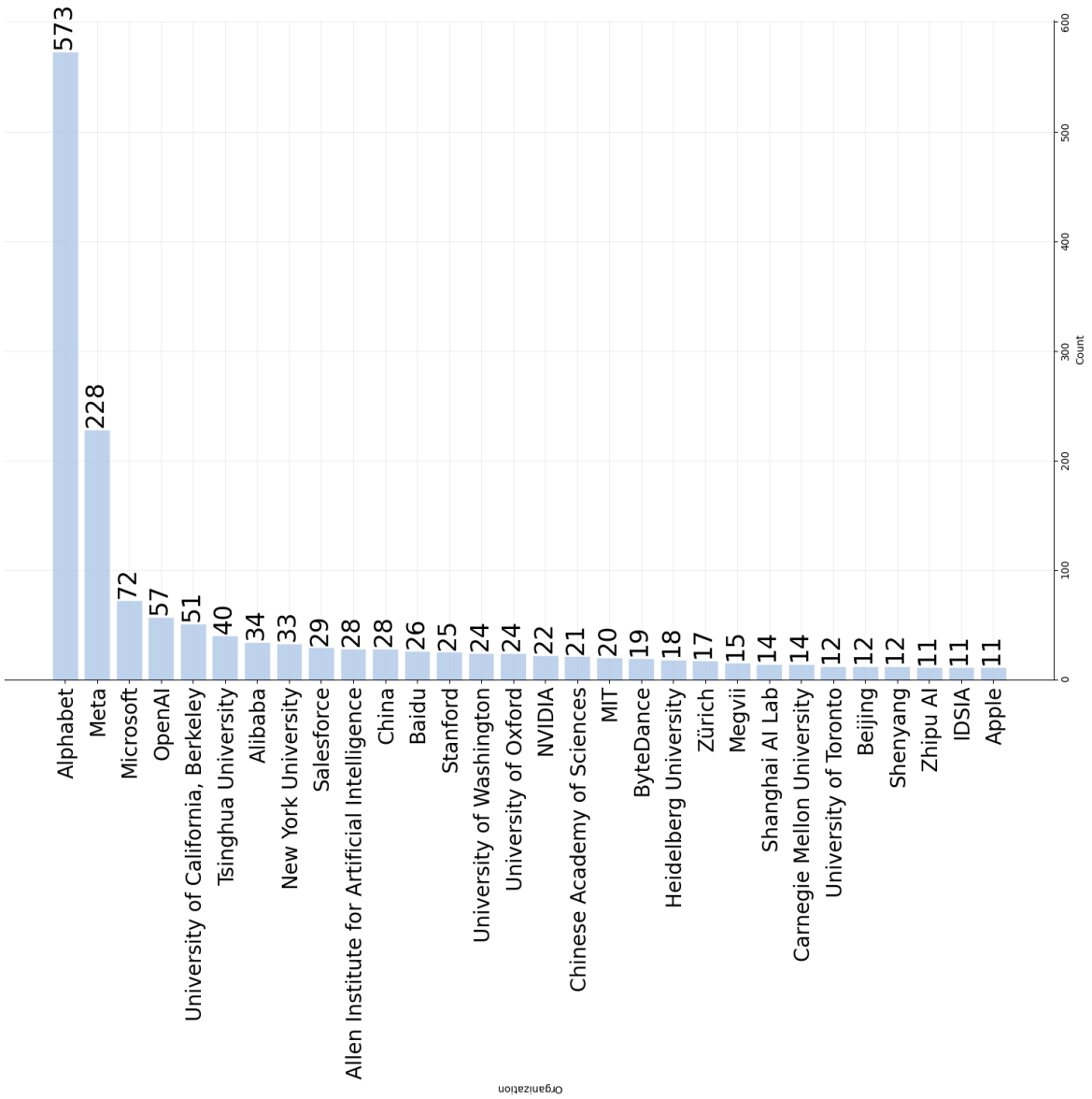


Figure A.4.: Full-size version of the number of Foundation Models Publications of the 30 most active Organizations

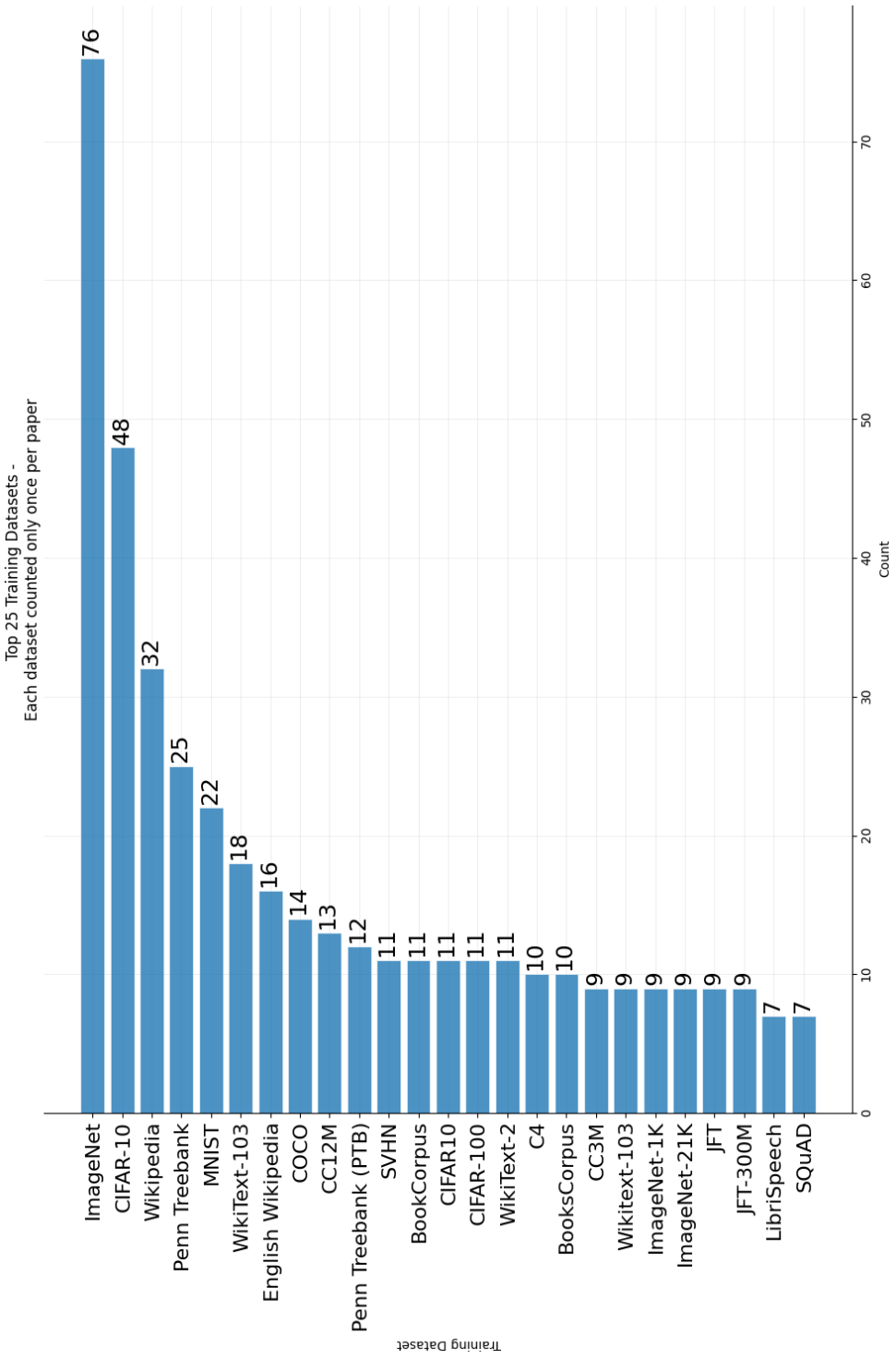


Figure A.5.: Full-size version of the most used datasets for training Foundation Models

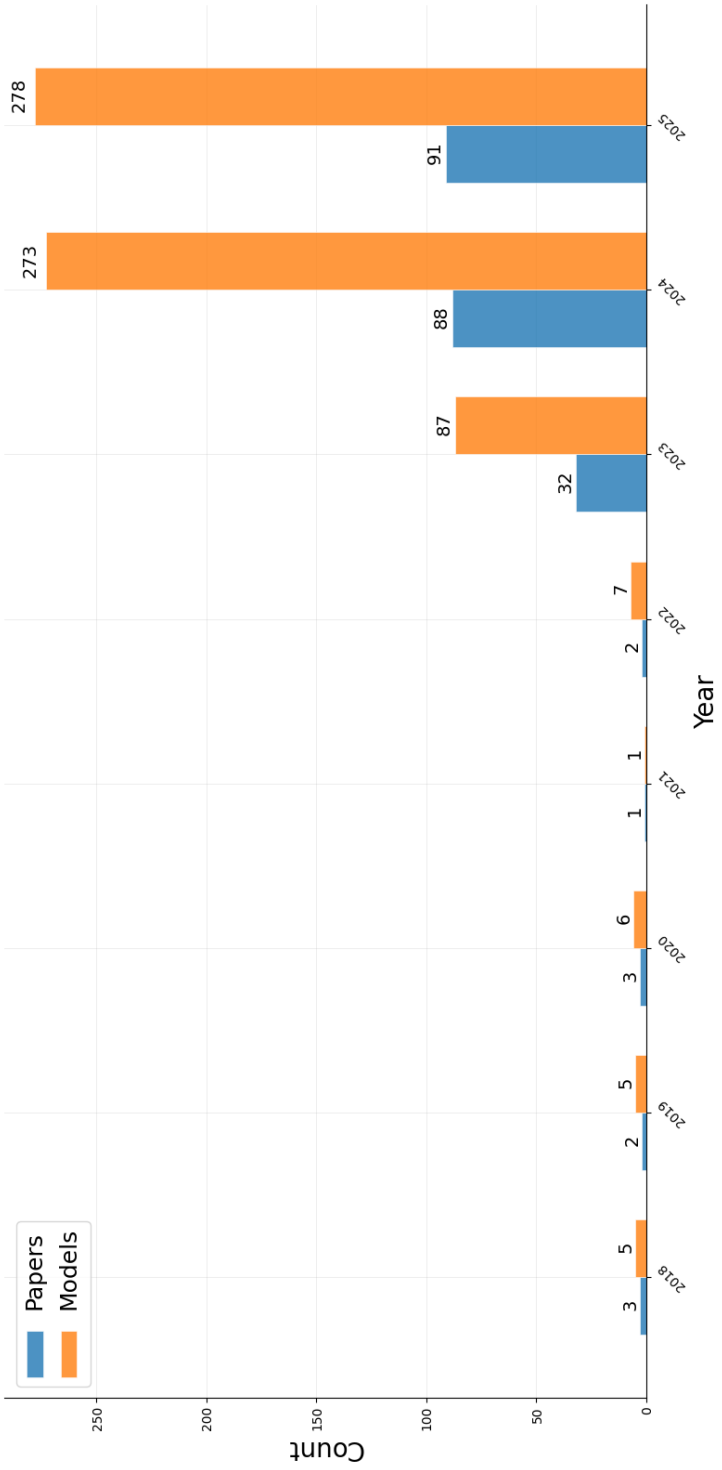


Figure A.6.: Full-size version of the number of relevant Publications (Blue) and Extracted Models (Orange) per year for the Robotics domain

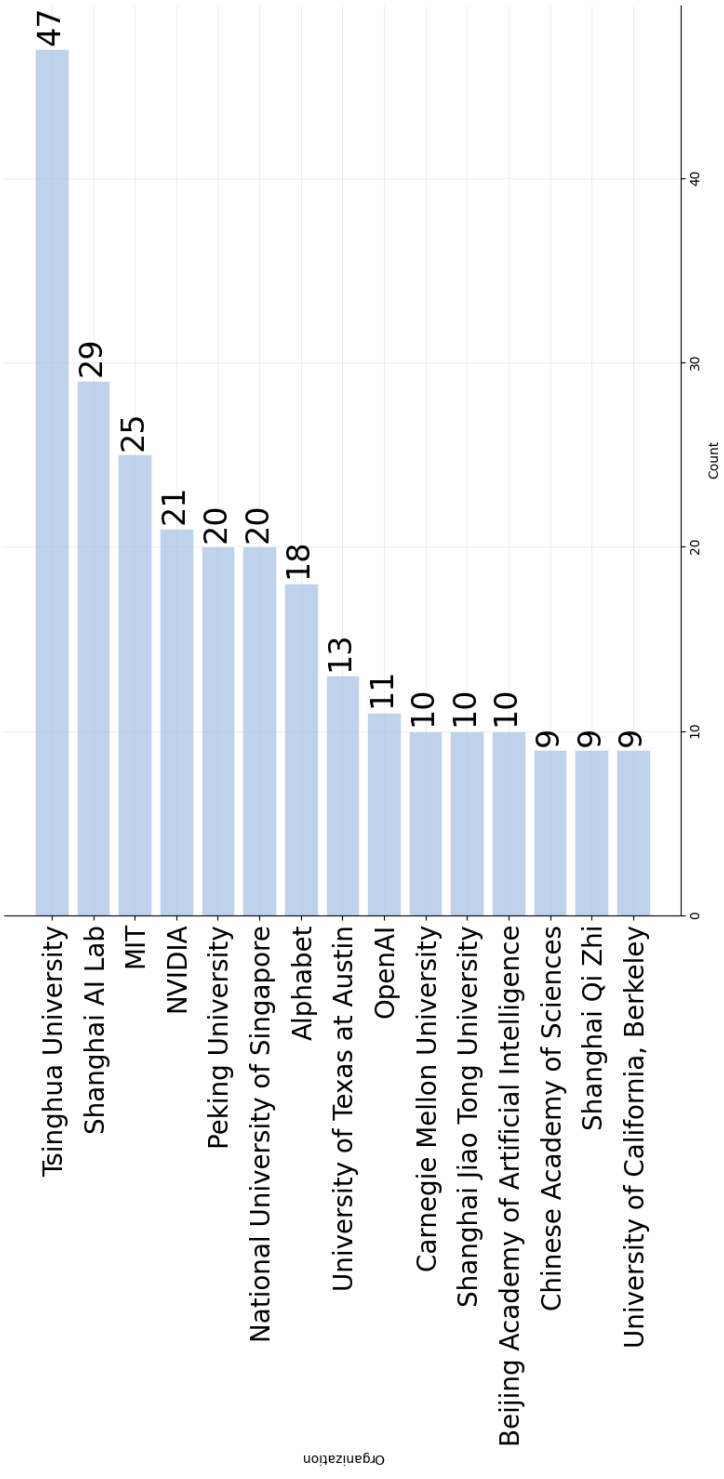


Figure A.7.: Full-size version of the number of Foundation Models Publications of the 15 most active Organizations in the Robotics domain

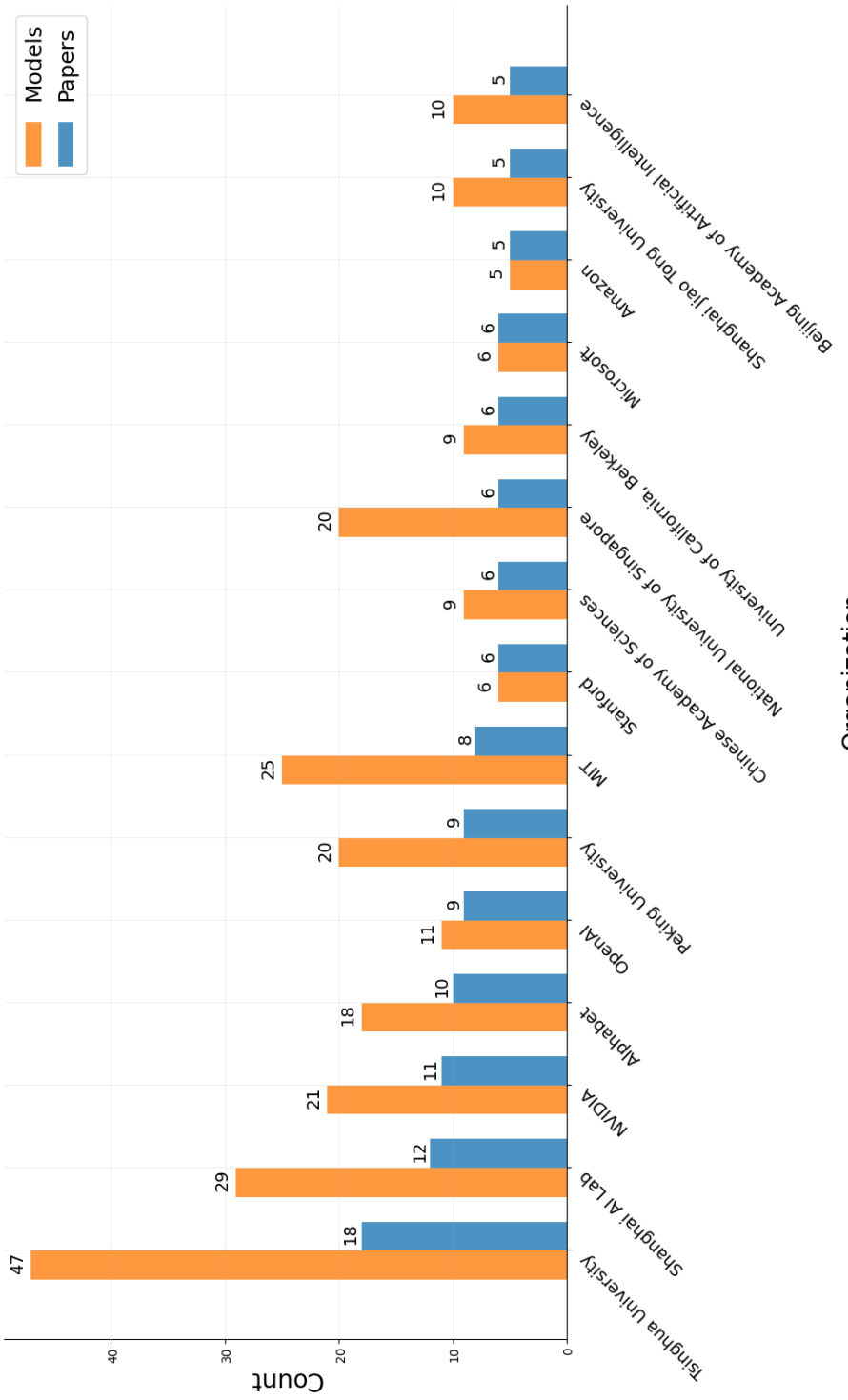
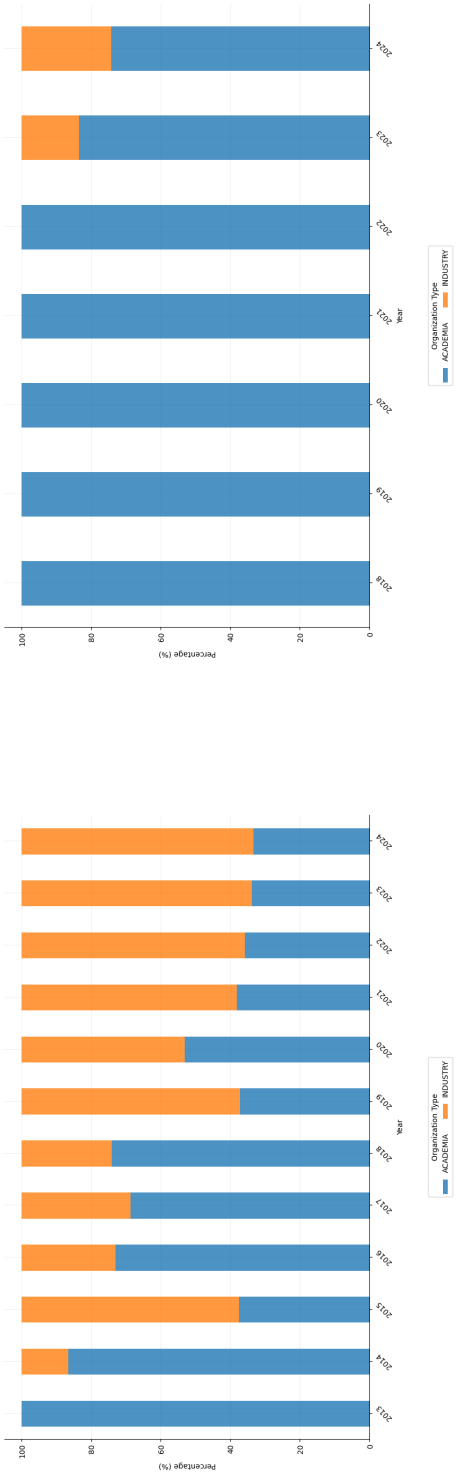


Figure A.8.: Full-size version of the number of relevant Publications (Blue) and Extracted Models (Orange) in the Robotics domain by the Top 15 Organizations



(a) Epoch-based Extraction

(b) Robotics-focused Extraction

Figure A.9.: Full-size version of the share of Publications by Academia (Blue) vs Industry (Orange) for the extracted Epoch-based and Robotics Datasets

List of Figures

List of Tables

Bibliography

[Ard+20] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. *Common Voice: A Massively-Multilingual Speech Corpus*. arXiv:1912.06670 [cs]. Mar. 2020. DOI: 10.48550/arXiv.1912.06670.

[Awa+23] M. Awais, M. Naseer, S. Khan, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, and F. S. Khan. *Foundational Models Defining a New Era in Vision: A Survey and Outlook*. arXiv:2307.13721 [cs]. July 2023. DOI: 10.48550/arXiv.2307.13721.

[Bel78] R. E. Bellman. *Artificial intelligence : can computers think?* Boyd & Fraser, 1978.

[Ber20] Berkeley School of Information. *What Is Machine Learning (ML)?* en-US. June 2020.

[Bha+24] N. Bhatt, N. Bhatt, P. Prajapati, V. Sorathiya, S. Alshathri, and W. El-Shafai. "A Data-Centric Approach to improve performance of deep learning models." en. In: *Scientific Reports* 14.1 (Sept. 2024). Publisher: Nature Publishing Group, p. 22329. ISSN: 2045-2322. DOI: 10.1038/s41598-024-73643-x.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. en. 2006.

[Bod+25] C. Bodnar, W. P. Bruinsma, A. Lucic, M. Stanley, A. Allen, J. Brandstetter, P. Garvan, M. Riechert, J. A. Weyn, H. Dong, J. K. Gupta, K. Thambiratnam, A. T. Archibald, C.-C. Wu, E. Heider, M. Welling, R. E. Turner, and P. Perdikaris. "A foundation model for the Earth system." en. In: *Nature* 641.8065 (May 2025). Publisher: Nature Publishing Group, pp. 1180–1187. ISSN: 1476-4687. DOI: 10.1038/s41586-025-09005-y.

[Bom+22] R. Bommasani, D. A. Hudson, E. Adeli, et al. *On the Opportunities and Risks of Foundation Models*. arXiv:2108.07258 [cs]. July 2022. DOI: 10.48550/arXiv.2108.07258.

[Bra24] A. Braun. "TUM AI Strategy." In: (2024).

[Bro+20] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language Models are Few-Shot Learners*. arXiv:2005.14165 [cs]. July 2020. DOI: 10.48550/arXiv.2005.14165.

[BTS24] A. Bauer, S. Trapp, and M. Stenger. *[2401.02524] Comprehensive Exploration of Synthetic Data Generation: A Survey*. 2024.

[Bun25] B. Buntz. *How xAI turned a factory shell into an AI 'Colossus' for Grok 3*. en-US. Feb. 2025.

[Cam+25] G. Camps-Valls, M.-Á. Fernández-Torres, K.-H. Cohrs, A. Höhl, A. Castelletti, A. Pacal, C. Robin, F. Martinuzzi, I. Papoutsis, I. Prapas, J. Pérez-Aracil, K. Weigel, M. Gonzalez-Calabuig, M. Reichstein, M. Rabel, M. Giuliani, M. D. Mahecha, O.-I. Popescu, O. J. Pellicer-Valero, S. Ouala, S. Salcedo-Sanz, S. Sippel, S. Kondylatos, T. Happé, and T. Williams. "Artificial intelligence for modeling and understanding extreme weather and climate events." en. In: *Nature Communications* 16.1 (Feb. 2025). Publisher: Nature Publishing Group, p. 1919. ISSN: 2041-1723. DOI: 10.1038/s41467-025-56573-8.

[Cha+14] E. Charniak, C. K. Riesbeck, D. V. McDermott, and J. R. Meehan. *Artificial Intelligence Programming*. 2nd ed. New York: Psychology Press, Jan. 2014. ISBN: 978-1-315-80225-1. DOI: 10.4324/9781315802251.

[Cha25] ChatPDF. *ChatPDF AI | Chat with any PDF | Free*. en. 2025.

[Che+20] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. *Big Self-Supervised Models are Strong Semi-Supervised Learners*. arXiv:2006.10029 [cs]. Oct. 2020. DOI: 10.48550/arXiv.2006.10029.

[Con25] Consensus. *Search - Consensus: AI Search Engine for Research*. 2025.

[Cot+25] B. Cottier, R. Rahman, L. Fattorini, N. Maslej, T. Besiroglu, and D. Owen. *The rising costs of training frontier AI models*. arXiv:2405.21015 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2405.21015.

[Dar+25] B. Daraqel, A. Owayda, H. Khan, D. Koletsi, and S. Mheissen. "Artificial intelligence as a tool for data extraction is not fully reliable compared to manual data extraction." In: *Journal of Dentistry* 160 (Sept. 2025), p. 105846. ISSN: 0300-5712. DOI: 10.1016/j.jdent.2025.105846.

[Dev+19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805 [cs] version: 2. May 2019. DOI: 10.48550/arXiv.1810.04805.

[Dod21] J. Dodge. *[2104.08758] Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus*. 2021.

[Epo25a] Epoch AI. *Data on AI Models*. en. 2025.

[Epo25b] Epoch AI. *Who we are - Epoch AI*. en. 2025.

[Eur] European Union. *EU Artificial Intelligence Act*. en-US.

[Hau89] J. Haugeland. *Artificial Intelligence: The Very Idea*. en. The MIT Press, Jan. 1989. ISBN: 978-0-262-29114-9. DOI: 10.7551/mitpress/1170.001.0001.

[Hel+25] T. Helms Andersen, T. M. Marcussen, A. D. Termannsen, T. W. H. Lawaetz, and O. Nørgaard. "Using Artificial Intelligence Tools as Second Reviewers for Data Extraction in Systematic Reviews: A Performance Comparison of Two AI Tools Against Human Reviewers." In: *Cochrane Evidence Synthesis and Methods* 3.4 (July 2025), e70036. ISSN: 2832-9023. DOI: 10.1002/cesm.70036.

[Hen+20] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, C. Hallacy, B. Mann, A. Radford, A. Ramesh, N. Ryder, D. M. Ziegler, J. Schulman, D. Amodei, and S. McCandlish. *Scaling Laws for Autoregressive Generative Modeling*. arXiv:2010.14701 [cs]. Nov. 2020. DOI: 10.48550/arXiv.2010.14701.

[Hes+17] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou. *Deep Learning Scaling is Predictable, Empirically*. arXiv:1712.00409 [cs]. Dec. 2017. DOI: 10.48550/arXiv.1712.00409.

[HJA20] J. Ho, A. Jain, and P. Abbeel. *Denoising Diffusion Probabilistic Models*. arXiv:2006.11239 [cs]. Dec. 2020. DOI: 10.48550/arXiv.2006.11239.

[Hof+22] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. v. d. Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. *Training Compute-Optimal Large Language Models*. arXiv:2203.15556 [cs]. Mar. 2022. DOI: 10.48550/arXiv.2203.15556.

[Hou23] T. W. House. *Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence*. en-US. Oct. 2023.

[Hu+24] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, H.-S. Fang, S. Zhao, S. Omidshafiei, D.-K. Kim, A.-a. Aghamohammadi, K. Sycara, M. Johnson-Roberson, D. Batra, X. Wang, S. Scherer, C. Wang, Z. Kira, F. Xia, and Y. Bisk. *Toward General-Purpose Robots via Foundation Models: A Survey and Meta-Analysis*. arXiv:2312.08782 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2312.08782.

[ISO] ISO. *Machine learning (ML): All there is to know*. en.

[IV+21] R. L. L. IV, I. Balažević, E. Wallace, F. Petroni, S. Singh, and S. Riedel. *Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models*. arXiv:2106.13353 [cs]. July 2021. DOI: 10.48550/arXiv.2106.13353.

[Jog+24] O. Joglekar, T. Lancewicki, S. Kozlovsky, V. Tchuiev, Z. Feldman, and D. D. Castro. *Towards Natural Language-Driven Assembly Using Foundation Models*. arXiv:2406.16093 [cs]. June 2024. DOI: 10.48550/arXiv.2406.16093.

[Jum+21] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. "Highly accurate protein structure prediction with AlphaFold." en. In: *Nature* 596.7873 (Aug. 2021). Publisher: Nature Publishing Group, pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2.

[Kap+20] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. *Scaling Laws for Neural Language Models*. arXiv:2001.08361 [cs]. Jan. 2020. DOI: 10.48550/arXiv.2001.08361.

[Kha+25] W. Khan, S. Leem, K. B. See, J. K. Wong, S. Zhang, and R. Fang. *A Comprehensive Survey of Foundation Models in Medicine*. arXiv:2406.10729 [cs]. Jan. 2025. DOI: 10.48550/arXiv.2406.10729.

[KM18] P. Korshunov and S. Marcel. *DeepFakes: a New Threat to Face Recognition? Assessment and Detection*. arXiv:1812.08685 [cs]. Dec. 2018. DOI: 10.48550/arXiv.1812.08685.

[KM25] I. Kumar and S. Manning. *Trends in Frontier AI Model Count: A Forecast to 2028*. arXiv:2504.16138 [cs] version: 1. Apr. 2025. DOI: 10.48550/arXiv.2504.16138.

[Kni25a] K. Knibbs. "Meta Secretly Trained Its AI on a Notorious Piracy Database, Newly Unredacted Court Docs Reveal." en-US. In: *Wired* (Jan. 2025). Section: tags. ISSN: 1059-1028.

[Kni25b] K. Knibbs. "Thomson Reuters Wins First Major AI Copyright Case in the US." en-US. In: *Wired* (Feb. 2025). Section: tags. ISSN: 1059-1028.

[Koc+22] D. Kocetkov, R. Li, L. B. Allal, J. Li, C. Mou, C. M. Ferrandis, Y. Jernite, M. Mitchell, S. Hughes, T. Wolf, D. Bahdanau, L. v. Werra, and H. d. Vries. *The Stack: 3 TB of permissively licensed source code*. arXiv:2211.15533 [cs]. Nov. 2022. DOI: 10.48550/arXiv.2211.15533.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012.

[Kur92] R. Kurzweil. *The Age of Intelligent Machines*. en. Cambridge, MA, USA: MIT Press, Jan. 1992. ISBN: 978-0-262-61079-7.

[Lan+20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. arXiv:1909.11942 [cs]. Feb. 2020. DOI: 10.48550/arXiv.1909.11942.

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning." en. In: *Nature* 521.7553 (May 2015). Publisher: Nature Publishing Group, pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

[Lew+21] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv:2005.11401 [cs]. Apr. 2021. DOI: 10.48550/arXiv.2005.11401.

[Lin+15] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. *Microsoft COCO: Common Objects in Context*. arXiv:1405.0312 [cs]. Feb. 2015. DOI: 10.48550/arXiv.1405.0312.

[Mas+25] N. Maslej, L. Fattorini, R. Perrault, Y. Gil, V. Parli, N. Kariuki, E. Capstick, A. Reuel, E. Brynjolfsson, J. Etchemendy, K. Ligett, T. Lyons, J. Manyika, J. C. Niebles, Y. Shoham, R. Wald, T. Walsh, A. Hamrah, L. Santarlasci, J. B. Lotufo, A. Rome, A. Shi, and S. Oak. *Artificial Intelligence Index Report 2025*. arXiv:2504.07139 [cs]. July 2025. DOI: 10.48550/arXiv.2504.07139.

[McC+06] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955." In: *AI Magazine* 27.4 (Dec. 2006), p. 12. DOI: 10.1609/aimag.v27i4.1904.

[Mem+21] Z. A. Memish, M. M. Altuwaijri, A. H. Almoeen, and S. M. Enani. "The Saudi Data & Artificial Intelligence Authority (SDAIA) Vision: Leading the Kingdom's Journey toward Global Leadership." In: *Journal of Epidemiology and Global Health* 11.2 (June 2021), pp. 140–142. ISSN: 2210-6006. DOI: 10.2991/jegh.k.210405.001.

[Mis25a] Mistral AI. *Mistral OCR | Mistral AI*. en. Mar. 2025.

[Mis25b] Mistral AI. *Tokenization | Mistral AI*. en. 2025.

[Nil98] N. J. Nilsson. *Artificial Intelligence: A New Synthesis*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Mar. 1998. ISBN: 978-0-08-049945-1.

[NVI25] NVIDIA Corporation. *NVIDIA Corporation - Company Overview*. May 2025.

[Ope24] OpenAI. *Introducing ChatGPT*. en-US. Mar. 2024.

[PMG98] D. Poole, A. Mackworth, and R. Goebel. *Computational Intelligence: A Logical Approach*. Jan. 1998. ISBN: 978-0-19-510270-3.

[Pra+20] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert. "MLS: A Large-Scale Multilingual Dataset for Speech Research." In: *Interspeech 2020*. arXiv:2012.03411 [eess]. Oct. 2020, pp. 2757–2761. DOI: 10.21437/Interspeech.2020-2826.

[Pre25] A. Press. "AI startup Anthropic agrees to pay $1.5bn to settle book piracy lawsuit." en-GB. In: *The Guardian* (Sept. 2025). ISSN: 0261-3077.

[PW17] O. Press and L. Wolf. *Using the Output Embedding to Improve Language Models*. arXiv:1608.05859 [cs]. Feb. 2017. DOI: 10.48550/arXiv.1608.05859.

[Rad+22] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. *Robust Speech Recognition via Large-Scale Weak Supervision*. arXiv:2212.04356 [eess]. Dec. 2022. DOI: 10.48550/arXiv.2212.04356.

[Rep20] Rep. Eddie Bernice Johnson. *H.R.6216 - 116th Congress (2019-2020): National Artificial Intelligence Initiative Act of 2020*. eng. legislation. Archive Location: 2020-03-12. Mar. 2020.

[Ric91] E. Rich. *Artificial intelligence*. eng. New York : McGraw-Hill, 1991. ISBN: 978-0-07-052263-3 978-0-07-100894-5 978-0-07-460081-8.

[RND10] S. J. Russell, P. Norvig, and E. Davis. *Artificial Intelligence: A Modern Approach*. en. Prentice Hall, 2010. ISBN: 978-0-13-604259-4.

[Sca25] M. Scarcella. "Apple sued by authors over use of books in AI training." en. In: *Reuters* (Sept. 2025).

[Sch+22] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. *LAION-5B: An open large-scale dataset for training next generation image-text models*. arXiv:2210.08402 [cs]. Oct. 2022. DOI: 10.48550/arXiv.2210.08402.

[Sch25] Scholarcy. *Scholarcy - Knowledge made simple*. en. 2025.

[Sev+22] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos. "Compute Trends Across Three Eras of Machine Learning." In: *2022 International Joint Conference on Neural Networks (IJCNN)*. arXiv:2202.05924 [cs]. July 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9891914.

[SJZ25] N. L. Schroeder, C. D. Jaldi, and S. Zhang. *Large Language Models with Human-In-The-Loop Validation for Systematic Review Data Extraction*. arXiv:2501.11840 [cs]. Jan. 2025. DOI: 10.48550/arXiv.2501.11840.

[Sou25] Sourcely. *Sourcely | Find Academic Sources with AI*. en. 2025.

[SS18] P. P. Shinde and S. Shah. "A Review of Machine Learning and Deep Learning Applications." In: *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. Aug. 2018, pp. 1–6. DOI: 10.1109/ICCUBEA.2018.8697857.

[Tak+25] H. Takita, D. Kabata, S. L. Walston, H. Tatekawa, K. Saito, Y. Tsujimoto, Y. Miki, and D. Ueda. "A systematic review and meta-analysis of diagnostic performance comparison between generative AI and physicians." en. In: *npj Digital Medicine* 8.1 (Mar. 2025). Publisher: Nature Publishing Group, p. 175. ISSN: 2398-6352. DOI: 10.1038/s41746-025-01543-z.

[Tou+23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. *LLaMA: Open and Efficient Foundation Language Models*. arXiv:2302.13971 [cs]. Feb. 2023. DOI: 10.48550/arXiv.2302.13971.

[UK] UK Government. *AI Foundation Models: Initial report*. en. UK Goverment.

[Vas+23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. arXiv:1706.03762 [cs]. Aug. 2023. DOI: 10.48550/arXiv.1706.03762.

[Wan+20] Z. Wang, T. Nayfeh, J. Tetzlaff, P. O'Blenis, and M. H. Murad. "Error rates of human reviewers during abstract screening in systematic reviews." en. In: *PLOS ONE* 15.1 (Jan. 2020). Publisher: Public Library of Science, e0227742. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0227742.

[Wan+25] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. *Wan: Open and Advanced Large-Scale Video Generative Models*. arXiv:2503.20314 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2503.20314.

[Win93] P. H. Winston. *Artificial intelligence*. eng. Reading, Mass. : Addison-Wesley Pub. Co., 1993. ISBN: 978-0-201-53377-4 978-0-201-60086-5.

[WXL25] R. Wang, Z. Xu, and F. X. Lin. *WhisperFlow: speech foundation models in real time*. arXiv:2412.11272 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2412.11272.

[Xu+22] C. Xu, T. Yu, L. Furuya-Kanamori, L. Lin, L. Zorzela, X. Zhou, H. Dai, Y. Loke, and S. Vohra. "Validity of data extraction in evidence synthesis practice of adverse events: reproducibility study." en. In: *BMJ* 377 (May 2022). Publisher: British Medical Journal Publishing Group Section: Research, e069155. ISSN: 1756-1833. DOI: 10.1136/bmj-2021-069155.

[Yua+21] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, C. Liu, M. Liu, Z. Liu, Y. Lu, Y. Shi, L. Wang, J. Wang, B. Xiao, Z. Xiao, J. Yang, M. Zeng, L. Zhou, and P. Zhang. *Florence: A New Foundation Model for Computer Vision*. arXiv:2111.11432 [cs]. Nov. 2021. DOI: 10.48550/arXiv.2111.11432.

Statement on the Use of Artificial Intelligence

Portions of this thesis were supported by the use of generative artificial intelligence tools in full compliance with all relevant TUM guidelines at the time of writing and submission.

The following tools have been utilized:

During the initial research proposal and literature review phase, *Perplexity* has been used to find publications and sources to gain a deeper understanding of specific topics.

During the software development phase, *Github Copilot* as well as *Cursor* have been used with different underlying models, including mostly the *Claude* model family by *Anthropic*, to generate, change, format, and refactor code based on highly specific instructions. Importantly, the tools have solely been used for development tasks, but not for any major software architecture or design decisions.

As extensively described, the main content of the thesis relied on leveraging Large Language Models to analyze publications and extract information. Hence, models by *Google Gemini* and *DeepSeek* have been used for these tasks and accessed via their respective APIs.

During the writing process, *Grammarly* has been used to check basic grammar and spelling of the written thesis. In addition, *Google Gemini* and *Chat-GPT* were used to create charts, tables, and other figures in correct LaTeX-syntax, based on provided data.

No tools have been used for generating original arguments, interpreting results, drawing conclusions, or writing the thesis itself.