THE UNIVERSITY *of* EDINBURGH

A Unified Framework for Decomposing Neural Representations and Analyzing Specialization in Language Models

Zheng Zhao

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2025

Abstract

The rise of large, pre-trained Transformer models has transformed Natural Language Processing (NLP), yet the internal mechanisms by which these models handle diverse and heterogeneous data remain insufficiently understood. This thesis addresses this gap by developing and applying a unified analytical framework to examine how such models represent, differentiate, and specialize for distinct subpopulations of data. The central contribution is the **Model-Oriented Sub-population and Spectral Analysis (MOSSA)** framework, which systematically contrasts a generalist model, trained on multiple domains, languages, or tasks, with a suite of specialist control models trained on individual subpopulations. Through a set of advanced matrix analysis techniques, MOSSA quantifies representational similarities layer by layer, revealing where and how knowledge encoding and adaptation occur within the model architecture.

The framework is applied across three major studies of increasing complexity. The first investigates domain learning using Singular Vector Canonical Correlation Analysis (SVCCA) to assess how model capacity and data scale affect the encoding of domain-specific information. The findings show that larger models not only generalize across domains but also embed domain-specialist behavior within their internal representations, particularly for domain-specific vocabulary.

The second study extends this approach to multilingual modeling. A joint matrix factorization method is introduced to analyze representational structures across 33 languages. The analysis uncovers systematic variation in the encoding of morphosyntactic information across layers, shaped by linguistic properties such as script and morphological complexity. Moreover, the learned representations align with cross-lingual task performance and yield linguistically meaningful phylogenetic structures.

The third study explores the dynamics of massively multi-task instruction tuning in Large Language Models (LLMs). Using Centered Kernel Alignment (CKA) within MOSSA, we examine how an LLM represents over 60 NLP tasks. The results reveal a distinct architectural segmentation: early *shared layers* encode general-purpose features, intermediate *transition layers* rapidly acquire task-specific information, and later *refinement layers* optimize representations for precise task execution.

Together, these studies establish a principled methodology for probing and

interpreting the internal organization of large neural models. The thesis demonstrates that generalist language models systematically partition their representational space, forming specialized subspaces tailored to different data regimes. This work identifies where such specialization arises within model depth and clarifies the mechanisms underlying adaptation, multilinguality, and multi-task learning in contemporary NLP systems.

Lay Summary

Artificial Intelligence systems such as ChatGPT can write emails, translate languages, and assist with homework. Yet one fundamental question remains: how does a single AI learn to perform so many tasks without confusing them? How can it process recipes, legal documents, and computer code while keeping them distinct? Although these models are remarkably capable, their internal workings are still largely unknown. They operate as powerful black boxes whose internal logic remains hidden.

This thesis introduces a new framework to look inside that black box, called Model-Oriented Sub-population and Spectral Analysis (MOSSA). The main idea is similar to studying a polyglot who speaks many languages. To understand how they manage this skill, one might compare their brain activity to that of several specialists, each fluent in only one language. If the polyglot's brain shows a "French-like" pattern when speaking French, it suggests a dedicated area for that language. In a similar way, MOSSA compares a large, general-purpose AI model to smaller, specialized models. By measuring how closely their internal patterns align, the framework reveals where the generalist model develops specialized sections for particular tasks or data types.

This method is applied in three main studies. The first shows that larger AI models better separate different text domains, such as Amazon reviews about books and electronics. The second examines multilingual models trained on over thirty languages and finds that their internal organization reflects linguistic families, grouping related languages such as French and Spanish. The third analyzes instruction-tuned models that perform dozens of tasks and discovers that their layers act like an assembly line: early layers build general understanding, middle layers specialize, and later layers refine the output.

The thesis concludes that large AI models are not disordered systems but structured networks that create specialized internal regions for different kinds of knowledge. This understanding provides a blueprint for how such models think and offers a foundation for building more transparent, efficient, and trustworthy AI systems.

Acknowledgements

This thesis represents the result of many years of work, a journey that would not have been possible without the help and support of many people. I am deeply thankful to everyone who has guided, supported, and encouraged me along the way.

First, I would like to express my deepest gratitude to my primary supervisor, Shay Cohen. I feel very fortunate to have had you as my mentor. Your curiosity, clear thinking, and high standards have shaped how I approach research. I will always remember our meetings with scratch paper and notebooks, where we explored problems through linear algebra and derived equations to understand their underlying principles. Those moments taught me not only the technical aspects of our field but also the joy of doing real science. Your trust in my ideas and your constant support helped me grow as a researcher and as a person.

I am also very grateful to my second supervisor, Bonnie Webber, who not only provided sharp and insightful comments on our projects but also introduced me to Shay in the first place, an act for which I will always be grateful. Your ability to go over examples and pinpoint the small, crucial details that matter most has consistently set our work on the right track. Your wisdom and perspective have been invaluable.

I owe special thanks to Yftah Ziser, who has been both a mentor and a friend, and who guided me as if he were a third supervisor. Thank you for the many late-night discussions and for always being willing to talk through difficult questions. Your enthusiasm for research and your open way of thinking have been a great source of inspiration.

My research was also shaped by my time spent in industry. I would like to thank Emilio Monti and Clara Vania for their guidance during my internships at Amazon, and Nicola Cancedda for giving me a wonderful opportunity to learn at Meta FAIR. These experiences helped me connect academic ideas with real-world applications.

I would also like to thank Pinzhen Chen, who has been a close friend since my undergraduate years. Our conversations have always been thoughtful and motivating. I am grateful to Marcio Fonseca, Yifu Qiu, and Shun Shao for our collaborations, from which I learned an enormous amount. I also thank Ashok Urlana, Manuj Malik, Dongqi Liu, Chenxi Whitehouse, Rohit Saxena, Joshua

Ong, Aryo Gema, Sohee Yang, Edoardo Ponti, and Anna Korhonen for their collaborations, each of which was a valuable experience.

The day-to-day life of a PhD student is shaped by the people around them. I am thankful to my officemates, Anil Batra, Agostina Calabrese, Verna Dankers, Radina Dobreva, Wenyu Huang, Matthias Lindemann, Danyang Liu, and Mengyu Wang, for their company, good conversations, and the many coffee breaks that made the office feel like home. I am also grateful to everyone in the CDT programme, in particular: Aida, Amr, Eddie, Emelie, Gautier, Giulio, Nikita, Parag, Rimvydas, Shangmin, Siqi, Steph, Tom, and Wanqiu. A very special thank you goes to Sally Galloway, the CDT administrator, whose kindness and patience made the programme run smoothly for all of us. I also thank Patrick Hudson from the IGS and the Informatics administrative team for their continued support. I am thankful to the members of Shay's research group, Balint, Dominik, Ke, Matt, Marcio, Nickil, Ronald, Shiwen, Shun, and Yifu, for their helpful discussions. I truly enjoyed our reading groups and the programming sessions. I will miss the pizzas!

I am deeply thankful to my family. To my parents and grandparents, thank you for your endless love, encouragement, and belief in me. You built the foundation that made everything else possible. Your support, even from afar, has been my greatest strength.

Finally, I want to thank my partner, Cecilia. Your love, patience, and constant support have been my anchor through every stage of this journey. Thank you for celebrating the good times, for standing by me through the hard times, and for bringing happiness and balance into my life. None of this would have been possible without you.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Zheng Zhao*)

Contents

Chapter 1

Introduction

The Transformer architecture (Vaswani et al., 2017) has transformed Natural Language Processing (NLP). Large Language Models (LLMs) such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), and their successors now define the state of the art across a broad range of linguistic tasks, from text classification and machine translation to complex, open-ended instruction following (Brown et al., 2020; Wei et al., 2022b; Touvron et al., 2023; Chowdhery et al., 2023; OpenAI et al., 2024). The prevailing approach involves pre-training a single large model on terabytes of unlabeled text (Howard and Ruder, 2018), followed by adaptation to specific applications through fine-tuning or prompting (Sanh et al., 2022; Ouyang et al., 2022). A central factor in this success is the model's ability to learn from heterogeneous data mixtures that cover multiple domains, numerous languages, and a wide variety of task formats. This *generalist* training paradigm has proven remarkably effective, but it also raises a fundamental question: How do these models work internally?

Despite their success, the internal mechanisms of large neural networks remain poorly understood. We still lack a clear explanation of how a single unified model learns to process data that vary so widely in structure and semantics. Does a model develop a universal internal language that serves all domains and tasks, or does it organize its parameters into specialized subspaces that handle different types of data? How and where in the architecture does such specialization occur, and what principles govern it? Addressing these questions is essential for improving model efficiency, diagnosing bias, enabling cross-domain and cross-lingual transfer, and building more transparent and reliable AI systems (Belinkov, 2022).

This thesis tackles these questions by proposing and validating a unified analytical framework for investigating how generalist Transformer models represent and specialize for distinct data subpopulations. The central contribution is a methodology termed **Model-Oriented Sub-population and Spectral Analysis (MOSSA)**. Rather than examining a single model in isolation, MOSSA compares the internal representations of a generalist model trained on a mixed dataset with those of specialist control models trained on individual subpopulations, such as a single domain, language, or task. By measuring layer-wise similarity between the generalist and the specialists, MOSSA identifies where and how the generalist model learns to emulate specialist behavior.

This investigation proceeds through three case studies of increasing complexity, each applying the MOSSA framework to a core problem in NLP:

1. **Domain Adaptation:** The first case study establishes the foundation of our methodology. Using Singular Vector Canonical Correlation Analysis (SVCCA), we analyze models trained on distinct text domains. The results show that model capacity plays a crucial role in enabling specialization: larger models more effectively learn domain-specific representations that align with those of domain-specialist models, especially for domain-dependent vocabulary (Zhao et al., 2022).

2. **Multilingualism:** The second study extends this approach to multilingual modeling. We develop a joint matrix factorization method to analyze representations across 33 languages. The analysis shows that morphosyntactic information appears at different depths depending on linguistic properties such as writing systems. Moreover, the learned representational structures correlate with cross-lingual task performance and reconstruct linguistically consistent language family trees (Zhao et al., 2023).

3. **Massively Multi-task Instruction Tuning:** The third study applies MOSSA to instruction-tuned LLMs. Using Centered Kernel Alignment (CKA), we examine how a single model represents over 60 NLP tasks. The analysis reveals a functional segmentation within the network architecture: early *shared layers* encode general-purpose information, middle *transition layers* rapidly acquire task-specific patterns, and later *refinement layers* optimize representations for task execution (Zhao et al., 2024b).

Together, these studies support a central conclusion: generalist language models do not form a single, uniform representation of language. Instead, they partition their representational space, dynamically creating specialized subspaces to process different types of data. This thesis provides a robust and scalable framework for studying this phenomenon, yielding new insights into domain adaptation, multilingualism, and multi-task learning in Transformer-based systems. By decomposing the internal structures of these models, we move from viewing them as black boxes toward understanding them as structured and interpretable computational systems.

1.1 Aim of the Thesis

The primary aim of this thesis is to develop and validate a unified analytical framework, Model-Oriented Sub-population and Spectral Analysis (MOSSA), to systematically investigate how large, generalist Transformer models represent and specialize for distinct data subpopulations. By applying this framework to key challenges in NLP, including domain adaptation, multilingualism, and multi-task learning, this work seeks to reveal the underlying principles that govern representation learning and adaptation in these complex models.

1.2 Thesis Overview

This thesis is organized into seven chapters, progressing from foundational concepts to the introduction of the MOSSA framework and its empirical applications.

Chapter 1: Introduction. This present chapter sets the stage for the thesis. It introduces the problem of interpretability in generalist language models, states the primary research aims, and provides an overview of the structure and contributions of the work.

Chapter 2: Background. This chapter reviews the foundational knowledge required to understand the subsequent research. It details the Transformer architecture, the lifecycle of modern language models from pre-training to fine-tuning, and surveys the field of interpretability, with a focus on the mathematical techniques for representation similarity analysis.

Chapter 3: Model-Oriented Sub-population and Spectral Analysis. This chapter details the core methodological contribution of the thesis. It formally introduces the MOSSA framework, its guiding principles of comparative analysis, and its adaptable toolkit of similarity metrics, including SVCCA, PARAFAC2, and CKA.

Chapter 4: Case Study I: Understanding Domain Learning. This chapter presents the first empirical application of the MOSSA framework. Using SVCCA, we investigate how models trained on multiple text domains represent domain-specific information, establishing the baseline findings for our investigation into specialization.

Chapter 5: Case Study II: A Joint Analysis of Multilingual Representations. The second case study scales up our analysis to multilingual models. It introduces a novel application of joint matrix factorization (PARAFAC2) within the MOSSA framework to simultaneously analyze representations across 33 languages.

Chapter 6: Case Study III: Specialization in Instruction-Tuned LLMs. The final case study applies the MOSSA framework, using CKA, to the modern paradigm of massively multi-task instruction-tuned LLMs. It investigates how a single model learns to represent over 60 distinct NLP tasks and identifies a functional segmentation of the model's architecture.

Chapter 7: Conclusion. The final chapter concludes the thesis by synthesizing the findings from the three case studies. It discusses the broader implications for understanding specialization in large language models, and outlines directions for future research.

1.3 Published Work

The core technical contributions of this thesis, presented in Chapters 4, 5, and 6, are based on three peer-reviewed publications:

- **Chapter 4**: Zheng Zhao, Yftah Ziser, and Shay Cohen. 2022. Understanding Domain Learning in Language Models Through Subpopulation

Analysis. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

- **Chapter 5**: Zheng Zhao, Yftah Ziser, Bonnie Webber, and Shay Cohen. 2023. A Joint Matrix Factorization Analysis of Multilingual Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

- **Chapter 6**: Zheng Zhao, Yftah Ziser, and Shay Cohen. 2024. Layer by Layer: Uncovering Where Multi-Task Learning Happens in Instruction-Tuned Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Chapter 2

Background

The research presented in this thesis builds on extensive work in neural network architectures, model training paradigms, and the field of interpretability. This chapter provides the background necessary to contextualize our contributions. We begin with a description of the Transformer architecture, the computational foundation of modern NLP. We then outline the lifecycle of a typical large language model, from self-supervised pre-training to adaptation through fine-tuning. Finally, we review the field of representation analysis, introducing the mathematical tools and concepts that directly inform the methodological framework developed in this thesis.

2.1 Notations and Conventions

Throughout this thesis, we adhere to the following notational conventions to ensure clarity and consistency in all mathematical descriptions:

- **Scalars** are denoted by lowercase italic letters (e.g., n, d, λ).
- **Vectors** are denoted by lowercase boldface letters (e.g., $\mathbf{x}, \mathbf{q}$). Unless otherwise specified, vectors are assumed to be column vectors. A row vector is denoted by the transpose, e.g., $\mathbf{x}^\top$.
- **Matrices** are denoted by uppercase boldface letters (e.g., X, W_Q).
- **Sets** are denoted by uppercase calligraphic letters (e.g., $\mathcal{X}$).
- The set of real numbers is denoted by $\mathbb{R}$. The space of d-dimensional real vectors is $\mathbb{R}^d$, and the space of $n \times d$ real matrices is $\mathbb{R}^{n \times d}$.

2.2 The Transformer Architecture

The Transformer architecture (Vaswani et al., 2017) marked a major advance in NLP. By removing recurrence and relying entirely on self-attention, the Transformer enabled unprecedented parallelization and scalability, which facilitated the development of large-scale language models that define the field today.

2.2.1 The Self-Attention Mechanism

The Transformer processes a sequence of input vectors (e.g., token embeddings) by allowing each vector to interact with all others in the sequence. This interaction, or self-attention, enables the model to dynamically weigh the importance of other tokens when updating each token's representation.

Query, Key, and Value Projections The self-attention mechanism is built upon three learned linear projections of the input token representations. For each token embedding $\mathbf{x}_i \in \mathbb{R}^{d_{model}}$, we create a **Query** vector ($\mathbf{q}_i$), a **Key** vector ($\mathbf{k}_i$), and a **Value** vector ($\mathbf{v}_i$) by multiplying it with learned weight matrices:

$$\mathbf{q}_i = W_Q^\top \mathbf{x}_i$$
$$\mathbf{k}_i = W_K^\top \mathbf{x}_i$$
$$\mathbf{v}_i = W_V^\top \mathbf{x}_i$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_{model} \times d_k}$ are learnable parameters. Conceptually, self-attention can be understood as a form of flexible information retrieval. The Query vector acts as a probe from the current token's perspective, seeking relevant information. The Key vectors serve as an index, allowing each token to signal its potential relevance. The Value vectors hold the content to be retrieved.

Scaled Dot-Product Attention The relevance of each token in the sequence to the current token is computed as the dot product between the current token's Query vector and every other token's Key vector. This raw similarity score is scaled and normalized via a softmax function to yield a probability distribution, or a set of *attention weights*. These weights determine the proportion of each token's Value vector that should be incorporated into the updated representation. The final output for a token is the weighted sum of all Value vectors in the sequence.

This entire operation can be expressed concisely in matrix form. Given matrices of queries Q, keys K, and values V, the attention output is computed as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

The scaling factor $\frac{1}{\sqrt{d_k}}$ is crucial for stabilizing gradients during training, preventing the dot products from growing too large (Vaswani et al., 2017). A diagram of this mechanism is shown in Figure 2.1.

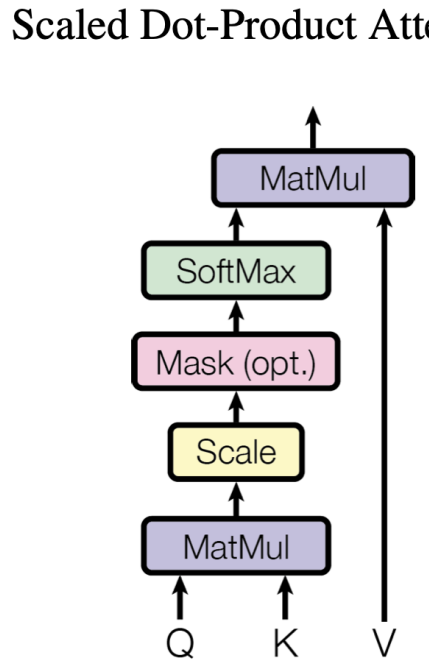


Figure 2.1: Illustration of the Scaled Dot-Product Attention mechanism, from Vaswani et al. (2017). The compatibility between a query and a set of keys is computed, scaled, and converted into weights, which are then used to create a weighted sum of the values.

2.2.2 Multi-Head Attention and Feed-Forward Networks

The rationale for multiple attention heads A single attention head may focus on only one type of relationship. To capture multiple relationships simultaneously, the Transformer uses **Multi-Head Attention**, applying several independent attention heads in parallel. Each head uses different projection matrices (W_Q^i, W_K^i, W_V^i) and learns to focus on different aspects, such as syntax, semantics, or coreference. The outputs from all heads are concatenated and linearly projected to form the final output of the sub-layer (Figure 2.2).

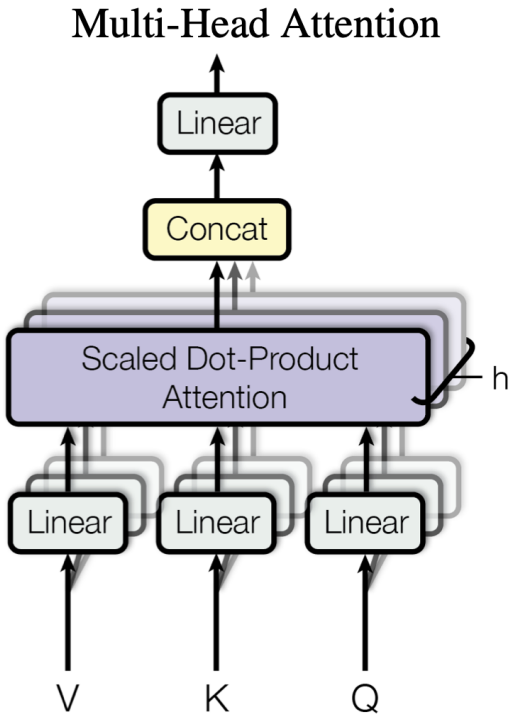


Figure 2.2: Illustration of the Multi-Head Attention mechanism, from Vaswani et al. (2017). Multiple attention heads independently project the input queries, keys, and values into subspaces, compute attention outputs, and concatenate them to form the final representation.

Position-wise Feed-Forward Networks (FFNs) Following the multi-head attention block, the representation for each token is processed independently by a Position-wise Feed-Forward Network (FFN). This typically consists of two linear transformations with a non-linear activation function in between, such as ReLU (Agarap, 2019) or GeLU (Hendrycks and Gimpel, 2023):

$$\mathrm{FFN}(\mathbf{x}) = \max(0, \mathbf{x}^\top W_1 + \mathbf{b}_1^\top) W_2 + \mathbf{b}_2$$

The FFN introduces non-linearity and can be seen as a content-based transformation, further refining the representation of each token.

Residual Connections and Layer Normalization To facilitate the training of deep networks, each sub-layer (both self-attention and FFN) is encased in a standard block containing two additional components. A **residual connection** (He et al., 2016) adds the input of the sub-layer to its output, creating a direct information pathway that mitigates the vanishing gradient problem. This is followed

by **layer normalization** (Ba et al., 2016), which stabilizes training dynamics by normalizing the activations across the features for each token independently. The complete operation for a sub-layer is thus $\text{LayerNorm}(\mathbf{x} + \text{Sublayer}(\mathbf{x}))$.

2.2.3 Architectural Variants: Encoder, Decoder, and Encoder-Decoder

The Transformer's building blocks can be arranged in several ways to suit different NLP tasks.

Bidirectional context in Encoders (e.g., BERT) An **encoder-only** architecture, famously used in BERT (Devlin et al., 2019), consists of a stack of Transformer blocks where the self-attention mechanism can attend to all tokens in the input sequence. This creates deep, bidirectional representations that are highly effective for natural language understanding (NLU) tasks such as text classification, named entity recognition, and question answering (Rogers et al., 2020).

Causal masking in Decoders for autoregressive generation (e.g., GPT) A **decoder-only** architecture, used in models like the GPT series (Radford et al., 2018, 2019; Brown et al., 2020), modifies the self-attention mechanism with **causal masking**. This mask prohibits any token from attending to subsequent tokens in the sequence, thereby enforcing the constraint that the prediction for a token at position i can only depend on known outputs at positions less than i. This autoregressive property is the functional imperative for language generation.

The combined architecture for sequence-to-sequence tasks The original Transformer was an **encoder-decoder** model designed for sequence-to-sequence tasks like machine translation. In this setup, an encoder processes the source sequence to produce a set of contextualized representations. A decoder then autoregressively generates the target sequence. Crucially, in addition to attending to previously generated target tokens, the decoder also performs *cross-attention* over the encoder's final output representations. This allows the generation process to be conditioned on the entirety of the source text. Models like BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020) have successfully leveraged this architecture.

2.3 The Lifecycle of a Modern Language Model

The success of modern LLMs is not just due to their architecture but also to a multi-stage training process that effectively leverages vast amounts of data.

2.3.1 Pre-training on Unlabeled Data

The foundational stage for any LLM is self-supervised pre-training on vast, unlabeled text corpora, often encompassing terabytes of data scraped from the public web. This process imbues the model with a comprehensive, general-purpose understanding of language (Brown et al., 2020; Liang et al., 2023).

Self-Supervised Learning Objectives: MLM and CLM The two dominant self-supervised objectives correspond to the main architectural variants.

- **Masked Language Modeling (MLM)**: Used for pre-training encoders like BERT, MLM involves randomly masking a fraction of input tokens (e.g., 15%) and training the model to predict the original identity of these masked tokens based on the unmasked, bidirectional context.
- **Causal Language Modeling (CLM)**: Also known as the next-token prediction objective, CLM is used for pre-training decoders like GPT. The model is trained to predict the next token in a sequence given only the preceding tokens.

The role of massive, diverse data in pre-training Pre-training on web-scale data allows the model to learn a rich, general-purpose representation of language, capturing everything from basic syntax and semantics to complex world knowledge and reasoning patterns (Zhao et al., 2025). The diversity of this data, spanning countless domains and styles, is critical for the model's ability to generalize to a wide range of downstream applications (Zhang et al., 2025).

2.3.2 Fine-tuning and Instruction Tuning

Once pre-trained, a model's general knowledge must be adapted for specific tasks.

Supervised Fine-tuning for downstream tasks The traditional adaptation method is supervised fine-tuning. In this paradigm, the pre-trained model is further trained on a smaller, labeled dataset tailored to a specific downstream task (e.g., sentiment analysis). Typically, a task-specific classification head is appended to the model, and all or a subset of the model's parameters are updated via backpropagation to minimize a task-specific loss function.

The emergence of Instruction Tuning A more recent and powerful paradigm is **instruction tuning** (Wei et al., 2022a; Sanh et al., 2022). Rather than specializing the model for a single task, instruction tuning fine-tunes it on a large, curated mixture of many different NLP tasks formatted as natural language prompts (e.g., "Translate the following English sentence to French: ..."). This process teaches the model not just how to perform individual tasks, but the more general skill of following instructions. This has been shown to dramatically improve a model's zero-shot and few-shot performance on entirely new, unseen tasks (Ouyang et al., 2022).

2.3.3 Alignment: Conforming to Human Preferences

A third stage of fine-tuning, popularized by InstructGPT (Ouyang et al., 2022), aims to align model behavior more closely with complex human preferences and values. The predominant technique for this is **Reinforcement Learning from Human Feedback (RLHF)**. This process typically involves three steps: 1) collecting a dataset of human preferences by asking annotators to rank multiple model-generated responses to a given prompt; 2) training a separate *reward model* to predict these human preference scores; and 3) using this reward model's output as a reward signal to further fine-tune the LLM using a reinforcement learning algorithm like Proximal Policy Optimization (PPO; Schulman et al. 2017). This process is highly effective at steering models to be more helpful, harmless, and conversational (Bai et al., 2022).

2.4 The Field of Interpretability

As the capabilities of language models have expanded, the scientific need to understand their internal mechanisms has grown as well. The field of interpretability

aims to move beyond viewing these models as opaque systems and instead develop methods to explain the computations they perform and the representations they learn.

2.4.1 Defining a Representation

A central object of study in interpretability is the notion of a **representation**. Within a neural network, a representation is an activation vector or matrix that encodes information about an input. This can range from the initial token embedding, which is largely context-independent, to the hidden states produced by each Transformer layer, which become progressively more contextualized, abstract, and task-relevant as information propagates deeper into the network. A primary goal of many interpretability methods is to decipher the information encoded within these layer-wise hidden states.

2.4.2 Extrinsic Analysis: Probing Classifiers

One popular family of methods for analyzing representations is known as extrinsic analysis, or **probing**.

Training simple linear models to predict linguistic properties The standard probing methodology (Alain and Bengio, 2016; Belinkov et al., 2017a; Giulianelli et al., 2018) involves freezing the parameters of a pre-trained model and extracting its internal representations for a corpus annotated with a specific linguistic property (e.g., part-of-speech tags, syntactic tree depth). A simple, often linear, classifier, the *probe*, is then trained on these frozen representations to predict the property. The performance of this probe is interpreted as a proxy for how explicitly the linguistic information is encoded in the representations.

Limitations of the probing paradigm While probing has yielded valuable insights, the paradigm suffers from known limitations. A principal concern is the representation vs. classifier dilemma: a high-performing probe might indicate not that the information is explicitly represented, but that the probe itself is powerful enough to learn the task from complex, non-linear features within the representations (Zhang and Bowman, 2018; Hewitt and Liang, 2019). Consequently, probing results can be sensitive to the probe's architecture and hyperparameters,

complicating definitive conclusions about the representation itself (Zhao et al., 2024a).

2.4.3 Intrinsic Analysis: Similarity-Based Methods

An alternative approach, intrinsic analysis, circumvents the issues of training an external model by directly analyzing the geometric and statistical properties of the representation space itself.

The motivation: directly comparing representation spaces Similarity-based methods (Li et al., 2015; Raghu et al., 2017; Saphra and Lopez, 2019) directly compare two sets of representations to quantify their similarity. This avoids the confounds of a probing classifier and allows for a more direct assessment of how a model's internal view of the data changes across layers, across different models, or after fine-tuning.

Why this is essential for subpopulation analysis This intrinsic, similarity-based approach is the philosophical and practical cornerstone of the MOSSA framework presented in this thesis. By directly comparing the representation spaces of a generalist model and a specialist model, we can quantify their alignment without relying performance on a downstream task. This allows for a standardized, comparable measure of specialization across highly diverse subpopulations like domains, languages, and tasks, for which a common extrinsic evaluation metric may not exist.

2.5 Mathematical Foundations of Representation Similarity

The similarity-based methods that underpin this thesis are built upon a foundation of well-established matrix analysis techniques.

2.5.1 Singular Value Decomposition (SVD)

Singular Value Decomposition is a fundamental factorization of any matrix $M \in \mathbb{R}^{n \times d}$ into three constituent matrices:

$$M = U\Sigma V^\top$$

where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{d \times d}$ are orthogonal matrices whose columns are the left- and right-singular vectors, respectively, and $\Sigma \in \mathbb{R}^{n \times d}$ is a rectangular diagonal matrix containing the non-negative singular values, sorted in descending order. Intuitively, SVD finds the optimal orthonormal bases for the input space (rows of M, basis in V) and output space (columns of M, basis in U) such that the transformation between them is a simple scaling by the singular values in Σ.

2.5.2 Principal Component Analysis (PCA)

Principal Component Analysis is a ubiquitous technique for dimensionality reduction that aims to find the directions of maximal variance in a dataset. Given a data matrix $X \in \mathbb{R}^{n \times d}$ containing n data points of d dimensions, PCA finds a new set of orthogonal coordinates, known as principal components. The first principal component is the direction along which the data varies the most. The second is the direction of greatest variance in the subspace orthogonal to the first, and so on.

Computationally, PCA is performed by applying SVD to the mean-centered data matrix, $X_c = X - \bar{X}$. The right-singular vectors (the columns of V) are the principal components. Projecting the data onto the first k principal components provides the optimal k-dimensional linear approximation of the data in a variance-preserving sense. In interpretability, PCA is frequently used to visualize high-dimensional representations by projecting them into two or three dimensions.

2.5.3 Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis (Hardoon et al., 2004) is a statistical method for measuring the linear relationship between two sets of variables. Given two mean-centered data matrices $X \in \mathbb{R}^{n \times d_1}$ and $Y \in \mathbb{R}^{n \times d_2}$ representing two different views of the same n data points, CCA finds a series of pairs of projection vectors $(\mathbf{a}_i, \mathbf{b}_i)$. The first pair, $(\mathbf{a}_1, \mathbf{b}_1)$, is chosen to maximize the Pearson correlation between the projected variables, or canonical variates, $X\mathbf{a}_1$ and $Y\mathbf{b}_1$. Subsequent pairs $(\mathbf{a}_i, \mathbf{b}_i)$

are found to maximize the correlation subject to the constraint that they are uncorrelated with all previous canonical variates.

The output is a set of canonical correlations $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_{\min(d_1,d_2)}$, which quantify the strength of the shared linear relationship between the two spaces. However, classical CCA is sensitive to the specific basis of the representation spaces. A simple rotation or scaling, which might not alter the information content, can drastically change the resulting canonical vectors, making it less reliable for the high-dimensional and often arbitrarily-oriented representations found in neural networks.

2.5.4 Singular Vector Canonical Correlation Analysis

Singular Vector Canonical Correlation Analysis (SVCCA; Raghu et al. 2017) is a powerful extension of CCA engineered to provide a more stable and robust measure of similarity for high-dimensional neural network representations. It mitigates CCA's sensitivity to the basis by first using SVD to identify and isolate the most significant, high-variance subspaces of each representation space before comparing them.

Given two mean-centered representation matrices, $X \in \mathbb{R}^{n \times d_1}$ and $Y \in \mathbb{R}^{n \times d_2}$, the SVCCA algorithm proceeds as follows:

1. **SVD Subspace Identification:** SVD is first performed on both matrices:

$$X = U_X \Sigma_X V_X^\top$$
$$Y = U_Y \Sigma_Y V_Y^\top$$

   This decomposition identifies the principal directions (the columns of V_X and V_Y) of the data. A truncation dimension, k_x, is chosen for X such that the first k_x singular vectors capture a significant fraction (e.g., 99%) of the variance. Let V_{X,k_x} be the matrix containing the first k_x right-singular vectors of X. The original representations are then projected onto this lower-dimensional subspace:

$$X' = XV_{X,k_x}$$

   The same procedure is applied to Y to obtain $Y' = YV_{Y,k_y}$. This step effectively de-noises the representations by removing low-variance directions.

2. **CCA on Subspaces:** Classical CCA is then performed on the resulting projected, dimension-reduced matrices, X' and Y', yielding a set of canonical correlations $\{\rho_i\}_{i=1}^{\min(k_x,k_y)}$.

The final SVCCA similarity score, ρ_{SVCCA}, is typically computed as the average of these canonical correlations:

$$\rho_{SVCCA} = \frac{1}{\min(k_x,k_y)} \sum_{i=1}^{\min(k_x,k_y)} \rho_i$$

This two-stage process ensures that the comparison is focused on the most meaningful, high-variance subspaces of each representation, making the resulting similarity score invariant to orthogonal transformations (like basis rotations) within the original spaces and more robust to noise.

2.5.5 Joint Matrix Factorization

While methods like CCA and SVCCA excel at pairwise comparisons, analyzing an entire collection of related representation spaces requires a different mathematical lens. Joint matrix factorization, particularly tensor decomposition methods, provides a mathematical framework for discovering shared structure across multiple matrices simultaneously. The method most relevant to this thesis is PARAFAC2 (Harshman, 1972).

Consider a set of K related data matrices, $\{M_k\}_{k=1}^{K}$, where each matrix $M_k \in \mathbb{R}^{d_k \times d}$. A key challenge in comparative analysis is that the number of data points d_k (the row dimension) may differ for each matrix, while the feature dimension d (the column dimension) is shared. Decomposing each M_k independently would yield unrelated basis vectors, precluding meaningful comparison.

PARAFAC2 addresses this by decomposing the set under the assumption that they share a common underlying structure in their shared column space. It approximates each matrix M_k as:

$$M_k \approx U_k \Sigma_k V^\top$$

The components of this factorization are:

- $V \in \mathbb{R}^{d \times R}$: A factor matrix of R basis vectors for the shared column space. Crucially, this matrix is **identical for all** $k = 1, \ldots, K$.

- $U_k \in \mathbb{R}^{d_k \times R}$: A matrix-specific factor matrix for the row space of M_k. This component is unique to each matrix in the set.

- $\mathbf{\Sigma}_k \in \mathbb{R}^{R \times R}$: A diagonal matrix of weights, or scaling factors, unique to each matrix k. These weights indicate the importance of each of the R shared basis vectors for reconstructing the specific matrix M_k.

The factors are typically found by minimizing the sum of squared reconstruction errors over all matrices, subject to certain constraints for identifiability (e.g., orthogonality of the columns of U_k):

$$\min_{\{U_k, \mathbf{\Sigma}_k\}, V} \sum_{k=1}^{K} ||M_k - U_k \mathbf{\Sigma}_k V^\top||_F^2$$

By enforcing a shared factor matrix V, PARAFAC2 establishes a common frame of reference, allowing the unique structure of each matrix M_k to be analyzed and compared via its specific factor matrices U_k and scaling weights $\mathbf{\Sigma}_k$.

2.5.6 Centered Kernel Alignment

Centered Kernel Alignment (CKA) is a similarity index, introduced to machine learning by Cortes et al. (2012) and popularized for neural network analysis by Kornblith et al. (2019). It is particularly well-suited for comparing high-dimensional representations, proving robust even when the number of samples is smaller than the number of dimensions. It measures similarity not between the representations directly, but between the relational geometry of the data points as captured by kernel matrices.

Given two representation matrices $X \in \mathbb{R}^{n \times d_1}$ and $Y \in \mathbb{R}^{n \times d_2}$ for the same n data points, the CKA procedure is as follows:

1. **Kernel Matrix Computation:** First, a kernel function $k(\cdot, \cdot)$ is chosen to compute the Gram matrices K_X and K_Y. For a linear kernel, this is simply the dot product of the representations:

$$[K_X]_{ij} = \mathbf{x}_i^\top \mathbf{x}_j \implies K_X = XX^\top$$
$$[K_Y]_{ij} = \mathbf{y}_i^\top \mathbf{y}_j \implies K_Y = YY^\top$$

   Each matrix $K \in \mathbb{R}^{n \times n}$ encodes the similarity relationships between all pairs of the n data points.

2. **Centering:** The kernel matrices are centered to make the measure insensitive to isotropic scaling. This is achieved by multiplying with a centering matrix $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, where $\mathbf{1}$ is an $n \times n$ matrix of ones:

$$K_X^c = HK_XH$$
$$K_Y^c = HK_YH$$

3. **Normalized HSIC Computation:** The similarity is computed using the Hilbert-Schmidt Independence Criterion (HSIC; Gretton et al. 2005), which measures the dependence between two random variables in a kernel space. The empirical estimator of HSIC is the Frobenius dot product of the centered kernel matrices, scaled by a constant. CKA is the normalized HSIC:

$$\text{CKA}(X,Y) = \frac{\text{HSIC}(K_X,K_Y)}{\sqrt{\text{HSIC}(K_X,K_X)\text{HSIC}(K_Y,K_Y)}}$$

where $\text{HSIC}(K_X,K_Y) = \text{vec}(K_X^c)^\top \text{vec}(K_Y^c)$. This can be seen as the squared cosine similarity between the vectorized kernel matrices.

The resulting score ranges from 0 to 1. CKA is invariant to orthogonal transformations (like rotations) and, importantly, also invariant to isotropic scaling of the representations. These properties make it a highly reliable and robust tool for comparing the relational structure of representations across different layers and models.

2.6 Chapter Summary

This chapter has laid the necessary groundwork for the research presented in this thesis. We have reviewed the Transformer architecture that powers modern language models, traced the multi-stage lifecycle through which these models are trained and adapted, and surveyed the key paradigms in the field of interpretability. Most importantly, we have detailed the mathematical foundations of representation similarity analysis, from SVD and CCA to the more advanced and robust SVCCA, PARAFAC2, and CKA techniques. These concepts and tools are not merely background; they provide the essential building blocks for the novel methodological framework that we will now synthesize and introduce in the following chapter.

Chapter 3

Model-Oriented Sub-population and Spectral Analysis

3.1 Decomposing Generalists via Specialists

The main challenge in understanding modern large-scale language models arises from their generalist nature. These models are trained on large and diverse datasets that cover many domains, languages, and tasks. Although this training strategy produces models with broad capability, it also makes it difficult to isolate the mechanisms that govern their behavior on any particular type of data. Traditional analysis methods, such as probing, aim to answer the question, "What linguistic information is encoded in this model's representations?" While useful, this approach can be hard to interpret in a consistent way, since the results often depend on the details of the probing classifier and the evaluation metric, which differ across tasks (Belinkov, 2022).

This thesis introduces and evaluates a different approach, termed **Model-Oriented Sub-population and Spectral Analysis (MOSSA)**. Instead of asking what a generalist model knows in isolation, MOSSA asks: "How does the generalist model's representation of a specific data subpopulation compare to that of a specialist model trained only on that subpopulation?" This comparative framework offers a systematic and interpretable way to decompose the behavior of a complex generalist model. It is conceptually related to Representational Similarity Analysis (Kriegeskorte et al., 2008), which measures similarities between activation patterns in cognitive neuroscience, but MOSSA extends this idea to controlled comparisons between generalist and specialist models in NLP.

3.1.1 Formalizing the Framework

Our analysis tool assumes a distribution $p(\mathbf{X})$ from which a set of examples $\mathcal{X} = \{\mathbf{x}^{(i)} \mid i \in [n]\}$ is drawn. It also assumes a family of binary indicators $\pi_1, \ldots, \pi_d$, where each $\pi_i(\mathbf{x})$ indicates whether the example $\mathbf{x}$ satisfies a certain *subpopulation* attribute i. For example, a subpopulation could be a specific text domain, a language, or an NLP task format. We denote by $\mathcal{X}\big|_{\pi_i}$ the set $\{\mathbf{x}^{(j)} \mid \pi_i(\mathbf{x}^{(j)}) = 1\}$, which is the subset of $\mathcal{X}$ that satisfies attribute i.

Unlike standard diagnostic classifier methods (Belinkov et al., 2017a; Giulianelli et al., 2018), rather than building a model to *predict* the attribute, we perform subpopulation analysis by training two key types of models:

- The **Experimental Model (E)**: This is the generalist model of interest, trained on the full, mixed dataset $\mathcal{X}$ containing multiple subpopulations (e.g., multiple domains, languages, or tasks).
- The **Control Models ($\{\mathbf{C}_i\}$)**: For each subpopulation of interest π_i, we train a corresponding specialist control model, $\mathbf{C}_i$. Each $\mathbf{C}_i$ has the same architecture as $\mathbf{E}$ but is trained only on the specific subset of data $\mathcal{X}|_{\pi_i}$.

We borrow the terminology of "experimental" and "control" from experimental design such as in clinical trials (Hinkelmann and Kempthorne, 2007). The experimental model corresponds to the experimental (or "treatment" in the case of medical trials) group in such trials and the control model corresponds to the control group. By treating the specialist control models as a ground truth for how a model should represent a particular subpopulation, the similarity between the experimental model and a control model becomes a direct measure of specialization. A high degree of similarity indicates that the generalist model has learned to embed a specialist-like representation for that specific subpopulation within its broader representational space.

3.2 The General Methodological Pipeline

The MOSSA framework follows a consistent, four-step pipeline, which is applied in each of the empirical studies in this thesis. This pipeline is illustrated in Figure 3.1.

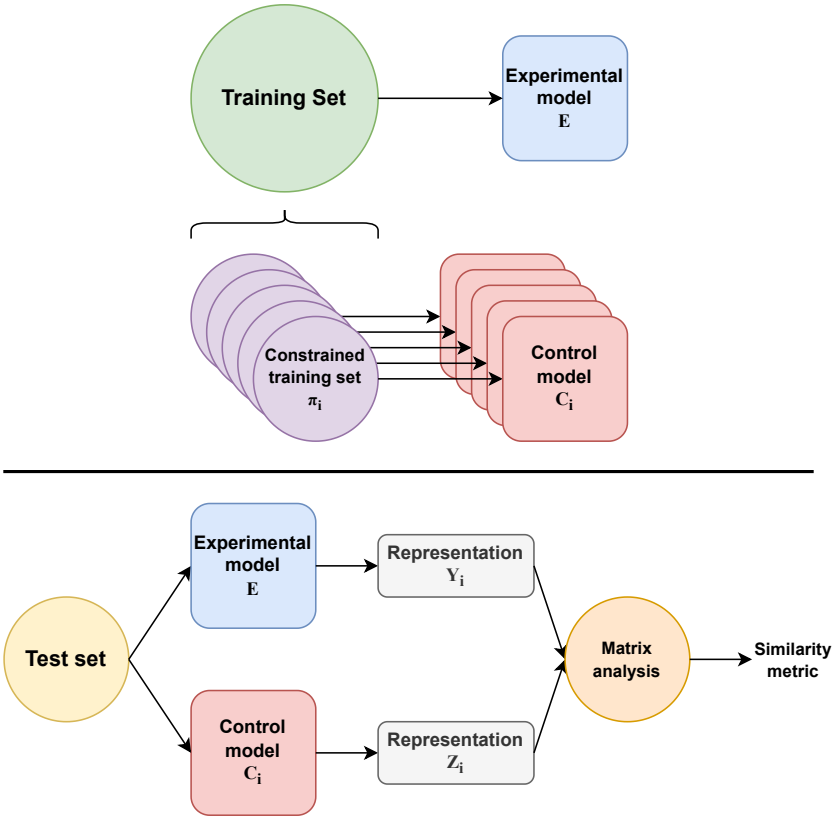


Figure 3.1: The general pipeline of the MOSSA framework. A generalist experimental model (**E**) is trained on mixed data, while specialist control models ($\mathbf{C}_i$) are trained on individual subpopulations. Their internal representations for a common set of inputs are then extracted and compared using a similarity metric.

1. **Define Subpopulations and Train Models:** First, we identify the subpopulations of interest within our dataset (e.g., text domains $\pi_1, \ldots, \pi_d$). We then train the single experimental model $\mathbf{E}$ on the complete dataset and a set of control models $\{\mathbf{C}_i\}$, one for each subpopulation.

2. **Extract Representations:** For a common set of input examples drawn from a specific subpopulation π_i, we feed these inputs through both the experimental model $\mathbf{E}$ and the corresponding control model $\mathbf{C}_i$. We extract the relevant internal representations from a chosen component of each model (e.g., the outputs of a specific layer, or the final token representation). This results in a pair of representation matrices, Y_i (from $\mathbf{E}$) and Z_i (from $\mathbf{C}_i$).

3. **Compute Representational Similarity:** We employ a matrix analysis technique to compute a similarity score between the paired representation matrices (Y_i, Z_i). The choice of similarity metric is a critical, context-

dependent decision, forming an adaptable toolkit for the framework.

4. **Analyze Similarity Patterns:** Finally, we analyze the resulting similarity scores across different model components (e.g., layers) and subpopulations to identify patterns of specialization. For instance, we can determine at which layers the generalist model's representations converge with or diverge from the specialist's.

Through their latent representations, the set of models $\{\mathbf{C}_i\}$ represent the information that is captured about $p(\mathbf{X})$ from the relevant subpopulation of data. By comparing the different models to each other, we can learn what information is captured in the latent representations when a subset of the data is used and whether this information is different from that captured when the whole set of data is used.

3.3 An Adaptable Analytical Toolkit

A key strength of the MOSSA framework is its flexibility. The core comparative principle remains constant, but the specific tool used for the similarity computation (Step 3) can be adapted to the unique challenges posed by the analysis. The research in this thesis demonstrates a clear evolution of this toolkit, with each subsequent study employing a more sophisticated technique to handle increasing complexity.

3.3.1 Pairwise Comparison: SVCCA

For our initial investigation into domain learning (Chapter 4), where the primary analysis involves comparing the multi-domain model to one single-domain model at a time, we employ SVCCA. SVCCA is well-suited for this pairwise comparison task.

Let us assume we have extracted two sets of representations as matrices Y and Z. The procedure first uses SVD to project the representations onto their principal components, filtering out low-variance directions that are often associated with noise. Then, it applies CCA to find the linear transformations that maximize the correlation between the two sets of projected representations. The final SVCCA score is the average of these maximal correlations, providing a single, intuitive measure of linear similarity between the two representation spaces. While

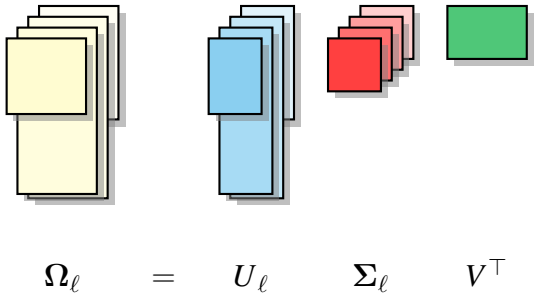


Figure 3.2: A diagram of the joint matrix factorization performed by PARAFAC2. All covariance matrices $\mathbf{\Omega}_\ell$ are decomposed using a shared transformation V.

powerful for one-to-one comparisons, its pairwise nature becomes a limitation when the number of subpopulations is large.

3.3.2 Joint Comparison of Many Subpopulations: PARAFAC2

The challenge of analyzing a multilingual model across 33 languages (Chapter 5) necessitated a move beyond pairwise methods. Comparing each of the 33 languages to the multilingual model one by one would be computationally intensive and, more importantly, would fail to capture the shared structure across all comparisons.

To address this, we adapted our framework to use PARAFAC2, a joint matrix factorization method. The key insight is to analyze the set of cross-covariance matrices $\{\mathbf{\Omega}_\ell\}_{\ell=1}^{L}$, where each $\mathbf{\Omega}_\ell$ captures the covariance between the multilingual model's representations and the ℓ-th monolingual model's representations. PARAFAC2 decomposes this entire set of matrices simultaneously, as illustrated in Figure 3.2, such that:

$$\mathbf{\Omega}_\ell \approx U_\ell \mathbf{\Sigma}_\ell V^\top$$

Crucially, the transformation matrix V, which projects the multilingual representations into a shared latent space, is *the same for all languages*. This joint decomposition allows us to obtain a unique, comparable "signature" for each language, given by the diagonal of $\mathbf{\Sigma}_\ell$. This vector, $\text{sig}(\ell)$, quantifies how the shared multilingual representation is scaled and specialized for language ℓ, enabling a principled comparison across dozens of subpopulations at once.

3.3.3 Robust Comparison for Large Language Models: CKA

For our final study on massively multi-task LLMs (Chapter 6), we faced a different set of constraints. LLM representations have extremely high dimensionality, and it is often only feasible to analyze them using a smaller number of input samples. This scenario poses a challenge for methods like SVCCA.

To ensure robustness, we adopted CKA as our similarity metric. CKA operates by first computing the Gram matrices of the representations, which captures the similarity structure between all pairs of input examples. It then measures the similarity between these Gram matrices. The CKA formulation is:

$$\mathrm{CKA}(Y_t, Z_t) = \frac{\mathrm{HSIC}(K_{Y_t}, K_{Z_t})}{\sqrt{\mathrm{HSIC}(K_{Y_t}, K_{Y_t}) \cdot \mathrm{HSIC}(K_{Z_t}, K_{Z_t})}}$$

where K represents the (centered) kernel matrices and HSIC is the Hilbert-Schmidt Independence Criterion. CKA's key advantages in this context are its invariance to orthogonal transformations and isotropic scaling, and critically, its robustness and reliability even when the number of samples is small relative to the feature dimensionality. This makes it an ideal tool for analyzing the representations of modern LLMs.

3.4 Chapter Summary

This chapter has detailed the Model-Oriented Sub-population and Spectral Analysis (MOSSA) framework, the central methodological contribution of this thesis. The framework is founded on the principle of comparing a generalist experimental model against a set of specialist control models to measure and locate representational specialization. We have outlined its general four-step pipeline and showcased its adaptability through a suite of evolving analytical tools, including SVCCA, PARAFAC2, and CKA, each chosen to meet the specific demands of the problem at hand.

This unified yet flexible framework provides the methodological backbone for the empirical investigations that follow. Having detailed the *how*, we will now proceed to demonstrate the power of this approach in practice. The next chapter will present the first application of the MOSSA framework, using SVCCA to dissect how language models learn to represent distinct text domains.

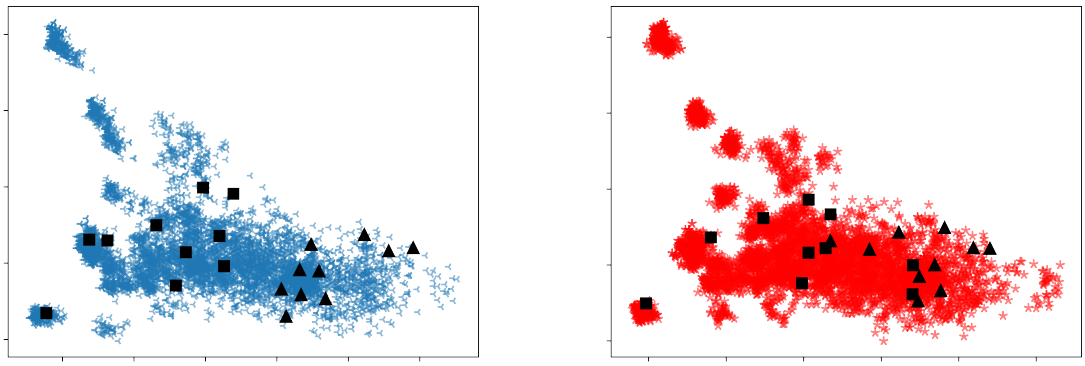
Chapter 4

Case Study I: Understanding Domain Learning

This chapter presents the first of three empirical investigations into the internal representations of generalist language models. The work described here is adapted from our publication, Zhao et al. (2022), and establishes the foundational methodology of MOSSA by applying it to the problem of domain learning. We demonstrate how this framework, which contrasts a generalist (multi-domain) model with specialist (single-domain) controls, can reveal the locus of domain-specific knowledge within a Transformer's architecture.

4.1 Introduction

Pre-trained language models (PLMs) have become an essential modeling component for state-of-the-art natural language processing (NLP) models. They process text into latent representations in such a way that allows an NLP practitioner to seamlessly use these representations for prediction problems of various degrees of difficulty (Wang et al., 2018, 2019). The opaqueness in obtaining these representations has been an important research topic in the NLP community. PLMs, and more generally, neural models, are currently studied to understand their process and behavior in obtaining their latent representations. These PLMs are often trained on large datasets, with inputs originating from different sources. In this chapter, we further develop our understanding of how neural networks obtain their latent representation and study the effect of learning from various domains on the characteristics of the corresponding latent representations.



(a) Experimental model (b) Control model

Figure 4.1: An example of a visualization used with our MOSSA tool. The experimental model, which includes all domain data, separates in its latent representations words related to the Books domain (▲) from general words (■). The control model, on the other hand, mixes them together.

Texts come from various domains that differ in their writing styles, authors and topics (Plank, 2016). In this work, we follow a simple definition of a domain as *a corpus of documents sharing a common topic*. We rely on our proposed subpopulation analysis tool MOSSA to compare and contrast latent representations obtained with and without a specific domain. Our analysis relies on constructing two types of models: *experimental* models, from multi-domain data, and *control* models, from single-domain data. Figure 4.1 describes an example in which this analysis is applied to study the way embeddings for domain-specific words cluster together in the experimental and control model representations.

We believe training in an implicit multi-domain setup is widespread and often overlooked. For example, SQuAD (Rajpurkar et al., 2016), a widely used question-answering dataset composed of Wikipedia articles from multiple domains, is often referred to as a single-domain dataset in domain adaptation works for simplicity (Hazen et al., 2019; Shakeri et al., 2020; Yue et al., 2021). This scenario is also common in text summarization, where many datasets consist of a bundle of domains for news articles (Grusky et al., 2018), academic papers (Cohan et al., 2018; Fonseca et al., 2022), and do-it-yourself (DIY) guides (Cohen et al., 2021). While models that learn from multiple domains are frequently used, their nature and behavior have hardly been explored.

This work sheds light on the way state-of-the-art multi-domain models encode domain-specific information. We focus on two main aspects highly relevant for

many training procedures: model capacity and data size. We discover that model capacity, indicated by the number of its parameters, strongly impacts the amount of domain-specific information multi-domain models store. This property might explain the performance gains of larger models (Devlin et al., 2019; Raffel et al., 2020; Clark et al., 2020b; Srivastava et al., 2022).

4.2 Methodology

This chapter presents the first empirical application of the Model-Oriented Subpopulation and Spectral Analysis (MOSSA) framework, which was detailed in Chapter 3. The central methodology involves comparing a generalist model trained on multiple data subpopulations against specialist models trained on individual subpopulations.

Application to Domain Learning In the context of this chapter, we define our subpopulations as **text domains**. A domain is considered a corpus of documents sharing a common topic. Consequently, our *experimental model* ($\mathbf{E}$) is a language model trained on a mixture of text from several domains, while our *control models* ($\{\mathbf{C}_i\}$) are specialist language models of the same architecture, each trained on text from only a single domain i. By analyzing the similarity between the representations of $\mathbf{E}$ and a given $\mathbf{C}_i$, we can quantify how much domain-specific information for domain i is encoded within the generalist multi-domain model.

Similarity Metric and Implementation For the pairwise comparison between the multi-domain experimental model and each single-domain control model, we employ **Singular Vector Canonical Correlation Analysis (SVCCA)** as our similarity metric. As described in Chapter 2, SVCCA provides a robust measure of the linear similarity between the high-variance subspaces of two sets of neural representations.

The application of the MOSSA pipeline in this study requires several key implementation decisions:

1. **Model Training:** The experimental model $\mathbf{E}$ is trained on a dataset composed of an equal mix of examples from all domains. Each control model $\mathbf{C}_i$ is trained on only the examples from its corresponding domain i.

2. **Representation Extraction:** For a common set of evaluation examples from a single domain i, we extract the hidden state representations from *every layer* of both $\mathbf{E}$ and $\mathbf{C}_i$. This yields a pair of representation matrices for each layer in the model architecture.

3. **Similarity Computation:** The SVCCA score is computed between the corresponding layers of the experimental and control models (e.g., layer j of $\mathbf{E}$ is compared to layer j of $\mathbf{C}_i$). The intensity of the resulting SVCCA score indicates the level of representational overlap for that layer (Saphra and Lopez, 2019).

Using this specific implementation of the MOSSA framework, we are particularly interested in studying the effect of two primary aspects of the training process on the learned domain representations: **dataset size** and **model capacity**. The following section details the specific datasets and model architectures used to conduct this investigation.

4.3 Experimental Setup

Data We use the Amazon Reviews dataset (Ni et al., 2019), a dataset that facilitates research in tasks like sentiment analysis (Zhang et al., 2020), aspect-based sentiment analysis, and recommendation systems (Wang et al., 2020). The reviews in this dataset are explicitly divided into different product categories that serve as domains, which makes it a natural testbed for many multi-domain studies. A noteworthy example of a research field that heavily relies on this dataset is domain adaptation (Blitzer et al., 2007; Ziser and Reichart, 2018b; Du et al., 2020; Lekhtman et al., 2021; Long et al., 2022), which is the task of learning robust models across different domains, closely related to our research.[1] We sort the domains by their review counts and pick the top five, which results in: Books, Clothing Shoes and Jewelry, Electronics, Home and Kitchen, and Movies and TV domains. To further validate our data quality, we use the 5-core subset of the data, ensuring that all reviewed items have at least five reviews authored by reviewers who wrote at least five reviews.

A representative dataset sample is presented in Table 4.1. We consider the different domains within the Amazon review dataset as *lexical domains*, i.e., do-

[1] We use the latest version of the dataset, consisting reviews from 1996 up to 2019.

Books: …the book didn't have a proper ending but rather a rushed attempt to conclude the story and put everyone away neatly …
Clothing: …clearly of awful quality, the design and paint was totally wrong, the mask was short and stumpy as well as slightly deformed and bent to the left …
Home: …there are no handles, and the plastic gets too hot to hold, so you have to awkwardly pour by the top …

Table 4.1: A representative sample of review snippets.

mains that share a similar textual structure and functionality but differ with respect to their vocabulary. For example, we see the review snippet from the Books domain contains an aspect ("ending") for which a negative sentiment is conveyed ("didn't have a proper"). Similarly, we find an aspect ("handle") with a corresponding conveyed sentiment ("too hot") for the Home domain. We can see this shared pattern across all domains, with different aspects and sentiment terms. We would not expect this to be the case for other datasets, which might have different differentiators for domains. For example, Amazon reviews and Wikipedia pages on Books domain may have a similar vocabulary, however, a review is more likely to convey sentiment toward a particular book, and a Wikipedia article is more likely to focus on describing the book. Thus, the Amazon Reviews dataset is an ideal testbed for our analysis.

In addition to the Amazon Reviews dataset, we experimented on the WikiSum dataset (Cohen et al., 2021) to further validate our findings. The WikiSum dataset is a coherent paragraph summarization dataset based on the WikiHow website.[2] WikiHow consists of do-it-yourself (DIY) guides for the general public, thus is written using simple English and ranges over many domains. Similar to Amazon Reviews, we also pick the top five domains for our experiments: Education, Food, Health, Home, and Pets. Since the dataset is designed for summarization, we concatenate the document and summary together for our MLM task. We present the results for this dataset at the end of § 4.4.

Task We study the language modeling task to understand the nature of multi-domain learning better. More precisely, we experiment with the masked language modeling (MLM) task, which randomly masks some of the tokens from the input, then predicts the masked word based on the context as the training objective. We

[2]https://www.wikihow.com

focus on the MLM task as it is a prevalent pre-training task for many standard models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019a) that serve as building blocks for many downstream tasks. Using examples from a set of pre-defined domains, we train a BERT model from scratch to fully control our experiment and isolate the effect of different domains. This is crucial since a pre-trained BERT model is already trained on multiple domains, hence hard to drive correct conclusions through our analysis from such a model. Moreover, recent studies (Magar and Schwartz, 2022; Brown et al., 2020) showed the risk of exposure of large language models to test data in the pre-training phase, also known as *data contamination*.

Model We use the $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019) architecture for all of our experiments. We train two types of models: the experimental model $\mathbf{E}$, trained on all five domains with the MLM objective, and the control model $\mathbf{C}_i$ for $i \in [5]$ trained on the ith domain. We are particularly interested in the effect of two aspects on the model representation: model capacity and data size. We use the capacity of 100% for $\text{BERT}_{\text{BASE}}$ size. $\text{BERT}_{\text{BASE}}$ has 768-dimensional vectors for each layer, adding up to a total of 110M parameters. We also experiment with a reduced model capacity of 75%, 50%, 25%, and 10% by reducing the dimension of the hidden layers. We follow Devlin et al. (2019) design choices, e.g., 12 layers with 12 attention heads per layer. We set the base training data size (100%) for $\mathbf{E}$ to be 50K, composed of 10K reviews per domain. Each $\mathbf{C}_i$ is trained on single domain data containing 10K reviews. $\mathbf{E}$ and $\mathbf{C}_i$ *share all the examples of domain* i. To study the effect of data size on model representation, we take subsets from the data split and create smaller datasets: a 10% split and a 50% split. We also create a 200% split to simulate the case with abundant data.

Training We set the validation data size for $\mathbf{E}$ to be 10K, which is composed of 2K reviews from each domain. For validation set of each $\mathbf{C}_i$, we use the same 2K reviews used for $\mathbf{E}$ from each domain. For consistency, we use the same validation set for all data sizes. We use a test set with 2.5K reviews for each domain. The same test set is fed to both $\mathbf{E}$ and $\mathbf{C}_i$ across all model capacities and data sizes to obtain representations for subpopulation analysis. When it is clear from the context which $\mathbf{C}_i$ for $i \in [5]$ we are referring to (and under which training regime), we use the simplification $\mathbf{C}$.

	10%d	50%d	100%d	200%d
10%m	6.052	5.541	4.788	3.886
25%m	5.764	3.257	2.745	2.354
50%m	4.366	2.758	2.451	2.144
75%m	4.017	2.781	2.435	2.149
100%m	4.012	2.786	2.436	2.16

Table 4.2: Validation cross-entropy loss on the experimental model for different model capacities and data sizes where m refers to model capacity and d refers to data size used to train the model.

All models use the validation set cross-entropy loss to perform early stopping, and we train a model for a maximum of 500 epochs. We provide the validation loss (cross-entropy) for the **E** model in Table 4.2. From the results, we can see that for fixed data size, model performance saturates when reaching model capacity of 100%. Thus, unlike data size, we do not perform further experiments with model capacity larger than 100%. All models are trained on 4 NVIDIA A100 GPUs with a batch size of 32 per GPU. We use PyTorch (Paszke et al., 2019) and the HuggingFace library (Wolf et al., 2020) for all model implementation.

4.4 Experiments and Results

Our research questions (RQs) examine how domain-specific information is encoded in **E** by calculating its SVCCA score with $\mathbf{C}_i$ for a specific i. For a given domain, we use a held-out test set for getting the experimental and control model representations as an input for the SVCCA method. Intuitively, a high SVCCA score between **E** and $\mathbf{C}_i$ indicates **E** stores domain-specific information for domain i, as $\mathbf{C}_i$ was train solely on data from domain i. A low SVCCA score between **E** and $\mathbf{C}_i$ could mean one of two things: a) **E** can generalize to data from d_i without explicitly storing domain-specific information about it, or b) **E** can not store information about $\mathbf{C}_i$, as a result of, for example, lack of model capacity. The way to distinguish between the two is subjective and depends on whether one finds **E** performance when applied to data from d_i to be satisfactory. This analysis focuses on how information is stored at the model layers. As we inspect highly complex models consisting of multiple layers, it is challenging to determine

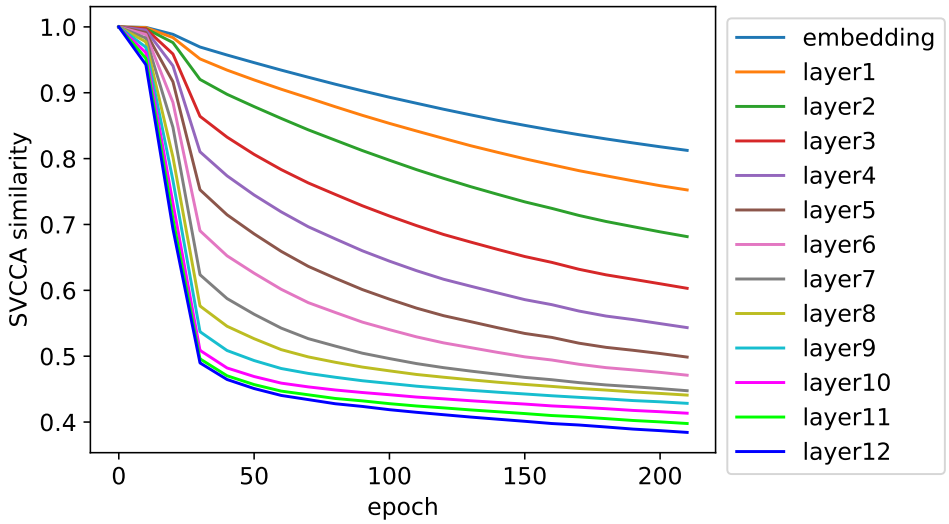


Figure 4.2: Training dynamics for all layers between **E** and $\mathbf{C}_{Books}$. Here both model and data size are 100%.

to what extent a certain layer contributes to a model's overall performance. For those reasons, when comparing equivalent layers of different models, we focus on the amount of domain-specific information encoded in **E** for a given layer. With these preliminaries in mind, we are now ready to ask the following research questions:

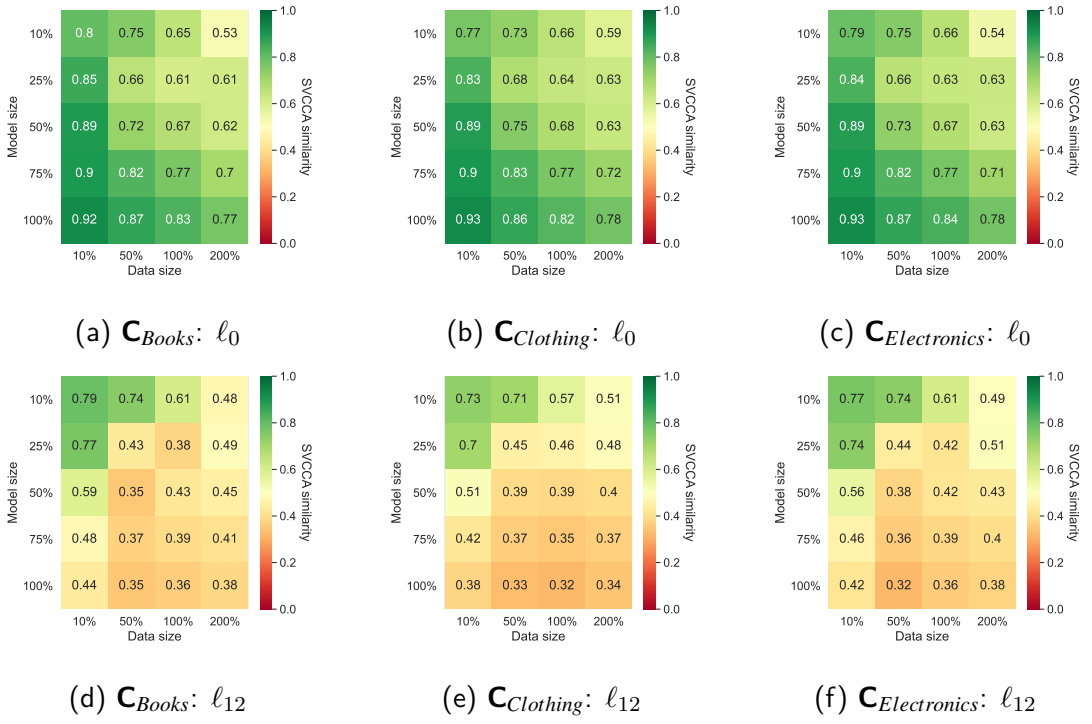
RQ1: How does the similarity between the corresponding layers in E and C evolve over training? We perform an iterative comparison between the **E** and $\mathbf{C}_i$ for each $i \in [5]$. After each epoch, we calculate the SVCCA score between corresponding layers of the models, i.e., layer j of **E** is compared to layer j of $\mathbf{C}_i$. As **E** is trained on more data points than $\mathbf{C}_i$, and both use the same batch size, for any given epoch, **E** had more weights' updates than $\mathbf{C}_i$. More precisely, after the kth epoch, $\mathbf{C}_i$ and **E** had completed k passes on data points from d_i, but **E** used additional data points from the rest of the domains. We choose this alignment to examine the effect of the additional training data drawn from other domains.

Figure 4.2 presents the training dynamics analysis for the Books domain (we denote the Books control model as $\mathbf{C}_{Books}$). We include training dynamics analyses of other control models and domains in Appendix A.1.1, as they demonstrate similar trends. Since both $\mathbf{C}_{Books}$ and **E** are initialized with the same weights, the

initial SVCCA score is 1 for all layers before training. We observe that as training progresses, the SVCCA values of higher layers (closer to the output) consistently become lower compared to the first layer. The order of SVCCA values is almost perfectly preserved with respect to the order of the layers in the network. The separation is higher for lower layers, with higher layers receiving similar SVCCA values. This is evidence that ***E*** *stores more domain-specific information in lower layers than in deeper layers throughout the training procedure.* Singh et al. (2019), who researched the nature of multilingual models, observed a similar pattern of dissimilarity in deeper layers for multilingual model representations of parallel sentences in different languages.

The alignment between the similarity of the layer pairs (**E** and **C**) and their depth also exists for models with random weights. It can be partially attributed to the mathematical artifact of decreasing correlation values for layers that are deeper because of the use of nonlinear activation units. To see to what extent this artifact plays a role in this alignment, we created ten models with random weights (no training, so there is no longer an experimental/control distinction) and calculated SVCCA between all 45 pairs for the first and last layers. We discovered that the mean difference between SVCCA scores of the first layer comparison and the last layer comparison is 0.139 (with a standard deviation of 0.001 over 45 pairs). In Figure 4.2, the difference is much larger when comparing the control model to the experimental model (0.428), indicating that the difference in layer SVCCA score cannot be only attributed to the mathematical artifact of increasing depth with more nonlinear activation. We still note that one should exercise caution when using linear methods, such as SVCCA, to analyze nonlinear processes.

The observed training dynamics motivates us to focus on the embedding layer (ℓ_0) and final layer (ℓ_{12}) for the rest of our analysis, as they serve as a lower bound (ℓ_0) and an upper bound (ℓ_{12}) with respect to the SVCCA scores of $\mathbf{C}_i$ and **E** throughout the training process. In addition, those layers have interesting attributes that we would like to explore. ℓ_0, a non-contextualized word embeddings layer, is known for encoding mainly lexical information (de Vries et al., 2020; Vulić et al., 2020). The highly contextualized ℓ_{12} is fed directly to the masked word classifier, thus playing a significant role in the MLM task. Our interest in the fully-trained models leads us to the following question:



(a) $\mathbf{C}_{Books}$: ℓ_0 (b) $\mathbf{C}_{Clothing}$: ℓ_0 (c) $\mathbf{C}_{Electronics}$: ℓ_0

(d) $\mathbf{C}_{Books}$: ℓ_{12} (e) $\mathbf{C}_{Clothing}$: ℓ_{12} (f) $\mathbf{C}_{Electronics}$: ℓ_{12}

Figure 4.3: The SVCCA scores between $\mathbf{E}$ and different $\mathbf{C}_i$s for different data sizes and model capacities. We only display for three domains here, and we provide the rest in Appendix A.1.2. The top row presents the results for the embedding layer ℓ_0, and the bottom row presents them for the last layer ℓ_{12}.

RQ2: How do data size and model capacity affect domain encoding in ℓ_0 and ℓ_{12}? To answer this question, we measure the SVCCA score between variants of $\mathbf{E}$ and their corresponding $\mathbf{C}_i$ for different domains. The variants differ with respect to two parameters, data size and model capacity.

Figure 4.3 presents our results. We observe training the model on larger datasets decreases the SVCCA scores across all model capacities and domains for both ℓ_0 and ℓ_{12}. For each data point we add to the control model, we add d data points to the general model, where $d-1$ out of them belong to other domains. This means while we keep a constant ratio between the number of datapoints for the domains, the absolute gap between a given domain and the rest of the domains is growing for larger data sizes. This might explain why adding more data points increase $\mathbf{E}$ and $\mathbf{C}$ divergence.

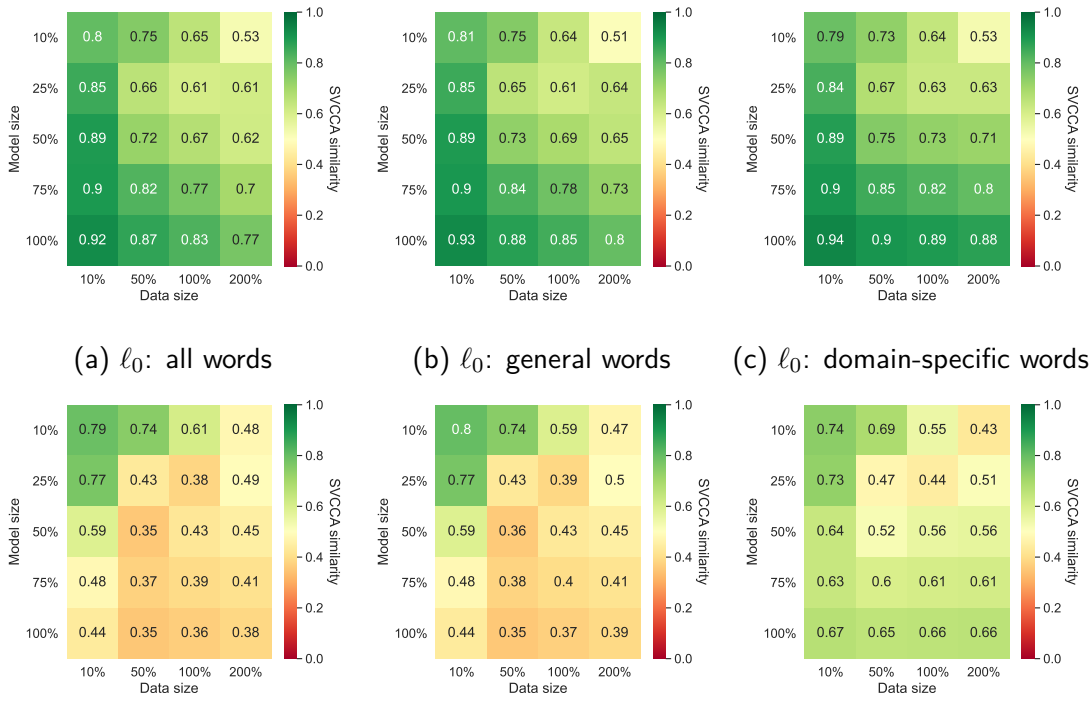
A possible explanation for these trends might be how we define domains. The Amazon reviews dataset is divided by product categories which can be seen as lexical domains (see § 4.3). More precisely, all the domains share a similar

General words: totally, preference, cost, mistake, hello, noticeable, play, factor, common, friend, previously, upon, explain, future, everyone
Books: gutenberg, appendix, autobiographical, grammatically, bookshelves, democrat, asides, arabic, stagnant, curriculum, minutiae, gripped, publishers, referencing, socialism **Clothing**: marten, docker, florsheim, rockports, skechers, buckles, 38d, fleece, nylons, insoles, tees, pantyhose, puckered, slippers, footwear **Electronics**: printable, wifi, 105mm, aux, energizer, recordable, directories, reinstall, gigabit, reboots, portability, vga, hitachi, configurations, wirelessly **Home**: cupcakes, kitchenaid, undercooked, ikea, chopper, mugs, steamers, juices, fiesta, kettles, aroma, toasted, rinsed ovens, airtight **Movie**: scenic, 16x9, nightclub, cheesiest, filmakers, supernova, serials, weepy, purists, incarnations, lionsgate, reportedly, suggestive, 1931, choreography

Table 4.3: A representative sample of general words (top row) and domain-specific words (bottom rows) taken from different categories (domains) of the dataset.

structure and writing style of Amazon product reviews. The differences lie in the vocabulary of each domain. We hypothesize that the $\boldsymbol{E}$ *uses the increased capacity to keep more domain-specific information in* ℓ_0*, where the lexical information is kept and diverges from* $\boldsymbol{C}$ *in* ℓ_{12}*, where the highly contextualized representations are stored.* As we hypothesize that our domains differ mostly with respect to their vocabularies, we refine the mentioned above experiment by raising the following research question:

RQ3: To what extent does E encode domain-specific information for domain-specific words? To shed light on the domains' lexical nature, we inspect the patterns of domain-specific and general words. Domain-specific words need to appear with at least 20 reviews in the domain in hand and no more than 10 reviews in total for the rest of the other domains. General words must appear in at least 20 reviews in each domain. Those definitions are often used in domain adaptation works to describe domain discrepancy and find adaptable features



(a) ℓ_0: all words (b) ℓ_0: general words (c) ℓ_0: domain-specific words

(d) ℓ_{12}: all words (e) ℓ_{12}: general words (f) ℓ_{12}: domain-specific words

Figure 4.4: The SVCCA score between $\mathbf{E}$ and $\mathbf{C}_{Books}$ for different subsets of tokens. The top row presents the results for the embedding layer ℓ_0, and the bottom row presents them for the last layer ℓ_{12}.

(Blitzer et al., 2007; Ziser and Reichart, 2017). We provide some examples of domain-specific and general words in Table 4.3. It is noteworthy that the union of the domain-specific and general words is not the complete vocabulary. To calculate the SVCCA scores for a subset of words, we first apply SVD to all inputs. Then we use the corresponding representations of the subset tokens to calculate the CCA similarity.

Figure 4.4 presents our results for the Books domain.[3] We present the Books domain analysis for all the words taken from RQ2 for reference (on the left-hand side of the figure). We observe high SVCCA scores for domain-specific words for ℓ_{12}. For large data sizes (100% and 200%), the trends of domain-specific words are opposite to the ones of RQ2, i.e., $\mathbf{E}$ uses the additional capacity to encode more domain-specific information. This indicates that as model capacity increases, $\mathbf{E}$ can capture similar information to $\mathbf{C}_{Books}$ for domain-specific words. This justifies the construction of large language models, mixing multiple subpopulations, as it

[3]The rest of the domains exhibit similar patterns. We provide all results in Appendix A.1.3.

demonstrates that *if the* $\boldsymbol{E}$ *model has large enough capacity, it separately creates representations for the different subpopulations that are similar to* $\boldsymbol{C}_i$ *model, which is a specialized model for a given domain.* Domain-specific words and their representations are crucial for the success of many NLP tasks, for example, Named Entity Recognition (Rocktäschel et al., 2013; Shang et al., 2018; Gu et al., 2021). We can see that the SVCCA scores for all the words and general words are almost identical. These findings make us suspect that word frequency and domain specificity are strongly connected. Indeed, we find out that the average frequency for Books domain-specific words is 75 with a median of 43. For general words, the average is 7696, and the median is 1440, making general words the main factor in the SVCCA scores for all words.

Finally, we would like to ensure the patterns we observe throughout this chapter affect the behavior of the model:

RQ4: Do the observed trends manifest in the models' behavior? We conducted two qualitative analyses to understand better if the models' behavior expresses our findings. For the first analysis, we compare MLM predictions of $\mathbf{E}$ and $\mathbf{C}$ to check whether higher SVCCA values are associated with similar word predictions. For ℓ_0, we calculate the k-nearest neighbors of the word embeddings for a given word as a proxy to make predictions. For ℓ_{12}, we follow the standard procedure by feeding the last layer representation to the final MLM classifier in BERT. Table 4.4 presents our analyses. We can see that for ℓ_0, as we increase the model capacity, we get more similar predictions for both domain-specific and general words. This finding agrees with the trend in Figure 4.3 that higher model capacity is associated with higher SVCCA similarity for ℓ_0. For ℓ_{12}, we can see that as model capacity increases, predictions for the general word becomes inconsistent, whereas, for domain-specific words, it is the opposite. This finding also agrees with our findings in RQ2 and RQ3, in which we observe the ℓ_{12} SVCCA values are decreasing for general words as we increase the model capacity and decrease for domain-specific words. We provide additional examples in Appendix A.1.4.

For the second analysis, we employ principal component analysis (PCA) to reduce the dimension of general and domain-specific representations for ℓ_0 and ℓ_{12} for both $\mathbf{E}$ and $\mathbf{C}_{Books}$. We provide visualizations in Figure 4.5. We can see that as model capacity increases, ℓ_0 representations of both general and domain-

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
blackberry	proxy	linux	mac
linux	linux	mac	linux
biologist	peer	blackberry	computers
viking	windows	vista	windows
samsung	servers	xp	xp

(a) 5-nearest neighbors for the domain-specific word **Macintosh** with i=Electronics.

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
functioning	riding	functioning	functioning
work	running	work	repair
worked	work	worked	work
playing	walking	looking	riding
responding	cleaning	works	looking

(b) 5-nearest neighbors for the general word **working** with i=Home and Kitchen.

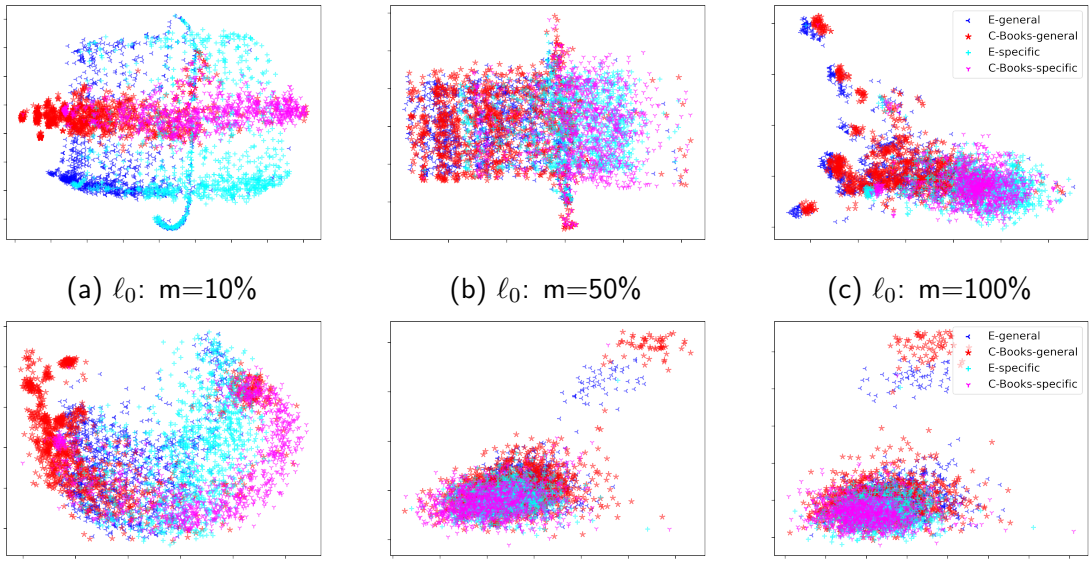
m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
networks	connections	routers	router
phones	networks	products	networks
devices	ports	systems	connections
problems	computers	mice	computers
models	cables	connections	products

(c) *Other wired and wireless [MASK] I had never had this problem.* The masked word is a domain-specific word **routers** with i=Electronics.

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
away	apart	apart	aside
apart	off	flat	apart
aside	away	short	down
downhill	downhill	out	back
asleep	asleep	off	along

(d) *Sadly, those hopes began to fall [MASK] shortly after I finished the Prologue.* The masked word is a general word **apart** with i=Books.

Table 4.4: (a) and (b) are the 5-nearest neighbors using the embedding layer weights. (c) and (d) are model predictions using last layer representations. m denotes model capacity. All models here use a data size of 100%.



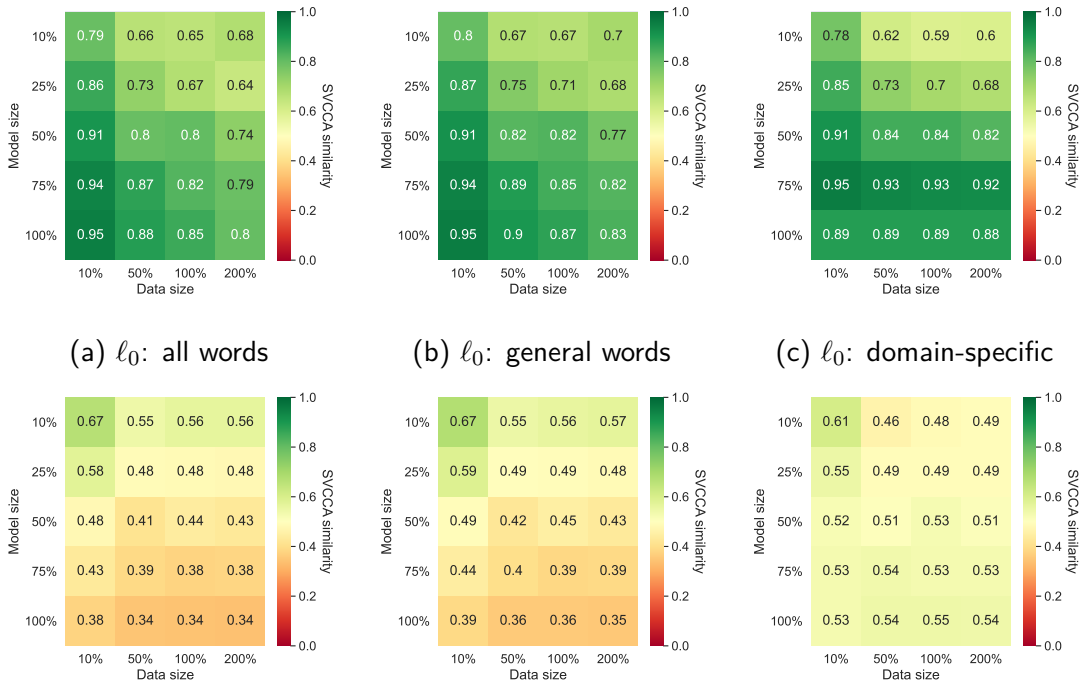
(a) ℓ_0: m=10% (b) ℓ_0: m=50% (c) ℓ_0: m=100%

(d) ℓ_{12}: m=10% (e) ℓ_{12}: m=50% (f) ℓ_{12}: m=100%

Figure 4.5: Visualization for ℓ_0 and ℓ_{12} representations for **E** and $\mathbf{C}_{Books}$. We use colors (blue/cyan for **E** and red/magenta for $\mathbf{C}_{Books}$) to separate representations for generals and domain-specific words. m denotes model capacity. All models here use a data size of 100%.

specific words from **E** and $\mathbf{C}_{Books}$ are aligned to a similar subspace. Additionally, ℓ_{12} representations of general words and domain-specific words for both models exhibit opposite behavior: domain-specific words are more aligned with increasing model capacity while general words start to detach. All of these agree with our findings in corresponding SVCCA scores trends in Figure 4.4. Even though we did not explicitly examine the relations between general and specific words in our work, we can observe that general and domain-specific word representations form different clusters in both models. Those clusters are more separated in ℓ_0 than in ℓ_{12}, suggesting that models use their increased capacity to keep more domain-specific information in ℓ_0.

WikiSum results Due to the lack of computational resources required, we only validate our main findings, namely, RQ2 and RQ3, using WikiSum. We present the results in Figure 4.6. We choose Health domain as it is the largest domain of this dataset. We observe that the trend in SVCCA scores across different scenarios on WikiSum is generally the same as those on Amazon Reviews, demonstrating that our findings are consistent.



(a) ℓ_0: all words (b) ℓ_0: general words (c) ℓ_0: domain-specific

(d) ℓ_{12}: all words (e) ℓ_{12}: general words (f) ℓ_{12}: domain-specific

Figure 4.6: The SVCCA score between $\mathbf{E}$ and $\mathbf{C}_{Health}$ on the WikiSum dataset for different subsets of tokens.

4.5 Related Work

While Chapter 2 provided a broad overview of the field, this section details the specific prior work that most directly informs the representation analysis techniques used in this particular study.

Analyzing neural representations Raghu et al. (2017) proposed SVCCA for comparing representations for the same data points from different layers and networks invariant to an affine transform. They also discovered that lower layers in a multi-layer neural network converge more quickly to their final representations in contrast to higher layers. Building off of SVCCA, Morcos et al. (2018) developed projection weighted CCA (PWCCA) using an aggregation technique. Using the SVCCA tool, Saphra and Lopez (2019) studied the learning dynamics of neural language models by probing the evolution of syntactic, semantic, and topic representations across time and models. Kudugunta et al. (2019) used SVCCA to understand massively multilingual neural machine translation representations over 100 languages. Their major findings are that encoder representations of

different languages form clusters based on their linguistic similarities.

Diagnostic Classifiers Another prominent tool for analyzing learned representations is diagnostic classifiers (DCs; Belinkov et al., 2017a,b; Giulianelli et al., 2018). DCs measure the amount of information encoded in representations about a particular task by using them as input to a classifier, which is trained on the task in a supervised manner. DC users assume that the higher their performance for this task, the more task-specific information is encoded in the representations. While widely adopted, DCs have several pitfalls. For example, Zhang and Bowman (2018) showed that learning a classifier on top of random embeddings is often competitive and, in some cases, even better than doing so with representations taken from a pre-trained model when trained on enough data. Saphra and Lopez (2019) demonstrated that, unlike SVCCA, DCs showed a stable correlation between language models and target labels throughout training epochs, in contrast to the language models' immense improvement over time.

4.6 Conclusion and Transition

In this chapter, we used the MOSSA methodology based on subpopulation analysis to understand how distinct domains are represented in a multi-domain model. Our findings show that neural models encode domain information differently across their depth, with lower layers retaining more domain-specific information, especially for domain-specific vocabulary. Crucially, we demonstrated that as model capacity increases, the generalist experimental model (**E**) learns to embed representations for domain-specific phenomena that are highly similar to those learned by a specialist control model (**C**), effectively preserving a copy of the specialist's behavior.

This study validates our core analytical framework and provides a key insight: when given sufficient capacity, generalist models learn to partition their representational space to embed specialist behaviors. The logical next step is to test the limits and generality of this finding in a more complex and challenging setting. The following chapter will therefore apply an extended version of this methodology to the problem of multilingual representation, investigating how a single model learns to handle the structural and lexical diversity of over thirty different languages.

Chapter 5

Case Study II: A Joint Analysis of Multilingual Representations

Building upon the MOSSA framework established in Chapter 3, this chapter extends the analysis to the more complex setting of multilingual learning. While the previous chapter applied MOSSA to domain-level comparisons using SVCCA, the multilingual setting, requiring the comparison of dozens of languages simultaneously, calls for a matrix decomposition method capable of handling multiple correlated datasets. To this end, we employ PARAFAC2 within the MOSSA framework, enabling joint analysis of representations across more than 30 languages. By contrasting a multilingual model with its monolingual counterparts, we examine how morphosyntactic features are encoded across languages and layers, extending the framework to capture broader representational variation. This chapter is adapted from our publication, Zhao et al. (2023).

5.1 Introduction

Pre-trained multilingual models (Conneau and Lample, 2019a; Conneau et al., 2020; Liu et al., 2020; Xue et al., 2021) have gained widespread adoption in recent years. They are initially pre-trained in many languages and subsequently fine-tuned for specific downstream tasks. Their aim is to leverage the linguistic knowledge acquired from similar languages, thereby benefiting low-resource languages and enabling zero-shot cross-lingual transfer ability. While numerous prior works have demonstrated these models have such abilities (Gerz et al., 2018; Ziser and Reichart, 2018a; Aharoni et al., 2019; K et al., 2020; Muller et al., 2021; Fujinuma

et al., 2022; Qiu et al., 2023), there are still open questions about the nature of the linguistic knowledge these models possess and the extent to which they acquire and incorporate linguistic information in their multilingual representations.

Previous work has used singular vector canonical correlation analysis (SVCCA; Raghu et al. 2017) and other similarity statistics like centered kernel alignment (CKA; Kornblith et al., 2019) to analyze multilingual representations (Singh et al., 2019; Kudugunta et al., 2019; Muller et al., 2021). However, such methods can only compare one pair of representation sets at a time. In contrast, this study analyzes all multilingual representations simultaneously using parallel factor analysis 2 (PARAFAC2; Harshman 1972), a method that allows us to factorize a set of representations jointly. More precisely, we extend the MOSSA subpopulation analysis method from Chapter 3, which compares two models as an alternative to probing: a *control model* trained on data of interest and an *experimental model*, which is identical to the control model but is trained on additional data. By treating the multilingual experimental model as a shared component in multiple comparisons with different monolingual control models, we can better analyze the multilingual representations.

As an alternative to probing, our representation analysis approach: a) enables standardized comparisons across languages within a multilingual model; b) directly analyzes model representations without auxiliary probing models; and c) compares multilingual versus monolingual representations for any inputs, avoiding reliance on labelled probing datasets.

We use PARAFAC2 to directly compare representations learned between multilingual models and their monolingual counterparts. We apply this efficient paradigm to answer the following research questions: **Q1)** How do multilingual language models encode morphosyntactic features in their layers? **Q2)** Are our findings robust in low-resource settings? **Q3)** Do morphosyntactic typology and downstream task performance reflect in the factorization outputs?

We experiment with two kinds of models, XLM-R (Conneau et al., 2020) and RoBERTa (Liu et al., 2019b). We apply our analysis tool on the multilingual and monolingual representations from these models to check morphosyntactic information in 33 languages from Universal Dependencies treebanks (UD; Nivre et al., 2017a). Our analysis reinforces recent findings on multilingual representations, such as the presence of language-neutral subspaces (Foroutan et al., 2022), and yields the following key insights:

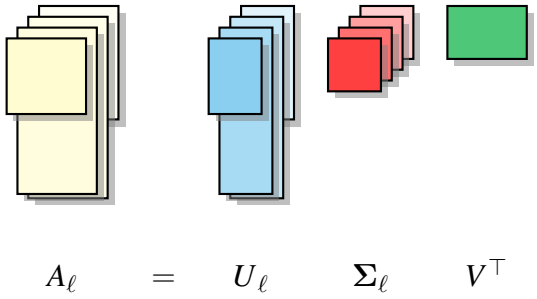


Figure 5.1: A diagram of the matrix factorization that PARAFAC2 performs. For our analysis, A_ℓ ranges over covariance matrices between multilingual model representations and a ℓ-th monolingual model's representations.

- Encoding of morphosyntactic information is influenced by language-specific factors such as writing system.
- Multilingual representations demonstrate distinct encoding patterns in subsets of languages with low language proximity.
- Representation of low-resource languages benefits from the presence of related languages.
- Our factorization method's utility reflects in hierarchical clustering within phylogenetic trees and prediction of cross-lingual task performance.

5.2 Joint Matrix Factorization for Multilingual Analysis

In this chapter, we propose to use PARAFAC2 for multilingual analysis. By jointly decomposing a set of matrices representing cross-covariance between multilingual and monolingual representations, PARAFAC2 allows us to compare the representations across languages and their relationship to a multilingual model. For an integer n, we use $[n]$ to denote the set $\{1,...,n\}$. For a square matrix $\mathbf{\Sigma}$, we denote by $\text{diag}(\mathbf{\Sigma})$ its diagonal vector.

PARAFAC2 Let ℓ index a set of matrices, such that $A_\ell = \mathbb{E}[\mathbf{X}_\ell \mathbf{Z}^\top]$, the matrix of cross-covariance between $\mathbf{X}_\ell$ and $\mathbf{Z}$, which are random vectors of dimensions d and d', respectively. This means that $[A_\ell]_{ij} = \text{Cov}([\mathbf{X}_\ell]_i, \mathrm{Z}_j)$ for $i \in [d]$ and $j \in [d']$.

PARAFAC2 on the set of matrices $\{A_\ell\}_\ell$ finds a set of transformations $\{U_\ell\}_\ell$, a shared matrix V, and a set of diagonal matrices $\{\mathbf{\Sigma}_\ell\}_\ell$ such that:

$$A_\ell \approx U_\ell \mathbf{\Sigma}_\ell V^\top.$$

We call the elements on the diagonal of $\mathbf{\Sigma}_\ell$ *pseudo-singular values*. The decomposition in Eq. 5.2 jointly decomposes the matrices such that each A_ℓ is decomposed into a sequence of three transformations: first transforming $\mathbf{Z}$ into a latent space (V), scaling it ($\mathbf{\Sigma}_\ell$) and then transforming it into a specific ℓ-th-indexed $\mathbf{X}_\ell$ space (U_ℓ). Unlike singular value decomposition, PARAFAC2 does not guarantee U_ℓ and V to be orthonormal. However, Harshman (1972) showed that a unique solution can be found by adding the constraint that $U_\ell^\top U_\ell$ is constant for all ℓ. We follow this variant, illustrated in Figure 5.1.

5.3 Experiment-Control Modeling for Multilingual Analysis

We employ factor analysis to generate a distinctive signature for a group of representations within an experimental model, contrasting them with representations from a set of control models. In our case, the experimental model is a jointly-trained multilingual PLM, and the control models are monolingual models. Formally, there is an index set of languages $[L]$ and a set of models consisting of the experimental model $\mathbf{E}$ and the control models $\mathbf{C}_\ell$ for $\ell \in [L]$.

We assume a set of inputs $\mathcal{X} = \bigcup_{\ell=1}^{L} \mathcal{X}_\ell$. Each set $\mathcal{X}_\ell = \{\mathbf{x}_{\ell,1}, \ldots, \mathbf{x}_{\ell,m}\}$ represents a set of inputs for the ℓ-th language. For each $\ell \in [L]$ and $i \in [m]$, we apply the models $\mathbf{E}$ and $\mathbf{C}_\ell$ to $\mathbf{x}_{\ell,i}$ to get two corresponding representations $\mathbf{y}_{\ell,i} \in \mathbb{R}^d$ and $\mathbf{z}_{\ell,i} \in \mathbb{R}^{d_\ell}$. Stacking these into matrices, we obtain $Y_\ell \in \mathbb{R}^{m \times d}$ and $Z_\ell \in \mathbb{R}^{m \times d_\ell}$. We calculate the covariance matrix $\mathbf{\Omega}_\ell = Z_\ell^\top Y_\ell$.

Use of PARAFAC2 Given an integer k, we apply PARAFAC2 on the set of matrices, decomposing each $\mathbf{\Omega}_\ell$ into:

$$\mathbf{\Omega}_\ell \approx U_\ell \mathbf{\Sigma}_\ell V^\top,$$

where $U_\ell \in \mathbb{R}^{d_\ell \times k}$ and $V \in \mathbb{R}^{d \times k}$.

To provide some intuition on this decomposition, consider Eq. 5.3 for a fixed ℓ. If we were following SVD, such decomposition would give two projections that

project the multilingual representations and the monolingual representations into a joint latent space (by applying U_ℓ and V on $\mathbf{z}$s and $\mathbf{y}$s, respectively). When applying PARAFAC2 jointly on the set of L matrices, we enforce the matrix V to be identical for all decompositions (rather than be separately defined if we were applying SVD on each matrix separately) and for U_ℓ to vary for each language. We are now approximating the $\mathbf{\Omega}_\ell$ matrix, which by itself could be thought as transforming vectors from the multilingual space to the monolingual space (and vice versa) in three transformation steps: first into a latent space (V), scaling it ($\mathbf{\Sigma}_\ell$), and then *specializing* it monolingually.

The diagonal of $\mathbf{\Sigma}_\ell$ can now be readily used to describe a *signature* of the ℓth language representations in relation to the multilingual model (see also Dubossarsky et al. 2020). This signature, which we mark by $\text{sig}(\ell) = \text{diag}(\mathbf{\Sigma}_\ell)$, can be used to compare the nature of representations between languages, and their commonalities in relationship to the multilingual model. In our case, this PARAFAC2 analysis is applied to different slices of the data. We collect tokens in different languages (both through a multilingual model and monolingual models) and then slice them by specific morphosyntactic category, each time applying PARAFAC2 on a subset of them.

For some of our analysis, we also use a condensed value derived from $\text{sig}(\ell)$. We follow a similar averaging approach to that used by SVCCA (Raghu et al., 2017), a popular representation analysis tool, where they argue that the single condensed SVCCA score represents the average correlation across aligned directions and serves as a direct multidimensional analogue of Pearson correlation. In our case, each signature value within $\text{sig}(\ell)$ from the PARAFAC2 algorithm corresponds to a direction, all of which are normalized in length, so the signature values reflect their relative strength. Thus, taking the average of $\text{sig}(\ell)$ provides an intensity measure of the representation of a given language in the multilingual model.

5.4 Experimental Setup

Data We use CoNLL's 2017 Wikipedia dump (Ginter et al., 2017) to train our models. Following Fujinuma et al. (2022), we downsample all Wikipedia datasets to an identical number of sequences in order to use the same amount of data for all language pre-training. The downsampled dataset is standardized to the Hindi corpus, which has the smallest size among all languages we examine. For

each language's pre-training data, there are about 30M tokens (approximately 200MB). In total, we experiment with 33 languages. We provide the full list of languages used for our experiments in Appendix B.1. We also create a validation set with 1K sequences (about 512 tokens per sequence) to measure model loss (cross-entropy) during pre-training. For morphosyntactic features, we use treebanks from UD 2.1 (Nivre et al., 2017a). These treebanks contain sentences annotated with morphosyntactic information and are available for a wide range of languages. We obtain a contextual representation for every word in the treebanks by feeding them to our multilingual/monolingual models. We then use the UniMorph schema (Kirov et al., 2018) to map each word with its parts of speech and morphosyntactic properties. We provide a list of morphosyntactic categories we use in Appendix B.1. We follow Stanczak et al. (2022) and use the converter (McCarthy et al., 2018) to switch morphosyntactic annotations from UD v2.1 to UniMorph schema.

Task For pre-training our models, we use masked language modeling (MLM). To fully control our experiments, we follow the methodology from the previous chapter (§ 4.3) and train our models from scratch.

Models We have two kinds of models: the multilingual model $\boldsymbol{E}$, trained using all $\boldsymbol{L}$ languages, and the monolingual model $\boldsymbol{C}_\ell$ for $\ell \in [\boldsymbol{L}]$ trained only using the ℓ-th language. We use the XLM-R (Conneau et al., 2020) architecture for the multilingual $\boldsymbol{E}$ model, and we use RoBERTa (Liu et al., 2019b) for the monolingual $\boldsymbol{C}_\ell$ model. We use the base variant for both kinds of models, which consists of 12 layers, 768 hidden dimensions, 8 attention heads for RoBERTa, and 12 attention heads for XLM-R. We use XLM-R's vocabulary and the SentencePiece (Kudo and Richardson, 2018) tokenizer for all our experiments provided by Conneau et al. (2020) in order to support all languages we analyze and enable fair comparison for all configurations. We do not use the original RoBERTa vocabulary and tokenizer since they only support English. We pre-train all models for a maximum of 150K steps, and all models use the validation set cross-entropy loss to perform early stopping. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4}. Our monolingual models were trained on four NVIDIA GeForce GTX 1080 Ti GPUs with a batch size of two per GPU, and our multilingual models were trained on four NVIDIA A100 GPUs with a

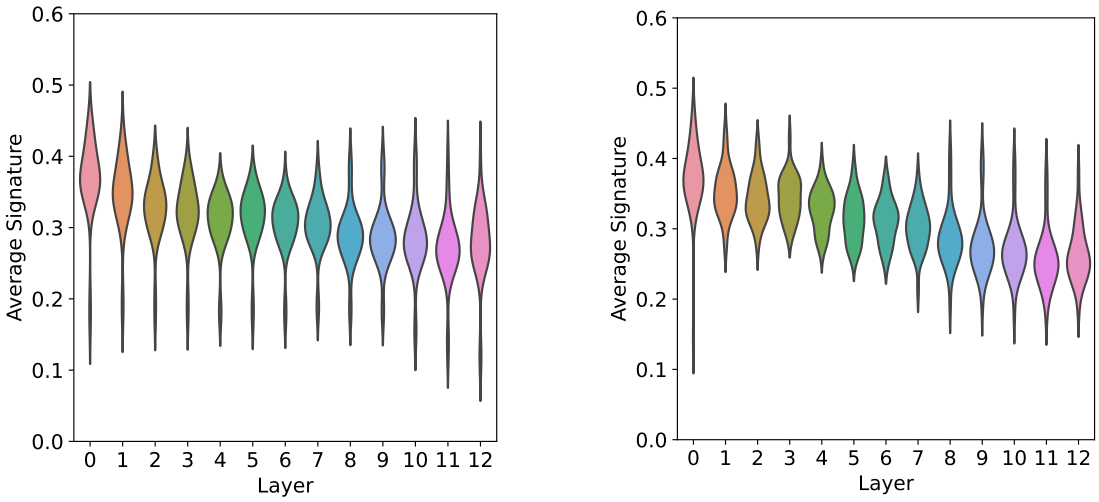
batch size of 16 per GPU. Both models take about two days to train. We use PyTorch (Paszke et al., 2019), the HuggingFace library (Wolf et al., 2020) and the TensorLy library (Kossaifi et al., 2019) for all model implementation and PARAFAC2 computation.

5.5 Experiments and Results

This section outlines our research questions (RQs), experimental design, and obtained results.

5.5.1 Morphosyntactic and Language Properties

Here, we address RQ1: *How do multilingual language models encode morphosyntactic features in their layers?* While in broad strokes, previous work (Hewitt and Manning, 2019; Jawahar et al., 2019; Tenney et al., 2019) showed that syntactic information tends to be captured in lower to middle layers within a network, we ask a more refined question here, and inspect whether different layers are specialized for specific morphosyntactic features, rather than providing an overall picture of all morphosyntax in a single layer. As mentioned in § 5.3, we have a set of signatures, $\text{sig}(\ell)$ for $\ell \in [L]$, each describing the ℓth language representation for the corresponding morphosyntactic category we probe and the extent to which it utilizes information from each direction within the rows of V. PARAFAC2 identifies a single transformation V that maps a multilingual representation into a latent space. Following that, the signature vector scales in specific directions based on their importance for the final monolingual representation it is transformed to. Therefore, the signature can be used to analyze whether similar directions in V are important for the transformation to the monolingual space. By using signatures of different layers in a joint factorization, we can identify comparable similarities for all languages. Analogous to the SVCCA similarity score (Raghu et al., 2017), we condense each signature vector into a single value by taking the average of the signature. This value encapsulates the intensity of the use of directions in V. A high average indicates the corresponding language is well-represented in the multilingual model. We expect these values to exhibit a general trend (either decreasing or increasing) going from lowers to upper layers as lower layers are more general and upper layers are known to be more



(a) Original data

(b) zh and ja romanized data

Figure 5.2: Average signature violin plots for all layers and languages on (a) original data and (b) data with Chinese (zh) and Japanese (ja) romanized.

task-specific (Rogers et al., 2020). In addition, the trend may be contrasting for different languages and morphosyntactic features.

Language Signatures Across Layers We begin by presenting the distribution of average sig(ℓ) values for all languages across all layers for all lexical tokens in Figure 5.2a. We observe a gradual decrease in the mean of the distribution as we transition from lower to upper layers. This finding is consistent with those from Singh et al. (2019), who found that the similarity between representations of different languages steadily decreases up to the final layer in a pre-trained mBERT model. We used the Mann-Kendall (MK) statistical test (Mann, 1945; Kendall, 1948) for individual languages across all layers. The MK test is a rank-based non-parametric method used to assess whether a set of data values is increasing or decreasing over time, with the null hypothesis being there is no clear trend. Since we perform multiple tests (33 tests in total for all languages), we also control the false discovery rate (FDR; at level $q = 0.05$) with corrections to the p-values (Benjamini and Hochberg, 1995). We found that all 33 languages except for Arabic, Indonesian, Japanese, Korean, and Swedish exhibit significant monotonically decreasing trends from lower layers to upper layers, with the FDR-adjusted p-values ($p < 0.05$). Figure 5.2a shows that the spread of the distribution for each layer (measured in variance) is constantly decreasing up until layer 6. From these layers forward, the spread increases again. A small spread indicates that

the average intensity of scaling from a multilingual representation to the monolingual representation is similar among all languages. This provides evidence of the multilingual model aligning languages into a *language-neutral* subspace in the middle layers, with the upper layers becoming more task-focused (Merchant et al., 2020). This result is also supported by findings of Muller et al. (2021) – different languages representations' similarity in mBERT constantly increases up to a mid-layer then decreases.

Logogram vs. Phonogram In Figure 5.2a, we observe a long bottom tail in the average sig(ℓ) plots for all languages, with Chinese and Japanese showing lower values compared to other languages that are clustered together, suggesting that our models have learned distinct representations for those two languages. We investigated if this relates to the logographic writing systems of these languages, which rely on symbols to represent words or morphemes rather than phonetic elements. We conducted an ablation study where we romanized our Chinese and Japanese data into Pinyin and Romaji,[1] respectively, and retrained our models. One might ask why we did not normalize the other languages in our experiment to use the Latin alphabet. There are two reasons for this: 1) the multilingual model appears to learn them well, as evidenced by their similar signature values to other languages; 2) our primary focus is on investigating the impact of logographic writing systems, with Chinese and Japanese being the only languages employing logograms, while the others use phonograms. Figure 5.2b shows that, apart from the embedding layer, the average sig(ℓ) are more closely clustered together after the ablation. Our findings suggest that logographic writing systems may present unique challenges for multilingual models, warranting further research to understand their computational processes. Although not further explored here, writing systems should be considered when developing and analyzing multilingual models.

Morphosyntactic Attributes Looking at individual morphosyntactic attributes, we observe that while most attributes exhibit a similar decreasing trend from lower to upper layers, some attributes, such as Comparison and Polarity, show consistent distributions across all layers. Since these attributes occur rarely in

[1]We use libraries available at: https://pypi.org/project/pypinyin/ and https://pypi.org/project/pykakasi/. We use the lazy_pinyin feature to generate Pinyins without tone marks.

our data ($< 1\%$ of tokens), it is possible that the model is only able to learn a general representation and not distinguish them among the layers. To investigate the effect of attribute frequency on our analysis, we performed a Pearson correlation analysis (for each attribute) between the average $\text{sig}(\ell)$ for all languages and their data size – the number of data points available in the UD annotations for a particular language and morphosyntactic feature. The results are shown in Figure 5.3. Our analysis of the overall dataset (all words) shows no evidence of correlation between attribute frequency and average $\text{sig}(\ell)$. However, upon examining individual categories, we observe a decrease in correlation as we move up the layers, indicating that the degree a morphosyntactic attribute is represented in the multilingual model is no longer associated with simple features like frequency but rather with some language-specific properties. This observation holds true for all categories, with the exception of Animacy, which is predominantly found in Slavic languages within our dataset. This aligns with the findings of Stanczak et al. (2022), who noted that the correlation analysis results can be affected by whether a category is typical for a specific genus. Next, we further explore the relationship between signature values and language properties.

Language Properties In addition to data size, we explore the potential relationship between language-specific properties and the average $\text{sig}(\ell)$. We consider two language properties: the number of unique characters and the type-token ratio (TTR), a commonly used linguistic metric to assess a text's vocabulary diversity.[2] TTR is calculated by dividing the number of unique words (measured in lemmas) by the total number of words (measured in tokens) obtained from the UD annotation meta-data. Typically, a higher TTR indicates a greater degree of lexical variation. We present the Pearson correlation, averaged across all layers, in Figure 5.4. To provide a comprehensive comparison, we include the results for data size as well. The detailed results for each layer can be found in Appendix B.2. Examining the overall dataset, we observe a strong negative correlation between the number of unique characters and signature values. Similarly, the TTR exhibits a similar negative correlation, indicating that higher language variation corresponds to lower signature values. When analyzing individual categories, we consistently find a negative correlation for both the number of unique characters

[2]To ensure accurate analysis, we filter out noise by counting the number of characters that account for 99.9% of occurrences in the training data. This eliminates characters that only appear very few times.

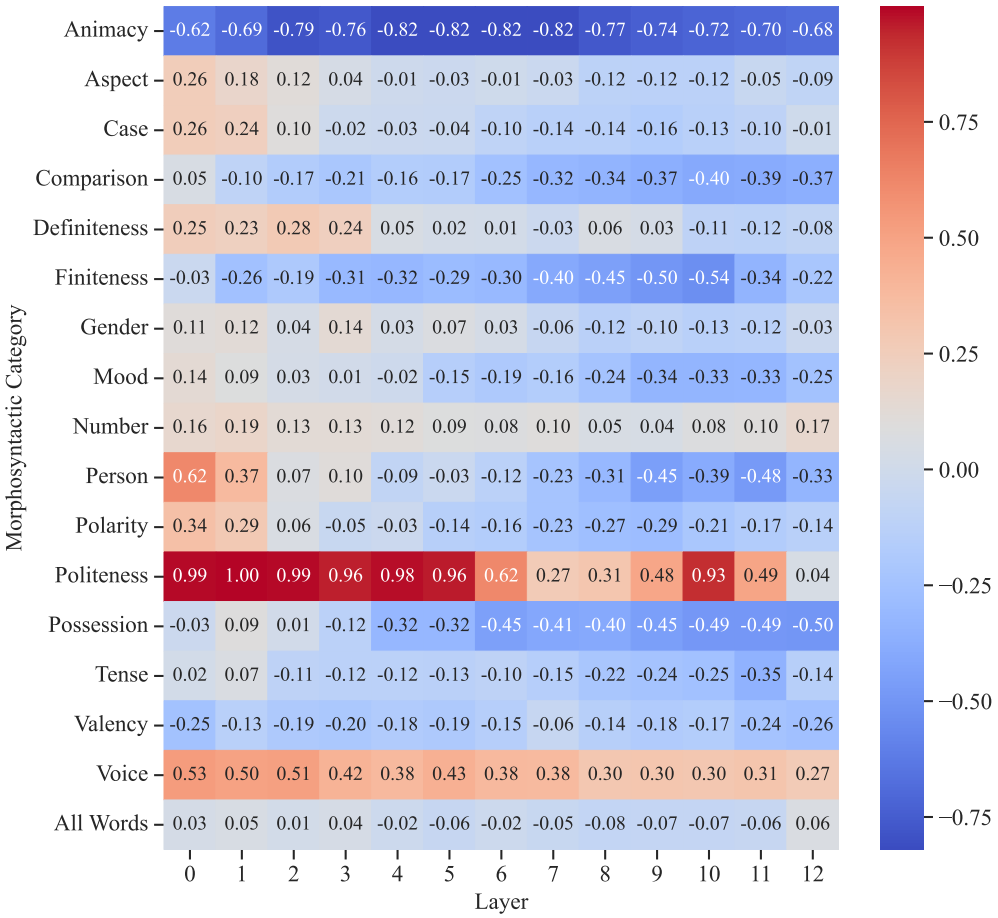


Figure 5.3: Pearson correlation results between the average $\mathrm{sig}(\ell)$ for all languages and their data size for each morphosyntactic category among all layers.

and the TTR. This further supports our earlier finding that Chinese and Japanese have lower signature values compared to other languages, as they possess a higher number of unique characters and TTR.

Generalization to Fully Pre-trained Models To ensure equal data representation for all languages in our experiment-controlled modeling, we downsampled the Wikipedia dataset and used an equal amount for pre-training our multilingual models. To check whether our findings are also valid for multilingual pre-trained models trained on full-scale data, we conducted additional experiments using a public XLM-R checkpoint.[3] The setup remained the same, except that we used representations obtained from this public XLM-R instead of our own trained XLM-R. We observe that the trends for signature values were generally similar, except for the embedding and final layers, where the values were very low. This

[3]https://huggingface.co/xlm-roberta-base

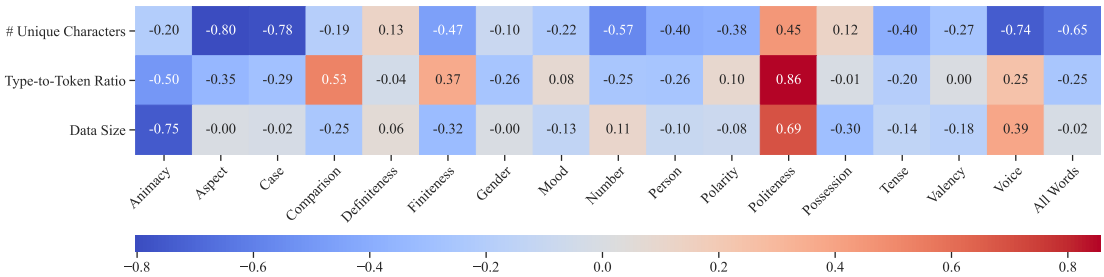


Figure 5.4: Pearson correlation results between the average $\text{sig}(\ell)$ for all languages and the number of unique characters, type-token ratio (TTR), and data size for each morphosyntactic category, averaged across all layers.

was expected, as the cross-covariance was calculated with our monolingual models. The similar trend among the middle layers further supports the idea that these layers learn language- and data-agnostic representations. Furthermore, the Pearson correlations between the number of unique characters, TTR, data size, and the average $\text{sig}(\ell)$ for the overall dataset were as follows: -0.65, -0.28, and -0.02, respectively. These values are nearly identical to those shown in Figure 5.4, confirming the robustness of our method and its data-agnostic nature.

5.5.2 Language Proximity and Low-resource Conditions

Here, we address RQ2: *Are our findings robust to address language subsets and low-resource settings?* In RQ1, our analysis was based on the full set of pre-training languages available for each morphosyntactic category we examine. In this question, we aim to explore subsets of representations derived from either a related or diverse set of pre-training languages, and whether such choices yield any alterations to the findings established in RQ1. Furthermore, we extend our analysis to low-resource settings and explore potential changes in results on low-resource languages, particularly when these languages could receive support from other languages within the same language family. We also explore the potential benefits of employing language sampling techniques for enhancing the representation of low-resource languages.

Language Proximity We obtain the related set of languages by adding all languages that are from the same linguistic family and genus (full information available in Appendix B.1). In total, we obtained three related sets of languages: Germanic languages, Romance languages, and Slavic languages. There are other

related sets, but we do not include them in our experiment since the size of those sets is very small. For the diverse set of languages, we follow Fujinuma et al. (2022) and choose ten languages from different language genera that have a diverse set of scripts . These languages are Arabic, Chinese, English, Finnish, Greek, Hindi, Indonesian, Russian, Spanish, and Turkish. We use the χ^2-square variance test to check whether the variance of the diverse set's average signatures from a particular layer is statistically significant from the variance of that of the related set, given a morphosyntactic category. We test layers 0 (the embedding layer), 6, and 12, covering the lower, middle, and upper layers within the model. We first find that for the overall dataset, the variance of the diverse set average signatures is significantly different (at $\alpha = 0.05$) from all three related set variances for all three layers. This suggests that, in general, multilingual representations are encoded differently for different subsets of languages with low language proximity. For the attributes of number, person, and tense, the variance within the diverse set significantly differs from the variances within the three related sets across all three layers, with a statistical significance level of $\alpha = 0.05$. This finding is sensible as all these three attributes have distinctions in the diverse set of languages. For example, Arabic has dual nouns to denote the special case of two persons, animals, or things, and Russian has a special plural form of nouns if they occur after numerals. On the other hand, for attributes like gender, we do not witness a significant difference between the diverse set and related set since there are only four possible values (masculine, feminine, neuter, and common) in the UD annotation for gender. We speculate that this low number of values leads to low variation among languages, thus the non-significant difference. This finding concurs with Stanczak et al. (2022), who observed a negative correlation between the number of values per morphosyntactic category and the proportion of language pairs with significant neuron overlap. Hence, the lack of significant differences in variance between the diverse and related sets can be attributed to the substantial overlap of neurons across language pairs.

Low-resource Scenario In order to simulate a low-resource scenario, we curtailed the training data for selected languages, reducing it to only 10% of its original size. The choice of low-resource languages included English, French, Korean, Turkish, and Vietnamese. English and French were selected due to the availability of other languages within the same language family, while the re-

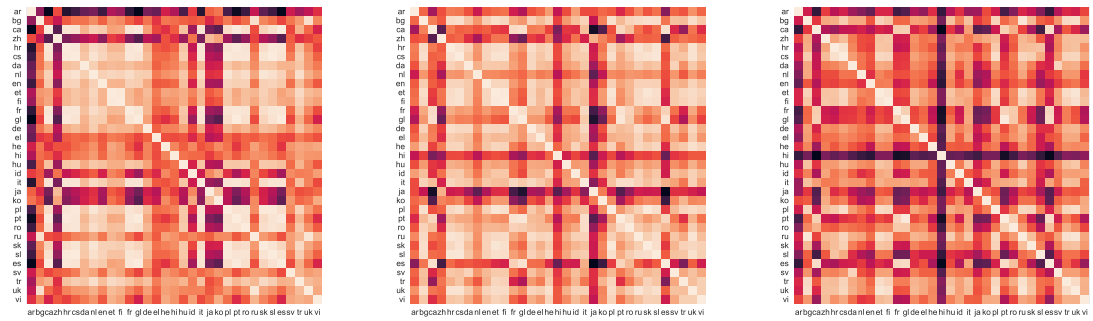
maining languages were chosen for their absence of such familial relationships. Notably, Korean was specifically selected as it utilizes a distinct script known as Hangul. To examine the impact of low-resource conditions on each of the selected languages, we re-trained our multilingual model, with each individual language designated as low-resource. To address potential confounding factors, we also re-trained monolingual models on the reduced dataset. Additionally, we explored a sampling technique (Devlin, 2019) to enhance low-resource languages.

Adhering to the current standard practice of language sampling during pre-training of multilingual models, we also experimented a setting inspired by the approach described by Devlin (2019). Following their approach, we applied a sampling technique to boost the representation of lower-resource languages. This involved sampling examples based on the probability $P(L) \propto |L|^{\alpha}$, where $P(L)$ represents the probability of selecting text from a given language during pre-training, and $|L|$ denotes the number of examples available in that language. For our study, we set the value of α to 0.3.

Our analysis reveals the impact of low-resource conditions on signature values. English and French, benefiting from languages within the same language family, exhibit minimal changes in signature values, indicating a mitigation of the effects of low-resource conditions on language representation. Remaining languages without such support experience a significant decline in signature values (dropping from 0.3 to nearly 0), particularly after the embedding layer. This implies that low-resource languages struggle to maintain robust representations without assistance from related languages. Additionally, our findings suggest that language sampling techniques offer limited improvement in signature values of low-resource languages.

5.5.3 Utility of Our Method

Here we address RQ3: *Do morphosyntactic typology and downstream task performance reflect in the factorization outputs?* Having conducted quantitative analyses of our proposed analysis tool thus far, our focus now shifts to exploring the tool's ability to unveil morphosyntactic information within multilingual representations and establish a relationship between the factorization outputs and downstream task performance. To investigate these aspects, we conduct two additional experiments utilizing the signature vectors obtained from our analysis



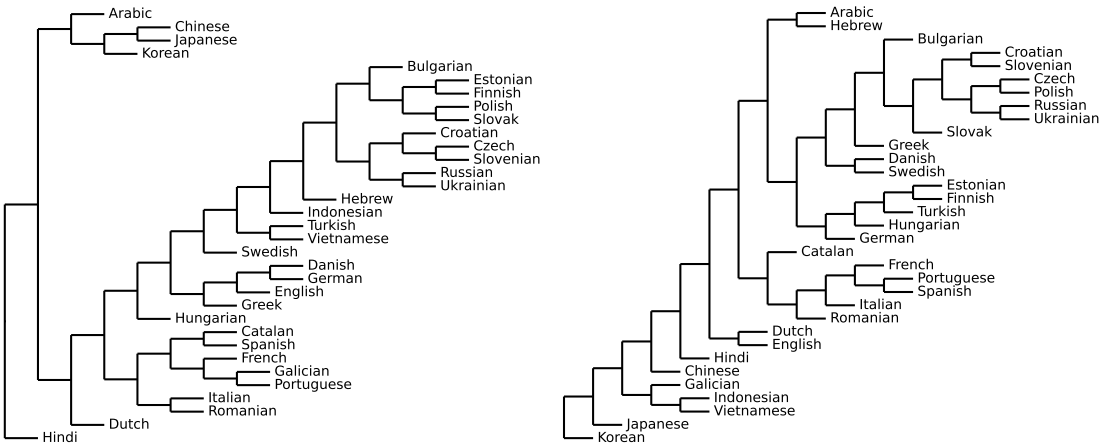
(a) Layer 0 (b) Layer 6 (c) Layer 12

Figure 5.5: Cosine distance matrices between all language pairs and their signature vectors based on overall representations obtained from layer 0, 6 and 12. Darker color indicates the cosine distance being close to 1.

tool. Firstly, we construct a phylogenetic tree using cosine distance matrices of all signature vectors. Subsequently, we examine the correlations between the results of the XTREME benchmark (Hu et al., 2020) and the sig(ℓ) values.

Phylogenetic Tree We first compute cosine distance matrices using all signature vectors for all 33 languages and 12 layers for each morphosyntactic attribute. Then, from the distance matrix, we use an agglomerative (bottom-up) hierarchical clustering method: unweighted pair group method with arithmetic mean (UPGMA; Sokal and Michener, 1958) to construct a phylogenetic tree. We show the distance matrices between all language pairs and their signature vectors based on overall representations obtained from layers 0, 6 and 12 in Figure 5.5. We can observe that signatures for Arabic, Chinese, Hindi, Japanese, and Korean are always far with respect to those for other languages across layers. From the distance matrix, we construct a phylogenetic tree using the UPGMA cluster algorithm. We present our generated trees from layer 6's matrix in Figure 5.6a.

There is ongoing discussion over the specifics of the linguistic evolutionary phylogenetic tree of languages, and a tree model has limitations because not all evolutionary connections are fully hierarchical, and it is difficult to account for horizontal transmissions (Singh et al., 2019). Despite this, we can still see that the constructed phylogenetic tree closely matches the language tree that linguists created to describe the links and development of human languages. We can see that generally, Germanic, Romance, and Slavic languages are clustered in different sub-trees. In particular, West Slavic languages, South Slavic languages, and East



(a) Using overall representations (from POS tags).

(b) Averaged over all morphosyntactic categories.

Figure 5.6: Phylogenetic trees of languages based on the distance between signature vectors for all languages.

Slavic languages are generally clustered together before being combined into the common Slavic language family. Also, Eastern Romance language Romanian are merged together with Western Romance languages to form the Romance language family cluster. Similar to the findings of Singh et al. (2019), we also observe that trees generated across different layers are generally similar. They may have different structures as the branching of the tree may differ, but languages within the same family or genus are also close in the tree.

So far, we have constructed trees based on the full slice of the data using representations from the PoS attribute. We also tried to generate trees using all other morphosyntactic attributes. However, since for most morphosyntactic attributes, some languages are always missing, i.e., the attribute is not available in that language, we construct an "average" tree by using the average distance matrices from all morphosyntactic categories except for PoS. If a language is missing in a category, we assign a distance of 1 to all other languages. We provide this average tree in Figure 5.6b. In this average tree, we observe a better fit, for example, Arabic and Hebrew are now in the same branch, and Chinese and Japanese are in their own branch as they belong to language families that are distinct from all other languages.

Performance Prediction To establish a robust connection between our factorization outputs and downstream task performances, we conducted an analysis using

Task	Dataset	#L	Metric	mBERT	XLM	XLM-R	MMTE
Cls.	XNLI (Conneau et al., 2018)	12	Acc.	.36	.30	.36	.21
	PAWS-X (Yang et al., 2019)	7	Acc.	.67	.65	.75	.69
Stru.	POS (Nivre et al., 2017b)	22	F1	.36	.36	.66	.40
	NER (Pan et al., 2017)	22	F1	.46	.46	.55	.46
QA	XQuAD (Artetxe et al., 2020)	10	F1/EM	.60/.35	.81/.56	.73/.45	.72/.61
	MLQA (Lewis et al., 2020b)	7	F1/EM	.23/.31	.46/.48	.64/.68	.28/-
	TyDiQA-GoldP (Clark et al., 2020a)	6	F1/EM	.41/.05	.43/.43	.46/.46	.66/.45
Ret.	BUCC (Zweigenbaum et al., 2017)	4	F1	.72	.96	.83	.63
	Tatoeba (Artetxe and Schwenk, 2019)	21	Acc.	.15	.24	.28	-

Table 5.1: Pearson correlations between final layer's $\mathrm{sig}(\ell)$ and XTREME benchmark performances on various tasks. #L indicates the number of languages, Cls. denotes classification, Stru. denotes structured prediction, and Ret. denotes retrieval.

the XTREME benchmark, which includes several models: mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019b), XLM-R, and MMTE (Arivazhagan et al., 2019). This benchmark encompasses nine tasks that span four different categories: classification, structured prediction, question answering, and retrieval. These tasks demand reasoning on multiple levels of meaning. To evaluate the relationship between the metrics of each task and our average $\mathrm{sig}(\ell)$ across all available languages for that task, we calculated the Pearson correlation. For each task's performance metrics, we use the results reported by Hu et al. (2020). The obtained correlation values using signature values from the last layer are presented in Table 5.1, along with pertinent details about each task, such as the number of available languages, and the metrics employed. For a comprehensive analysis, we also provide results using $\mathrm{sig}(\ell)$ from every layer in Appendix B.3.1. Observing the results, it becomes evident that the XLM-R model exhibits the highest correlation, which is expected since the $\mathrm{sig}(\ell)$ values obtained from our factorization process are also computed using the same architecture. Furthermore, for most tasks, the highest correlation is observed with the final layers, which is reasonable considering their proximity to the output. Notably, we consistently observe high correlation across all layers for straightforward tasks like POS and PAWS-X operating on the representation level. However, for complex reasoning tasks like XNLI, only the final layer achieves reasonable correlation. These results suggest that the factorization outputs can serve as a valuable indicator of performance for downstream tasks, even without the need for fine-tuning or the availability of task-specific data.

5.6 Related Work

While Chapter 2 provided a broad overview of representation analysis, this section focuses on prior work most relevant to the multilingual setting of this chapter.

Understanding the information within NLP models' internal representations has drawn increasing attention in the community. Singh et al. (2019) applied canonical correlation analysis (CCA) on the internal representations of a pre-trained mBERT and revealed that the model partitions representations for each language rather than using a shared interlingual space. Kudugunta et al. (2019) used SVCCA to investigate massively multilingual Neural Machine Translation (NMT) representations and found that different language encoder representations group together based on linguistic similarity. Libovický et al. (2019) showed that mBERT representations could be split into a language-specific component and a language-neutral component by centering mBERT representations and using the centered representation on several probing tasks to evaluate the language neutrality of the representations. Similarly, Foroutan et al. (2022) employed the lottery ticket hypothesis to discover sub-networks within mBERT and found that mBERT is comprised of language-neutral and language-specific components, with the former having a greater impact on cross-lingual transfer performance. Muller et al. (2021) presented a novel layer ablation approach and demonstrated that mBERT could be viewed as the stacking of two sub-networks: a multilingual encoder followed by a task-specific language-agnostic predictor.

Probing (see Belinkov 2022 for a review) is a widely-used method for analyzing multilingual representations and quantifying the information encoded by training a parameterized model, but its effectiveness can be influenced by model parameters and evaluation metrics (Pimentel et al., 2020). Choenni and Shutova (2020) probed representations from multilingual sentence encoders and discovered that typological properties are persistently encoded across layers in mBERT and XLM-R. Liang et al. (2021) demonstrated with probing that language-specific information is scattered across many dimensions, which can be projected into a linear subspace. Intrinsic probing, on the other hand, explores the internal structure of linguistic information within representations (Torroba Hennigen et al., 2020). Stanczak et al. (2022) conducted a large-scale empirical study over two multilingual pre-trained models, mBERT, and XLM-R, and investigated whether morphosyntactic information is encoded in the same subset of neurons in different

languages. Their findings reveal that there is considerable cross-lingual overlap between neurons, but the magnitude varies among categories and is dependent on language proximity and pre-training data size. Other methods, such as matrix factorization techniques, are available for analyzing representations (Raghu et al., 2017; Morcos et al., 2018; Kornblith et al., 2019) and even modifying them through model editing (Olfat and Aswani, 2019; Shao et al., 2023a; Kleindessner et al., 2023; Shao et al., 2023b). When applied to multilingual analysis, these methods are limited to pairwise language comparisons, whereas our proposed method enables joint factorization of multiple representations, making it well-suited for multilingual analysis.

5.7 Conclusion and Transition

In this chapter, we extended the MOSSA framework to a large-scale multilingual setting, applying it to representations from 33 languages. Within this framework, we employed PARAFAC2 as the joint matrix factorization method to handle the heterogeneity of multilingual data. Our findings show that the encoding of morphosyntactic information varies across layers and is influenced by language properties such as writing systems. The resulting factors were interpretable, correlated with cross-lingual task performance, and enabled the reconstruction of plausible phylogenetic trees. This study demonstrates that the MOSSA framework can be effectively scaled to complex multilingual settings through the integration of more expressive analytical methods like PARAFAC2.

Having established that generalist models create specialized representations for both domains (Chapter 4) and languages (this chapter), we now turn to the final and most contemporary challenge: massively multi-task learning. The next chapter will adapt our framework one last time to investigate instruction-tuned LLMs. We will analyze how a single model represents dozens of distinct NLP tasks, seeking to uncover the principles that govern task-specific adaptation in the modern large language model paradigm.

Chapter 6

Case Study III: Specialization in Instruction-Tuned LLMs

The preceding chapters introduced and refined the MOSSA framework, demonstrating its utility in analyzing representational subpopulations across domains and languages. This chapter, adapted from our publication, Zhao et al. (2024b), extends MOSSA to its most contemporary setting: massively multi-task learning in instruction-tuned LLMs. Using CKA as the similarity metric, we analyze how a single instruction-tuned model represents over 60 distinct NLP tasks. This study aims to uncover the principles underlying task specialization within instruction-tuned LLMs, completing our exploration of how generalist models organize and differentiate heterogeneous information.

6.1 Introduction

While pre-trained LLMs exhibit impressive performance across diverse tasks and demonstrate remarkable generalization capabilities (Brown et al., 2020; Wei et al., 2022b; Touvron et al., 2023; Chowdhery et al., 2023; OpenAI et al., 2024), the representations they learn and the task-specific information encoded during pre-training remain largely opaque and unexplored.

Recent research has investigated fine-tuning strategies to adapt LLMs to specific tasks, including supervised fine-tuning on task-specific datasets and instruction tuning (Mishra et al., 2022; Chung et al., 2022; Sanh et al., 2022). While these approaches have shown promising results in tailoring LLMs for improved task performance, a comprehensive understanding of their impact on the learned

representations is still lacking.

In this study, we perform a set of analyses to investigate task-specific information encoded in pre-trained LLMs and the effects of instruction tuning on their representations. The analysis leverages the MOSSA sub-population analysis technique introduced in Chapter 3, which provides an alternative to traditional probing methods for analyzing model representations within specific sub-populations of the training data. MOSSA involves comparing two models: a *control* model trained on the data relevant to the sub-population of interest (e.g., a particular task), and an *experimental* model that is identical to the control model but is also trained on additional data from different sources (e.g., multiple tasks). By analyzing the representational differences between these models, we can isolate the task-specific information encoded within the control model for the sub-population of interest.

To compare the representations learned by different LLM variants, we leverage the Center Kernel Alignment (CKA; Kornblith et al., 2019) metric. CKA measures the alignment between representations in a kernel space, providing a robust measure of similarity that is insensitive to scaling and centering. By using MOSSA and CKA, we investigate the following research questions:

1. To what extent are different NLP tasks already encoded in pre-trained LLMs?

2. In what ways does instruction tuning modify the representational landscape of LLMs?

3. Do the representational effects of instruction tuning generalize to unseen tasks?

Through an extensive analysis spanning over 60 diverse NLP tasks following the Flan framework (Longpre et al., 2023), we shed light on the underlying mechanisms that govern the encoding and adaptation of task-specific information within LLMs under instruction tuning. A key finding of our work is the identification of three functional groups of layers: a) shared layers, in which more general information is learned and shared across tasks; b) transition layers, in which task-specific information is intensified; c) refinement layers, in which the LLMs continue to refine their representations towards task-specific predictions. Our findings contribute to a deeper understanding of the inner workings of LLMs and

hold promising implications for future research in parameter-efficient fine-tuning (PEFT), multi-task learning (MTL), and model compression.

We structure this chapter as follows: § 6.2 describes our methodology for our analysis, while § 6.3 outlines the experimental setup and tools used to train and analyze our LLMs. § 6.4 then attempts to answer each of the research questions outlined above by presenting and analyzing our results. Finally, in § 6.5, we summarize our key findings and discuss their potential implications.

6.2 Methodology

We use the MOSSA framework introduced in Chapter 3. Unlike standard probing methods (Belinkov et al., 2017a,b; Giulianelli et al., 2018), which build a model to predict a downstream task for quantifying encoded information, MOSSA compares representations from two models: a control model trained on data of interest and an experimental model trained on additional data from different sources. Here, the data of interest refers to tasks. Probing methods, while useful, can be limited because they rely on different metrics to evaluate performance across various tasks, making it challenging to directly compare the amount of information stored about tasks as diverse as sentiment analysis and translation. MOSSA, on the other hand, circumvents this issue by comparing the latent representations of models rather than their downstream performance metrics. MOSSA calculates the similarity between the representations of the control and experimental models, thus representing the information captured from the relevant sub-population of data through their latent representations. By comparing different models to each other, we can learn what information is captured when a subset of the data is used versus the whole dataset.

We use matrix analysis to compare representation similarity between the experimental model, such as pre-trained, instruction-tuned, and corresponding single-task control models trained on individual tasks. Intuitively, a high similarity between the experimental and control models indicates the experimental model stores task-specific information learned by the control model, which was fine-tuned solely on data from that task. The similarity is measured using the CKA metric, which quantifies the similarity between two representations in a kernel space.

Formally, let $[\boldsymbol{T}]$ be an index set of tasks, and let $\mathbf{E}$ be the experimental

model and $\mathbf{C}_t$ be the control model for task $t \in [T]$. We assume a set of inputs $\mathcal{X} = \bigcup_{t=1}^{T} \mathcal{X}_t$, where each $\mathcal{X}_t = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,n}\}$ represents a set of input instructions for task t. For simplicity, we assume that all sets have the same size n, although this is not a strict requirement.[1]

For each $t \in [T]$ and $i \in [n]$, we apply the experimental model $\mathbf{E}$ and the control model $\mathbf{C}_t$ to the input instruction $\mathbf{x}_{t,i}$ to obtain two corresponding representations $\mathbf{y}_{t,i} \in \mathbb{R}^d$ and $\mathbf{z}_{t,i} \in \mathbb{R}^{d_t}$, respectively. Here, d is the dimension of the experimental model representations, and d_t is the dimension of the control model representations for task t. To obtain the representations $\mathbf{y}_{t,i}$ and $\mathbf{z}_{t,i}$, we use the last token representation following previous work (Qiu et al., 2024; Wang et al., 2024), as LLMs are decoder-only and the last token captures all input information. These representations can be extracted from any layers of the respective models.

By stacking these vectors into two matrices for each task t, we obtain the paired matrices $Y_t \in \mathbb{R}^{n \times d}$ and $Z_t \in \mathbb{R}^{n \times d_t}$. We calculate the CKA value between Y_t and Z_t following the procedure:

- Computing the kernel matrices $K_{Y_t} \in \mathbb{R}^{n \times n}$ and $K_{Z_t} \in \mathbb{R}^{n \times n}$ for Y_t and Z_t, respectively, using the same kernel function (e.g., linear, Gaussian, or polynomial).[2]

- Centering the kernel matrices by $K_{Y_t} = K_{Y_t} - \frac{1}{n}\mathbf{1}K_{Y_t} - \frac{1}{n}K_{Y_t}\mathbf{1} + \frac{1}{n^2}\mathbf{1}K_{Y_t}\mathbf{1}$, and similarly for K_{Z_t}, where $\mathbf{1}$ is a matrix of ones.

- Computing the CKA value by first computing the Frobenius inner product of the centered Gram matrices: $\mathrm{HSIC}(K_{Y_t}, K_{Z_t}) = \mathrm{Tr}(K_{Y_t}^{\top} K_{Z_t})$, where Tr denotes the trace of a matrix. Then the CKA value is normalized:

$$\mathrm{CKA}(Y_t, Z_t) = \frac{\mathrm{HSIC}(K_{Y_t}, K_{Z_t})}{\sqrt{\mathrm{HSIC}(K_{Y_t}, K_{Y_t}) \cdot \mathrm{HSIC}(K_{Z_t}, K_{Z_t})}}.$$

While other similarity metrics like SVCCA (Raghu et al., 2017) exist, they have a limitation due to the constraint of being invariant to invertible linear transformations, which requires the number of data points to be greater than the number of representation dimensions. We use CKA as it has shown robust results

[1] In our actual experimental setup for this work, we use different dataset sizes for each task, which reflects real-world scenarios. For more details, please refer to § 6.3.

[2] For a linear kernel, which we use in our experiments, $K_{Y_t} = Y_t Y_t^{\top}$, and $K_{Z_t} = Z_t Z_t^{\top}$.

when the data sample is smaller (Kornblith et al., 2019), as is sometimes the case for datasets used in this study.

Our method provides an approach to quantify the task-specific information encoded in the representations of LLMs. By comparing the experimental model's representations with those of single-task control models, we can gain insights into the extent to which the experimental model captures task-specific knowledge and how this knowledge is distributed across its representations.

6.3 Experimental Setup

Data We use the Flan 2021 dataset (Wei et al., 2022a) to fine-tune our LLMs. The Flan dataset is a comprehensive collection of more than 60 NLP datasets, including both language understanding and generation tasks. These datasets are organized into twelve task clusters, where datasets within a given cluster belong to the same task type. To enhance instruction diversity, we follow the approach of Wei et al. (2022a) and use ten unique natural language instruction templates for each dataset. These templates provide varying descriptions of the task to be performed. Our instruction tuning pipeline combines all datasets and randomly samples from each dataset during training. To mitigate the impact of dataset size imbalances, we limit the number of training examples per task cluster to 50k and use the examples-proportional mixing scheme (Raffel et al., 2020) with a mixing rate maximum of 3,000 per task. This means that no task receives additional sampling weight for examples in excess of 3,000. We provide further details about the dataset in Appendix C.1.

Models We have two types of models: the experimental model $\mathbf{E}$, fine-tuned using all T available tasks, and the single-task model $\mathbf{C}_t$ for $t \in [T]$, fine-tuned only on the t-th task. In some experiments, the model $\mathbf{E}$ can also be the pre-trained model. We use the Llama 2 models (Touvron et al., 2023) as the starting training checkpoint for both $\mathbf{E}$ and $\mathbf{C}_t$. Specifically, we use the 7B variant, which consists of 32 layers and 4096 hidden dimensions. This model allows us to conduct a more comprehensive set of experiments while maintaining control over experimental conditions. Since we have over 60 control models, exploring larger models or different families would have been computationally infeasible due to resource constraints. Given these limitations, we choose to fully explore a realistic multi-

task scenario, involving more than 60 different tasks, with the aim of extracting significant findings that we expect to generalize to other models.

Training We use LoRA (Hu et al., 2022) for fine-tuning our LLMs, with the rank r set to 8. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 5×10^{-5} for fine-tuning the instruction dataset. We use the same vocabulary, tokenizer, and learning rate scheduler for Llama 2-7B as in Touvron et al. (2023). We train the multi-task model $\mathbf{E}$ (which we refer to as Llama 2-SFT in our experiment) for a maximum of 100K steps and the single-task models $\mathbf{C}_t$ for a maximum of 10K steps, using validation set cross-entropy loss for early stopping. Our multi-task models are trained on four NVIDIA A100 GPUs with a batch size of 16 per GPU, while single-task models are trained on one NVIDIA A100 GPU with a batch size of 16. We use PyTorch (Paszke et al., 2019), the HuggingFace library (Wolf et al., 2020), and the LLaMA-Factory library (Zheng et al., 2024) for all model implementations and LoRA fine-tuning.

6.4 Experiments and Results

To shed light on the underlying mechanisms of MTL (Caruana, 1997) in LLMs, we start by examining what NLP tasks are encoded in the pre-trained LLM representations, establishing a baseline for comparison with the instruction-tuned model (§ 6.4.1). Then, using matrix analysis methods, we contrast the representational properties of the pre-trained and instruction-tuned LLMs to understand the effects of instruction tuning (§ 6.4.2, § 6.4.3, and § 6.4.4). Finally, we evaluate the generalization of our findings to unseen tasks (§ 6.4.5).

6.4.1 Task Information in Pre-trained LLMs

To identify task-relevant information in pre-trained LLMs, we compared representations from the pre-trained Llama 2 model with task-specific fine-tuned models ($\{\mathbf{C}_t\}_t$). Figure 6.1 shows the distribution of CKA similarities across all tasks and layers for the Llama 2 model. The CKA similarities between pre-trained Llama 2 and control models generally decrease through higher layers.

Llama 2 maintains high CKA similarities in earlier layers, and since CKA compares against control models fine-tuned on individual tasks, this suggests that representational changes in the earlier layers are minimal across tasks. However,

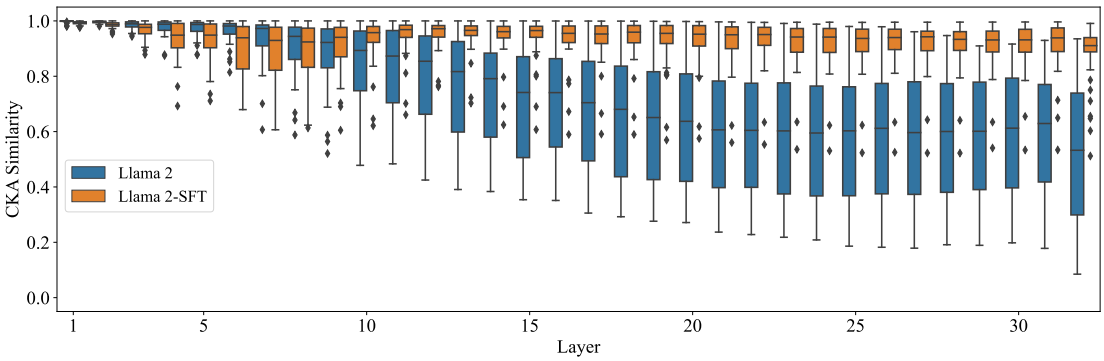


Figure 6.1: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model. The boxplots illustrate the spread and variation of CKA similarities between each model and the control models across different tasks. The comparison between the two models highlights the impact of instruction tuning on shaping task-specific representations in different layers.

we observe widespread variance in CKA values across different tasks in the middle and higher layers, suggesting that some tasks are better captured in the Llama 2 model representations than others.

To gain a more fine-grained understanding, we analyzed the CKA results at the task cluster level, where each cluster consists of a group of similar tasks. The Flan dataset organizes tasks into 12 different clusters, detailed in Appendix C.1. We present CKA results for a selection of representative clusters in Figure 6.2, with the full results provided in Appendix C.2.2.

For clusters like closed-book QA, commonsense reasoning, paraphrase detection, and sentiment analysis, which heavily rely on general linguistic and semantic understanding, the CKA similarity for Llama 2 is high. This indicates that pre-trained models already encode these tasks well in their representations. Conversely, for clusters like coreference resolution, reading comprehension, structured data to text generation, summarization, and translation, which require specialized, structured, or domain-specific knowledge involving complex transformations or extended context management, the CKA similarities are low, suggesting that next token prediction at pre-training is insufficient for encoding these tasks.

6.4.2 Impact of Instruction Tuning

Mapping Layers to Their Functionality To investigate how instruction tuning affects the representations learned by LLMs, we compared the instruction-tuned

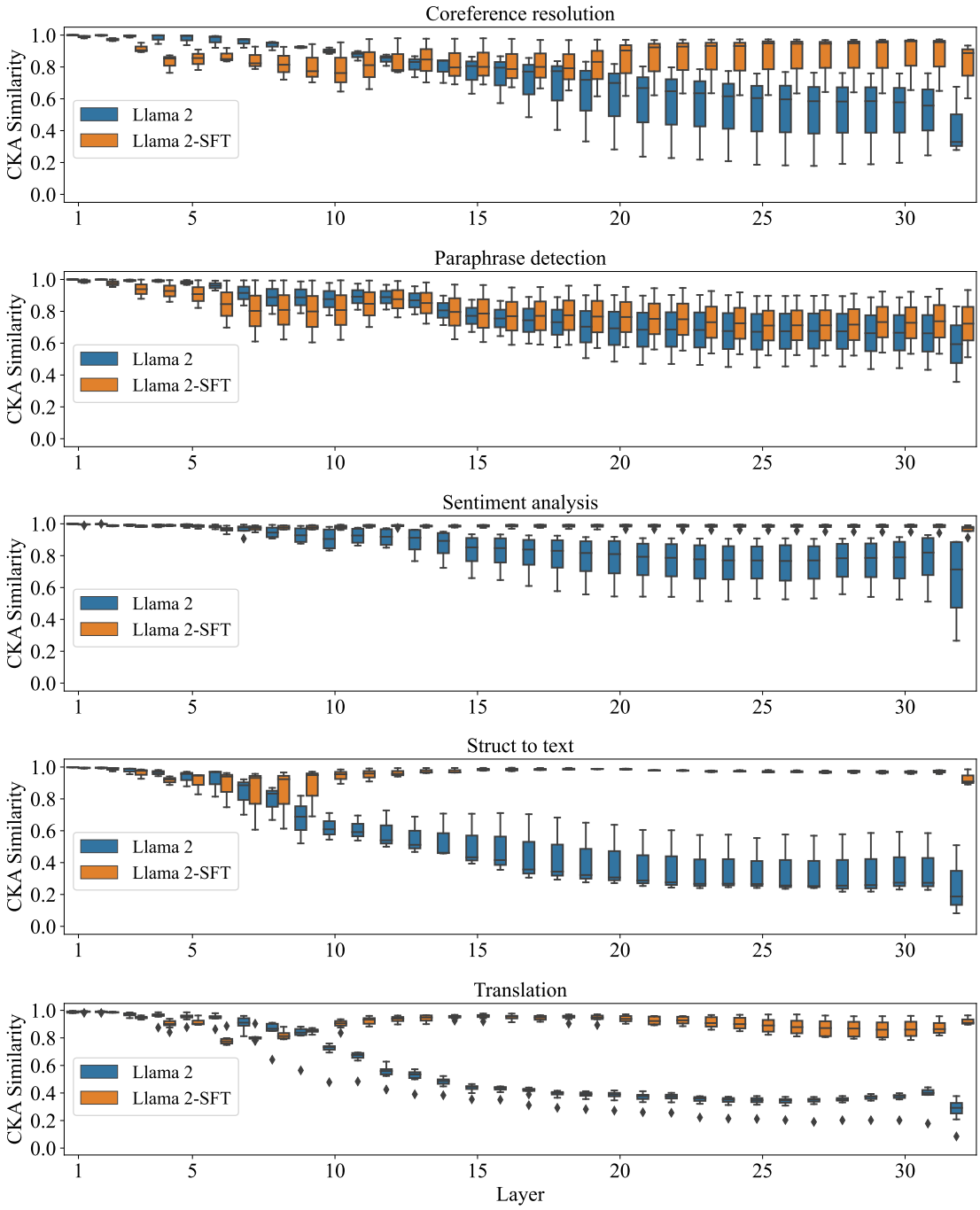


Figure 6.2: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model, grouped by different task clusters.

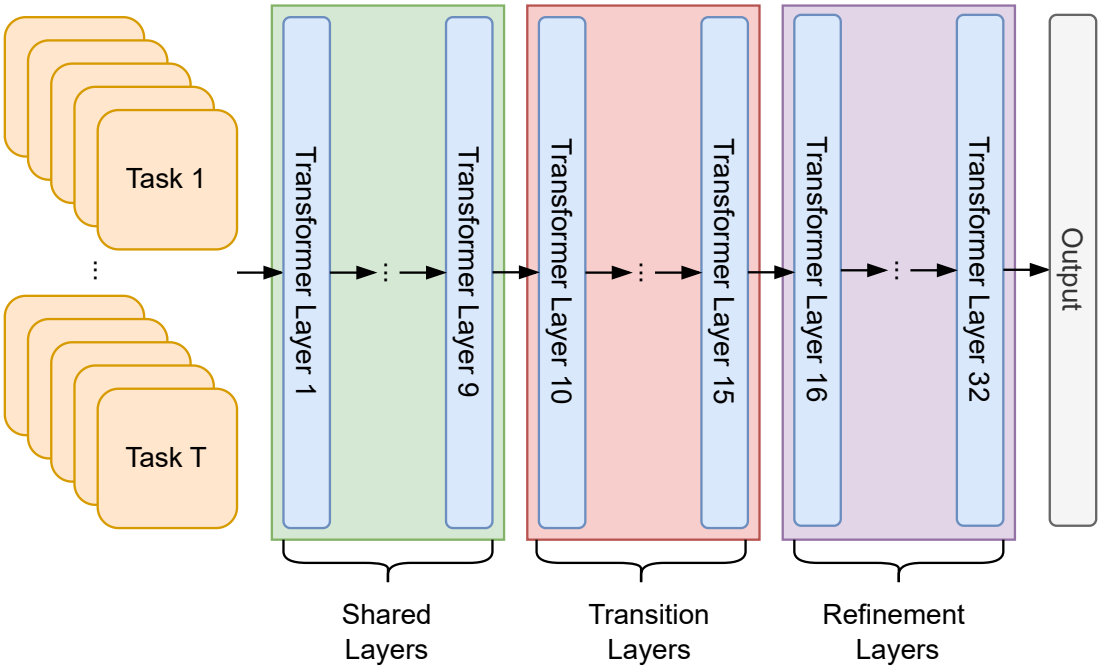


Figure 6.3: An illustration of our findings using the Llama 2 7B model (Touvron et al., 2023) as an example. We show that when instruction tuning on T different tasks, the layers are divided into three functional sections: the shared layers (layers 1 to 9) form general representations shared among all tasks, the transition layers (layers 10 to 15) transition the representations into task-specific information, and the refinement layers (layers 16 to 32) continue to refine the representations toward specific tasks.

model (Llama 2-SFT) with task-specific fine-tuned control models. As illustrated in Figure 6.1, the CKA similarities between Llama 2-SFT and the control models do not decrease as significantly as those for the pre-trained model (Llama 2) across layers. In the early layers (1 to 9), we observe that for many tasks, the CKA scores are lower for Llama 2-SFT compared to Llama 2, indicating that Llama 2-SFT representations diverge from those of the control models, which were fine-tuned on individual tasks (thus specializing in them). This suggests that, unlike the Llama 2 model, training Llama 2-SFT on a high number of tasks encourages it to diverge from the control models' representations and learn more general representations in the lower layers, a characteristic typical of MTL models. We denote layers 1-9 as "shared layers", as our findings suggest their representations are shared across tasks, similar to more studied MTL models.

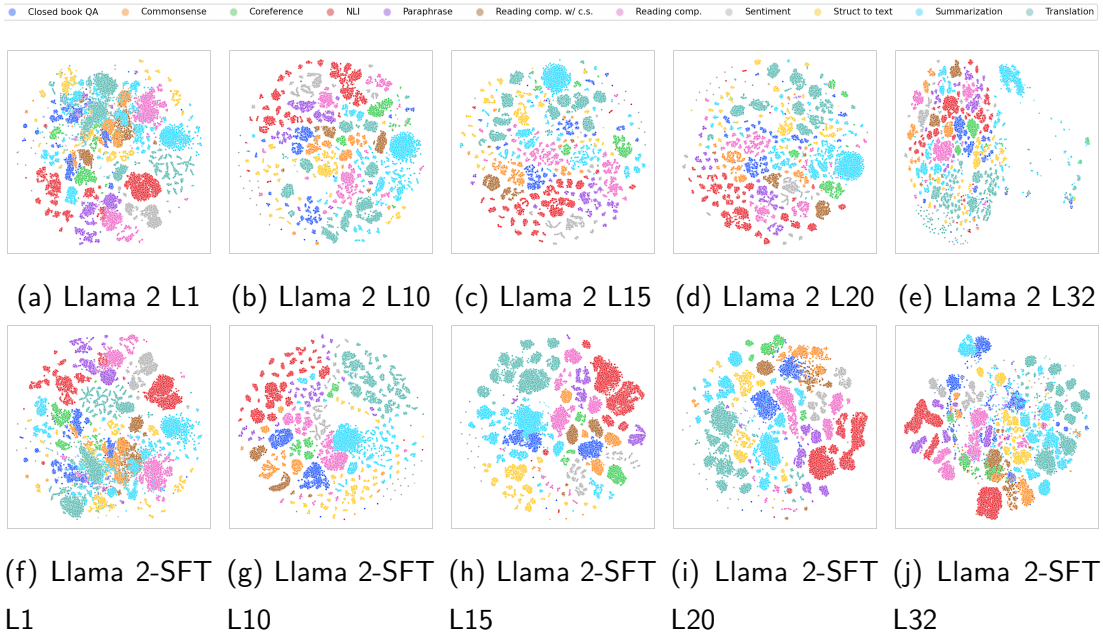
In the middle layers (10-15), there is a significant transition, with the Llama 2-SFT model exhibiting high similarity to *all control models*. This indicates that

these layers encode a high degree of task-specific information, as their representations are almost identical to those of the specialized control models. We denote layers 10-15 as "transitional layers", as our findings suggest the transition to task-specific representations occurs within these layers. This trend continues, albeit to a lesser extent, up to the final layers (16-32), which we denote as "refinement layers", as they keep refining the representations up to the final prediction. Based on our findings, we can map each layer in the Llama 2-SFT model to its corresponding function with respect to MTL (see Figure 6.3). While previous work (Wei et al., 2022a; Chung et al., 2022) has empirically demonstrated the effectiveness of instruction tuning for improving performance on a variety of NLP tasks, to the best of our knowledge, we are the first to propose such a mapping. In the following sections, we provide additional analyses to further validate our mapping.

Examining individual task clusters Figure 6.2 demonstrates that for tasks that are not well encoded in the pre-trained Llama 2 (e.g., structured data to text generation, translation), the CKA similarities from the instruction-tuned Llama 2-SFT remained high throughout all transition and refinement layers (10-32). Instruction tuning for these tasks induced significant representational shifts, adapting the model's internal structure to meet their specific demands. This aligns with prior work (Aghajanyan et al., 2021) showing that tasks requiring more sophisticated reasoning and modeling benefit greatly from task-specific tuning of pre-trained language models.

6.4.3 Representation Clustering and Variance Analysis

To further analyze representational differences, we used t-SNE (Van der Maaten and Hinton, 2008) to visualize task clusters across layers. Figure 6.4 presents a representative selection of layers, including a shared layer (layer 1), transition layers (layers 10 and 15), and refinement layers (layers 20 and 32). The full results for all layers are provided in Appendix C.2.2. In the first layer, both Llama 2 and Llama 2-SFT exhibit similar clustering. However, as we move to the transition layers, from layers 10 to 15, the Llama 2-SFT model forms more distinct task clusters compared to the Llama 2 model. This is further evidence that instruction tuning transforms the representations towards task-specificity in the transition layers. This clustering becomes increasingly pronounced in refinement layers,



(a) Llama 2 L1 (b) Llama 2 L10 (c) Llama 2 L15 (d) Llama 2 L20 (e) Llama 2 L32

(f) Llama 2-SFT L1 (g) Llama 2-SFT L10 (h) Llama 2-SFT L15 (i) Llama 2-SFT L20 (j) Llama 2-SFT L32

Figure 6.4: t-SNE visualizations of the representations for each task cluster in different layers of the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model. Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer of the respective model. "Reading comp." denotes reading comprehension tasks, and "reading comp. w/ c.s." denotes reading comprehension tasks with commonsense reasoning.

highlighting the effectiveness of instruction tuning in differentiating task-specific information and enhancing the ability to specialize representations for different tasks.

To quantify these differences, we performed variance analysis on the representations. We sought to determine if the model's ability to retain a large amount of task-specific information for many tasks affects its representation complexity. We analyzed the number of principal components required to explain 99% of the variance in representation matrices across layers. The average number of components over all tasks is presented in Figure 6.5. In the shared layers, both Llama 2 and Llama 2-SFT models require a similar number of dimensions. Then, in the transition layers, Llama 2-SFT model begins to require more dimensions, suggesting it captures more complex task-specific information. This further demonstrates that the transition layers are indeed the layers where the transition to the task-specific representations occurs.

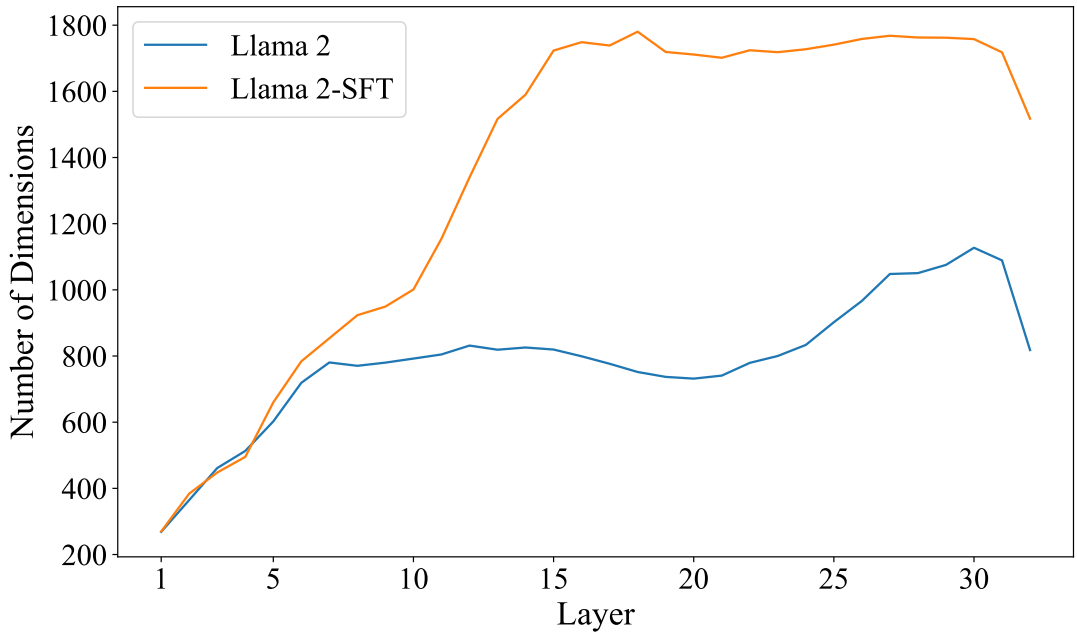
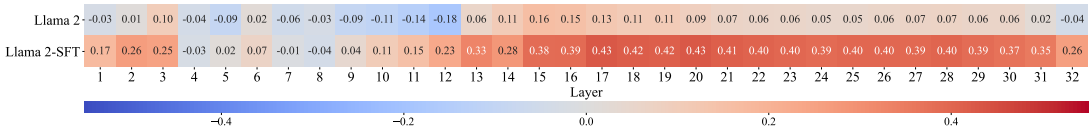


Figure 6.5: Average number of dimensions required to explain 99% of the representational variance across all tasks, as a function of the layer number.

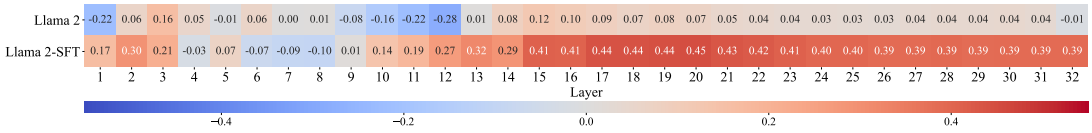
6.4.4 Assessing Task Specific Information via Readability

In the preceding sections, we observed that the Llama 2 model exhibited a high variance in the amount of task-specific information stored across different tasks. In contrast, the Llama 2-SFT model demonstrated a low variance, storing a high level of task-specific information in its transition and refinement layers. While the Llama 2-SFT model exhibited low variance, we aimed to investigate the task priorities within the representation and identify features that could predict it. Previous research by Zhao et al. (2022) has shown that when masked language models, such as BERT (Devlin et al., 2019), are trained on data from multiple domains, they tend to allocate their parameters to store domain-specific information. We followed a similar analysis to examine task-specific information, which is strongly related to domain-specific information (as tasks can be viewed as domains). We used readability as a proxy for domain-specific information, relying on the finding by Pitler and Nenkova (2008) that texts with more domain-specific and less commonly used words tend to have lower readability, resulting in higher reading difficulty scores.

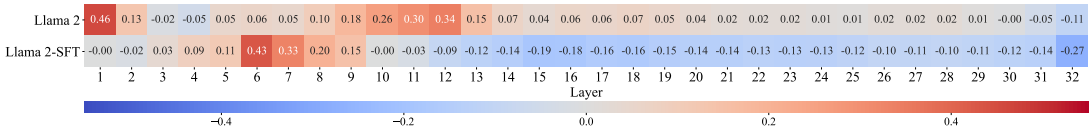
We used two highly popular reading difficulty measures: the Flesch-Kincaid grade level score (Kincaid et al., 1975) and the Coleman-Liau Index (Coleman



(a) Flesch–Kincaid grade level



(b) Coleman-Liau index



(c) Data size

Figure 6.6: Pearson correlation results between the CKA similarities for all tasks, their reading difficulty, and data size across all layers. Higher values in reading difficulty measures correspond to greater reading difficulty.

and Liau, 1975). The Flesch-Kincaid score assesses text readability based on factors like average sentence length and syllables per word, with lower scores indicating easier reading. Similarly, the Coleman-Liau Index estimates the required reading grade level based on characters, words, and sentences, with higher values corresponding to greater difficulty. We performed Pearson correlation analyses between CKA similarity and reading difficulty measures for all tasks across all layers. Specifically, we first calculated the readability measure for each input instruction, then obtained CKA similarities for representations from each layer. Finally, we computed the Pearson correlation coefficients between each input's readability measure and the corresponding CKA similarities from each layer.

As illustrated in Figure 6.6a, we found a positive correlation between CKA similarity and the Flesch-Kincaid score for Llama 2-SFT. This correlation rapidly increases between layer 10 and layer 15 (the transition layers) and then saturates. These transitional layers are where task specialization transformations occur, as discussed earlier. This correlation is much weaker for the Llama 2 model. A similar pattern is observed with the Coleman-Liau Index, as shown in Figure 6.6b. These findings suggest that instruction-tuned models encode more information for

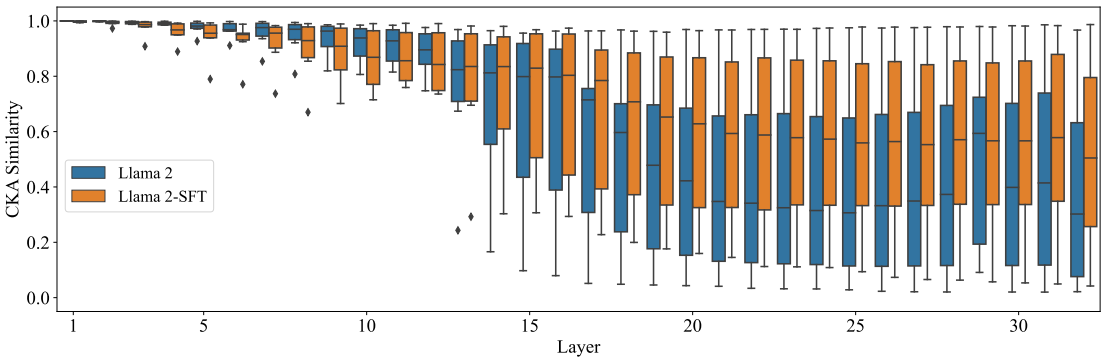


Figure 6.7: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model on unseen tasks.

tasks with more task-specific vocabulary, as measured by their texts' readability indices. These findings thus suggest that instruction-tuned models encode and preserve task-specific information in the transition layers and retain it through the refinement layers, complementing our earlier findings. Moreover, we previously noted that one of the advantages of CKA, compared to other similarity metrics, is its minimal requirement for a large number of data points in the analysis. To verify this, we conducted a correlation analysis between data size and CKA similarity, with the results presented in Figure 6.6c. The analysis revealed no clear correlation between data size and CKA similarities, indicating that the number of data points used for CKA per task does not impact the CKA similarity.

6.4.5 Evaluating Representations on Unseen Tasks

While our previous analyses focused on evaluating representations against models trained on the same task data, it is crucial to examine how well our findings generalize to unseen tasks. To investigate this, we held out a set of seven tasks, including conversational question answering, question classification, math problems, linguistic acceptability, and word sense disambiguation (details in Appendix C.1). Our instruction-tuned models had no exposure to any of these seven tasks during training.

The CKA similarity results in Figure 6.7 reveal an interesting pattern. For the lower layers (up to layer 12), the Llama 2 model exhibited slightly higher CKA similarities than Llama 2-SFT for several tasks, similar to what we find in § 6.4.2. This indicates that while the Llama 2-SFT model was not trained using these tasks, it produced more divergent representations in lower layers and thus

more general than the ones produced by Llama 2 (we refer the reader to the shared layers discussion in § 6.4.2 for more details). However, as we move to the middle and higher layers responsible for encoding more specialized, task-specific knowledge, the Llama 2-SFT model began matching and ultimately surpassing the CKA similarities of the Llama 2 model. We can also see high variances between task similarities for both models, showing that we can not identify transition layers for Llama 2-SFT in this setup, just shared and refinement layers. These findings suggest that in addition to being trained on instructions, instruction-tuned models benefit from more general and thus better feature representations in their lower layers, which boost their performance for unseen instruction-based tasks compared to pre-trained LLMs.

6.5 Discussion

Our study offers comprehensive insights into the impact of instruction tuning on the representations learned by LLMs. Previous work has discussed the benefits of instruction tuning (Wei et al., 2022b; Chung et al., 2022; Longpre et al., 2023), but ours is the first to analyze their effects from a representational perspective.

Our analysis revealed that LLMs instruction-tuned on multiple tasks learned different representations in the lower layers compared to LLMs tuned on individual tasks. Similar to MTL, such representations can be shared and used across tasks (Maurer et al., 2016). Our analysis uncovered a key novel finding – we observed clear differences between pre-trained and instruction-tuned models, with the most significant representational transformations occurring in the middle transitional layers. This finding highlights the critical role of middle layers in encoding the specialized task knowledge induced by instruction tuning. Similarly, previous studies in multilingual settings have also identified language-neutral transformations in the middle layers of the network (Muller et al., 2021; Zhao et al., 2023). Furthermore, our analysis suggests that in the refinement layers, instruction-tuned models continue to shape representations toward specific tasks but without substantial representational changes with respect to task-specific information. Overall, our finding about functionality for different layers in LLMs generally aligns with previous findings on BERT, which have shown that lower layers are more general, while upper layers are known to be more task-specific (Rogers et al., 2020; Merchant et al., 2020).

Our correlation analysis also revealed insights into the relationship between representations and task complexity. Instruction-tuned models exhibited a positive correlation with reading complexity measures in the transition and refinement layers, suggesting better encoding of task-specific information for tasks with more specific vocabulary – a capability not observed in pre-trained models. Notably, instruction tuning enabled models to preserve and enhance task-specific information across a broader range of layers, as evidenced by higher CKA similarities compared to control models. Our evaluation of unseen tasks further underscored the benefits of instruction tuning for improving generalization, with instruction-tuned models outperforming their pre-trained counterparts in deeper layers responsible for encoding complex task knowledge. This aligns with empirical evidence from Wei et al. (2022a) but also highlights how representational changes facilitated by instruction tuning strengthen cross-task transfer capabilities.

6.6 Conclusion and Transition

This chapter applied our MOSSA framework to investigate how instruction tuning shapes representations in LLMs across more than 60 tasks. Our analyses revealed that unlike its pre-trained counterpart, the instruction-tuned model retained a high amount of task-specific information for all tasks from the middle layers onward. Crucially, we mapped the layers of the instruction-tuned LLM into three distinct functional groups: initial *shared layers* (1-9) for general representations, middle *transition layers* (10-15) where rapid task specialization occurs, and final *refinement layers* (16-32) that hone these representations. This discovery of a consistent, functional segmentation of the model architecture represents the key finding of this final study.

This work completes the empirical arc of the thesis. We have demonstrated that the core principle of subpopulation specialization, first observed with domains in Chapter 4 and scaled to languages in Chapter 5, holds true even in the massively multi-task setting of modern LLMs. The MOSSA framework, adapted with progressively more sophisticated similarity metrics (SVCCA, PARAFAC2, CKA), has proven to be a robust and insightful tool for decomposing the internal representations of generalist models. We have consistently found that these models learn to partition their representational space to handle data heterogeneity. The final chapter will now synthesize the findings from all three studies, discuss

their broader implications for the field of NLP, and outline promising avenues for future research.

Chapter 7

Conclusion

This thesis began with a fundamental question that strikes at the heart of modern NLP: how do generalist language models, trained on vast and heterogeneous mixtures of data, learn to effectively handle the distinct demands of different domains, languages, and tasks? The remarkable success of these models has, in many ways, outpaced our scientific understanding of their internal mechanisms. The goal of this dissertation was to address this gap by developing and systematically applying a unified analytical framework capable of decomposing the internal representations of these complex systems. We proposed a methodology of **Model-Oriented Sub-population and Spectral Analysis (MOSSA)**, which moves beyond studying models in isolation and instead derives insights from the principled comparison of a generalist model with a series of specialist control models. Across three distinct and increasingly complex investigations, this framework has yielded a consistent and unifying answer to our initial question.

7.1 Synthesis of Findings

The empirical work of this thesis progressed through three studies, each building upon the last to test the principles of representational specialization in a new context.

In **Chapter 4**, we applied the methodology underlying our MOSSA framework to investigate domain learning. Using SVCCA, we compared a multi-domain model with single-domain specialists. This initial study provided the first piece of crucial evidence: model capacity is a key enabler of specialization. We demonstrated that larger models do not simply learn a more diffuse, general repres-

entation; instead, they use their additional parameters to embed specialist-like representations for domain-specific vocabulary, particularly in their lower layers. This finding validated our comparative approach and established the core hypothesis that generalist models learn to contain the behaviors of specialists.

In **Chapter 5**, we scaled this investigation to the more complex challenge of multilingualism, analyzing a model trained on over 30 languages. The pairwise nature of SVCCA was insufficient for this task, prompting the development of a novel analysis tool based on joint matrix factorization (PARAFAC2). This methodological innovation allowed for the simultaneous comparison of all languages against the shared multilingual model. The results were striking: we found that the encoding of morphosyntactic information varied systematically across the model's layers and was influenced by linguistic properties such as writing systems. The representational geometries uncovered by our analysis were not arbitrary; they correlated strongly with cross-lingual task performance and could even reconstruct plausible phylogenetic trees. This study demonstrated that our framework was not only scalable but could also reveal highly structured, linguistically meaningful organization within the model's representations.

Finally, in **Chapter 6**, we brought our framework to bear on the contemporary paradigm of massively multi-task instruction tuning in LLMs. Using CKA to compare a multi-task LLM against over 60 single-task specialist models, we uncovered the clearest evidence of functional specialization to date. Our analysis revealed a consistent, three-stage segmentation of the model's architecture. We identified:

1. **Shared Layers** (in the Llama 2-7B model, layers 1-9), which produce general, task-agnostic representations.

2. **Transition Layers** (layers 10-15), where a rapid specialization occurs and representations are transformed to become highly similar to their single-task counterparts.

3. **Refinement Layers** (layers 16-32), which fine-tune these task-specific representations for final prediction.

This discovery of a layered, functional architecture for multi-task processing provided a capstone for our investigation, demonstrating that the principle of specialization holds even in the most complex generalist models.

7.2 The Core Contribution: A Unified View of Specialization

Viewed collectively, these three studies make a single, cohesive argument: **generalist language models learn to partition their representational space to create specialized, quasi-independent subspaces for different data regimes.** Whether the subpopulation is a text domain, a human language, or an NLP task, a sufficiently large model does not learn a single, averaged representation. Instead, it learns to emulate the behavior of a specialist. This thesis provides a robust and scalable methodology for pinpointing where this specialization occurs within the model's architecture and elucidating the principles that govern it. The consistent pattern of divergence in lower layers for general representations and convergence in upper layers for specific ones reveals a fundamental organizational principle of these powerful neural networks.

7.3 Broader Implications

The findings of this dissertation have several broader implications for the field of Natural Language Processing and beyond.

- **For Interpretability Research:** This work presents a powerful alternative to probing-based analyses. The subpopulation framework is a versatile and scalable methodology that can be adapted with different similarity metrics (SVCCA, PARAFAC2, CKA) depending on the analytical need. It moves the field from asking "what is in this representation?" to "how does this representation differ from a specialist?", a question that is often more revealing about the nature of generalization and adaptation.

- **For Model Development:** Our findings have direct practical relevance. The identification of "transition layers" where task-specific knowledge is encoded suggests that parameter-efficient fine-tuning techniques, such as LoRA, could potentially be made more effective by targeting these specific layers. Understanding which layers are shared and which are specialized could inform more sophisticated methods for model pruning, compression, and modular architecture design. For multilingual models, insights into how

linguistic properties affect representations could guide better data sampling strategies during pre-training.

- **For Artificial Intelligence:** At a more abstract level, this thesis offers a perspective on how complex computational systems can achieve generality. The emergence of specialized subspaces within a unified architecture mirrors concepts of functional specialization in the brain. It suggests that a path to artificial general intelligence may not lie in discovering a single, universal learning algorithm, but rather in creating systems that have the capacity and mechanisms to internally modularize and specialize their processing for the diverse array of problems they encounter.

7.4 Future Directions

The framework and findings presented in this thesis lay the groundwork for a deeper understanding of generalist models and open up several exciting avenues for future research. The insights gained from our analyses naturally lead to a new set of questions that can be explored through more targeted and causal investigations.

- **Causal Interventions:** The layer mappings discovered in Chapter 6, which identified shared, transition, and refinement layers, provide a clear roadmap for causal experiments. Future work could investigate whether freezing the shared layers and fine-tuning only the transition and refinement layers can achieve comparable performance with significantly fewer updated parameters. Another compelling experiment would be to swap the transition-layer representations for one task with those of another during inference to directly test their role in task execution.

- **Scaling the Analysis:** While the models studied in this thesis are significant, the field of LLMs is rapidly advancing. Applying the MOSSA framework to state-of-the-art, frontier-scale models with hundreds of billions of parameters is a critical next step. Such a study would validate whether the principles of layered specialization and functional segmentation are a fundamental property of Transformer-based learning or if new organizational principles emerge at an even larger scale.

- **Dissecting Sub-Layer Mechanisms:** This work has focused on the hidden state representations at the output of each Transformer layer. A more fine-grained analysis could apply the MOSSA framework to dissect the specific roles of the sub-layer components. By comparing the outputs of the self-attention blocks versus the feed-forward networks, future research could determine where within a layer the specialization for different subpopulations is most pronounced, shedding light on the distinct computational contributions of attention and MLP modules.

- **Extending to Multimodality:** The MOSSA framework is modality-agnostic. A compelling future direction is to apply it to vision-language models. By comparing a generalist multi-modal model to vision-only and language-only specialist models, one could use the comparative analysis to pinpoint the precise layers where cross-modal fusion and conceptual alignment occur, providing a clearer picture of how these models learn to connect text and images.

- **Dynamic Specialization:** This thesis focused on a static, post-training analysis of model representations. An exciting frontier is to investigate how specialization might occur dynamically during inference. Do models learn to route inputs through different computational pathways based on the task or language identified in a prompt? Techniques inspired by mixture-of-experts could be used to explore whether models effectively activate different "virtual sub-networks" on the fly, leading to a more dynamic and efficient form of specialization.

In closing, the models that now define the field of artificial intelligence are among the most complex artifacts ever created by humankind. As their capabilities continue to grow, so too does the imperative to understand them. By decomposing their internal representations, this thesis has sought to replace opacity with structure, revealing a consistent and elegant principle of specialization that governs how these powerful models learn to master the diversity of human language. It is our hope that this framework and its findings will serve as a valuable step toward building more efficient, robust, and ultimately more understandable intelligent systems.

Appendix A

Appendix for Chapter 4

This appendix provides additional figures, tables, and analyses that support the findings presented in Chapter 4.

A.1 Supplemental Figures and Analyses

A.1.1 Additional Results for RQ1: Training Dynamics

This section provides the complete set of training dynamics plots for all domains, complementing the primary example shown in the main chapter. Figures A.1 through A.4 show the layer-wise SVCCA similarity between the experimental model (**E**) and the control models for the remaining four domains (Clothing Shoes and Jewelry, Electronics, Home and Kitchen, and Movies and TV). The observed trends are consistent across all domains, showing a clear divergence in the upper layers as training progresses.

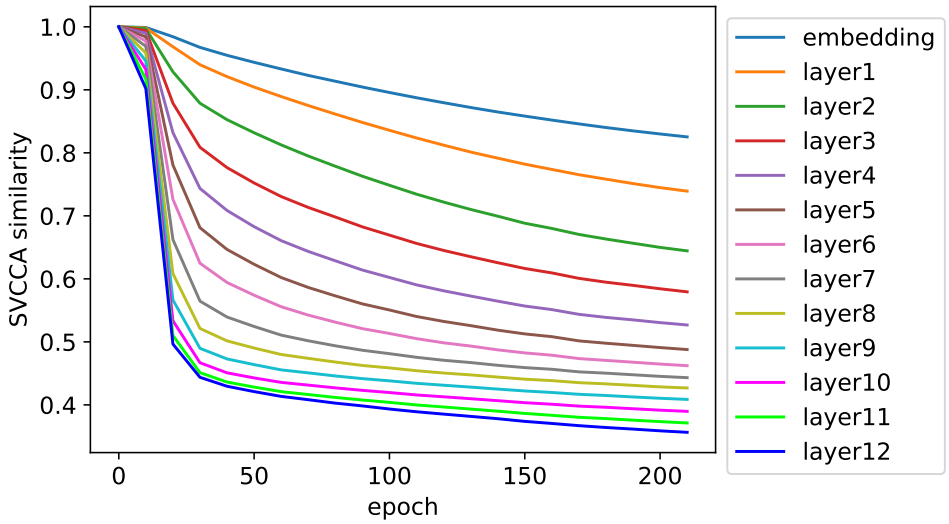


Figure A.1: Training dynamics for all layers between **E** and $\mathbf{C}_{Clothing}$. Here both model and data size are 100%.

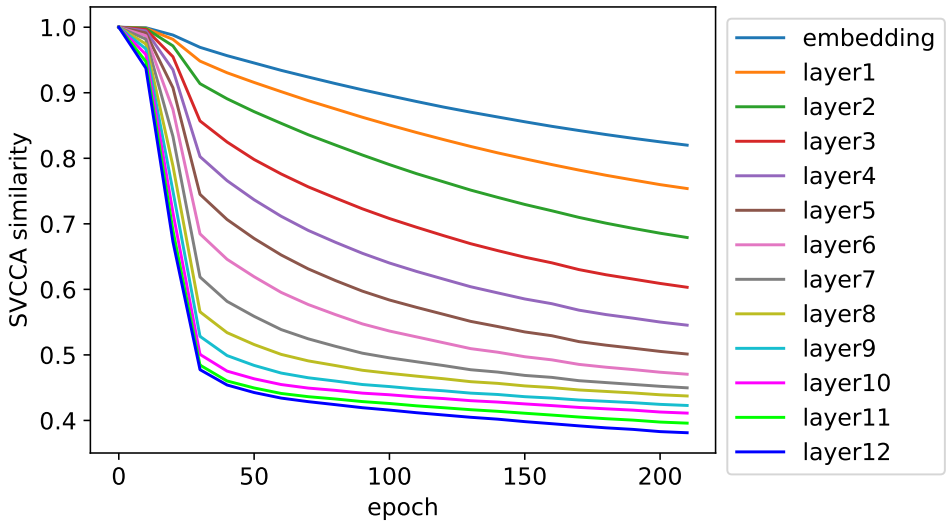


Figure A.2: Training dynamics for all layers between **E** and $\mathbf{C}_{Electronics}$. Here both model and data size are 100%.

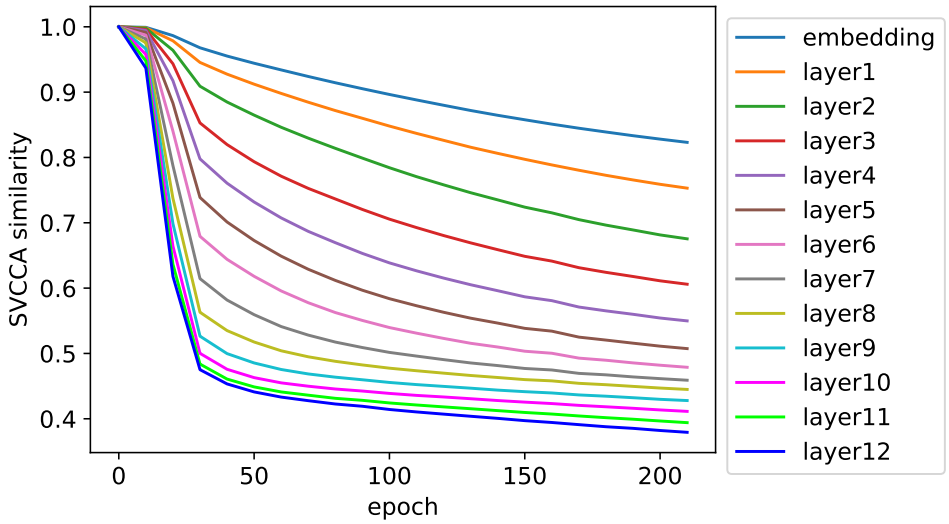


Figure A.3: Training dynamics for all layers between **E** and $\mathbf{C}_{Home}$. Here both model and data size are 100%.

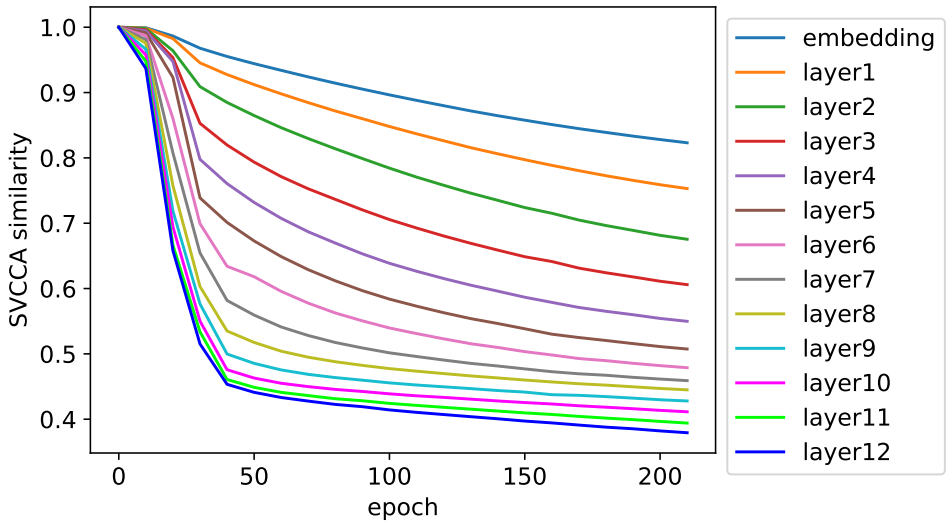
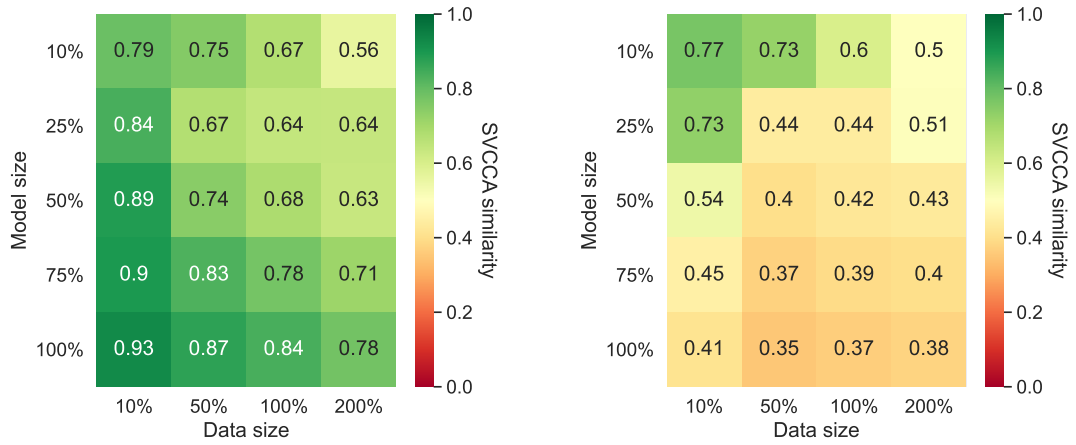


Figure A.4: Training dynamics for all layers between **E** and $\mathbf{C}_{Movies}$. Here both model and data size are 100%.

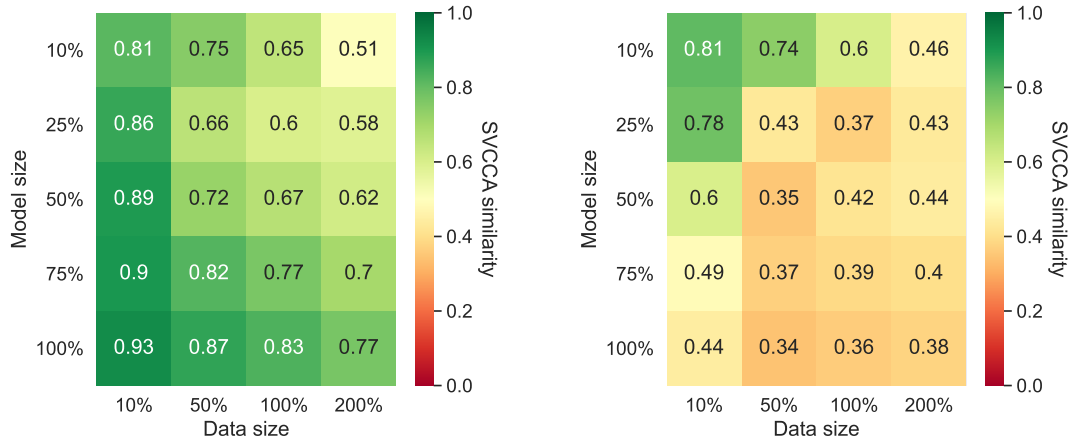
A.1.2 Additional Results for RQ2: Impact of Data Size and Model Capacity

In § 4.4, we provided SVCCA results for three domains. Here we present the results for the remaining two domains, Home and Kitchen and Movies and TV, in Figures A.5 and A.6.



(a) ℓ_0: Home and Kitchen (b) ℓ_{12}: Home and Kitchen

Figure A.5: SVCCA scores between **E** and $\mathbf{C}_{Home}$ for all words.

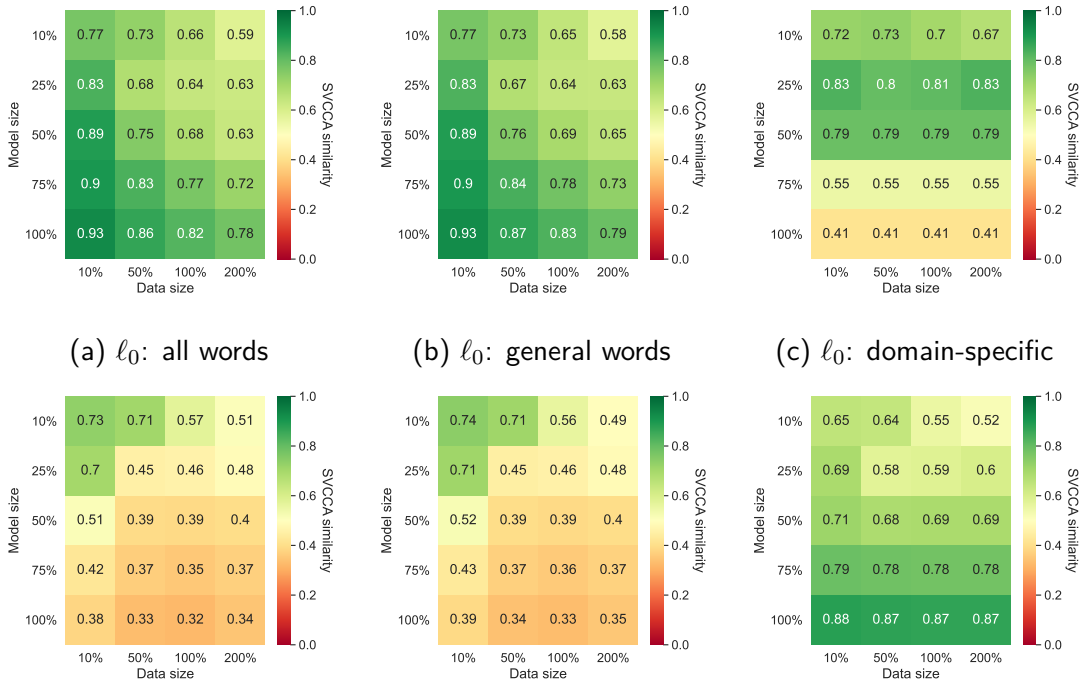


(a) ℓ_0: Movies and TV (b) ℓ_{12}: Movies and TV

Figure A.6: SVCCA scores between **E** and $\mathbf{C}_{Movies}$ for all words.

A.1.3 Additional Results for RQ3: Analysis by Word Type

This section contains the complete results for the SVCCA analysis broken down by word type (all, general, and domain-specific). Figure A.7 illustrates the results for $\mathbf{C}_{Clothing}$, Figure A.8 for $\mathbf{C}_{Electronics}$, Figure A.9 for $\mathbf{C}_{Home}$, and Figure A.10 for $\mathbf{C}_{Movies}$.

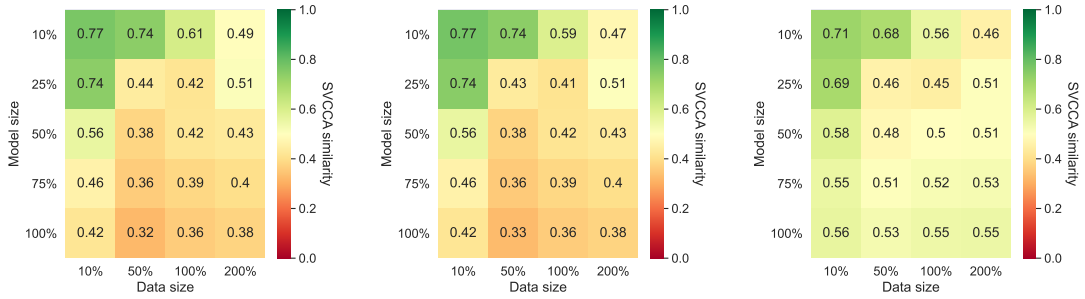


(a) ℓ_0: all words (b) ℓ_0: general words (c) ℓ_0: domain-specific

(d) ℓ_{12}: all words (e) ℓ_{12}: general words (f) ℓ_{12}: domain-specific

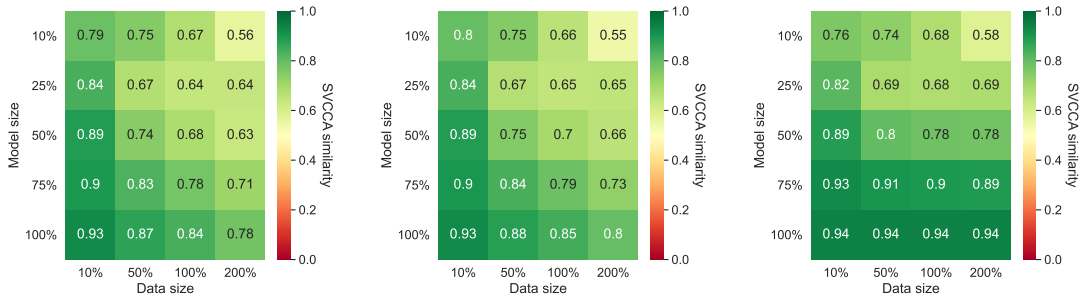
Figure A.7: The SVCCA score between $\mathbf{E}$ and $\mathbf{C}_{Clothing}$ for different subsets of tokens. The top row presents the results for the embedding layer ℓ_0, and the bottom row presents them for the last layer ℓ_{12}.

(a) ℓ_0: all words (b) ℓ_0: general words (c) ℓ_0: domain-specific

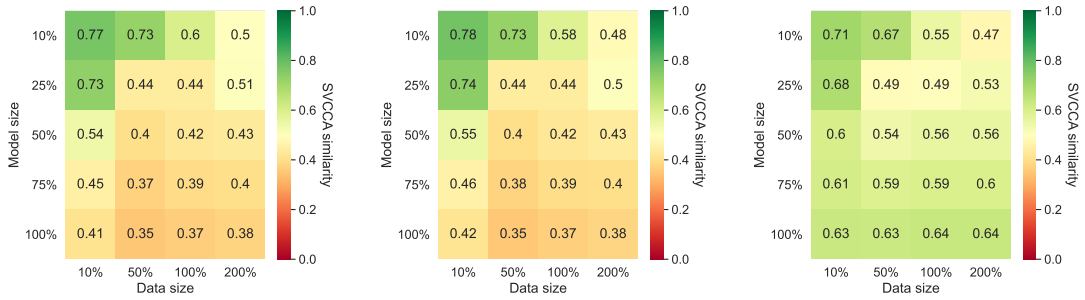


(d) ℓ_{12}: all words (e) ℓ_{12}: general words (f) ℓ_{12}: domain-specific

Figure A.8: The SVCCA score between $\mathbf{E}$ and $\mathbf{C}_{Electronics}$ for different subsets of tokens.

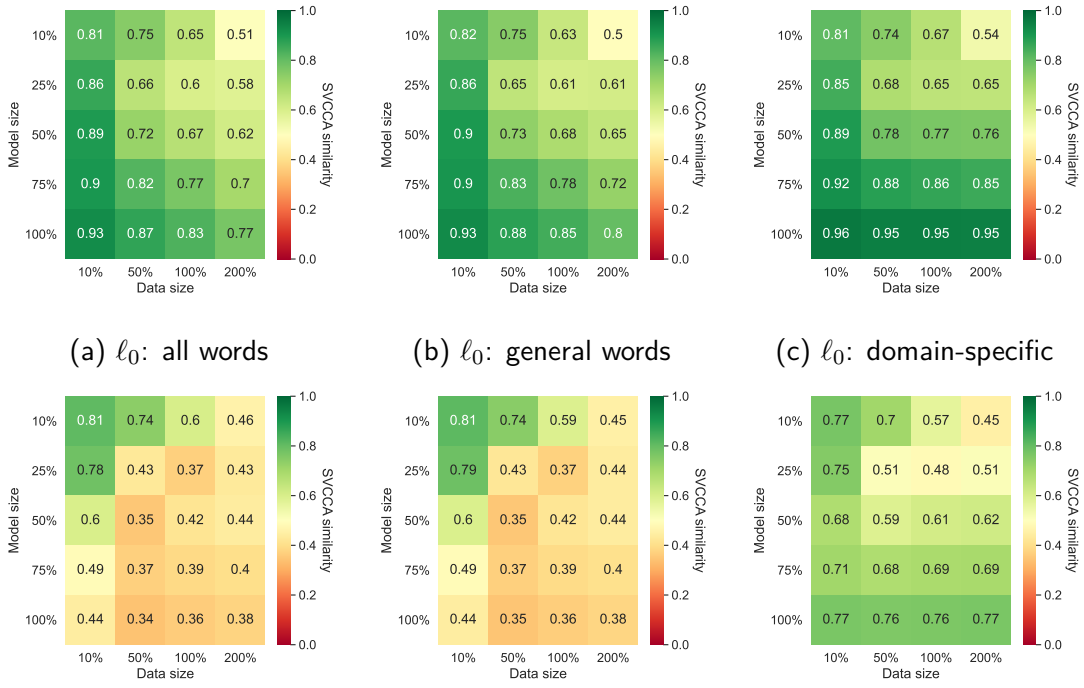


(a) ℓ_0: all words (b) ℓ_0: general words (c) ℓ_0: domain-specific



(d) ℓ_{12}: all words (e) ℓ_{12}: general words (f) ℓ_{12}: domain-specific

Figure A.9: The SVCCA score between $\mathbf{E}$ and $\mathbf{C}_{Home}$ for different subsets of tokens.



(a) ℓ_0: all words (b) ℓ_0: general words (c) ℓ_0: domain-specific

(d) ℓ_{12}: all words (e) ℓ_{12}: general words (f) ℓ_{12}: domain-specific

Figure A.10: The SVCCA score between **E** and $\mathbf{C}_{Movies}$ for different subsets of tokens.

A.1.4 Additional Results for RQ4: Qualitative Analysis

Here we provide more example MLM predictions of $\mathbf{E}$ and $\mathbf{C}_i$. Table A.1 presents predictions using k-nearest neighbors of the word embeddings. Table A.2 presents predictions using the final layer representation.

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
editors	volumns	editors	editors
publisher	buyer	publisher	publisher
heirs	listing	editor	editor
libraries	edit	writers	authors
universities	hardcover	authors	reviewers

(a) 5-nearest neighbors for the word **publishers** (i=Books).

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
towards	towards	towards	towards
beside	settled	against	at
surrounding	at	onto	onto
beneath	concerning	at	against
against	behind	beside	near

(b) 5-nearest neighbors for the word **toward** (i=Books).

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
comics	jokes	comics	comics
jokes	joke	comedian	joke
comedian	accolades	laughs	comedian
directors	critics	comedies	critics
commentators	reviewers	jokes	laughs

(c) 5-nearest neighbors for the word **comedians** (i=Movies and TV).

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
print	vinyl	plastic	plastic
plastic	bonded	print	vinyl
cloth	plastic	materials	cardboard
cardboard	junk	paperback	print
printed	cardboard	cardboard	tissue

(d) 5-nearest neighbors for the word **paper** (i=Clothing).

Table A.1: Example predictions of $\mathbf{E}$ and $\mathbf{C}_i$ using 5-nearest neighbors from embedding layer weights. 'm' denotes model capacity. All models here use data size of 100%.

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
food	counter	bottle	counter
counter	hands	refrigerator	bottle
wine	oil	wine	hands
oil	food	food	sink
salad	salad	fridge	stove

(a) *...sprayed on my [MASK] but...* (i=Home). Masked word: **salad**.

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
guy	guy	girl	guy
musician	woman	guy	woman
dude	man	killer	hero
kid	kid	gal	cop
vampire	person	dude	man

(b) *...smoothing-talking [MASK] who gets...* (i=Movies). Masked word: **joker**.

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
say	have	worry	worry
think	say	complain	say
complain	know	wonder	know
know	care	know	think
worry	understand	say	complain

(c) *...have to [MASK] about these drives.* (i=Electronics). Masked word: **worry**.

m=50%		m=100%	
E	$\mathbf{C}_i$	**E**	$\mathbf{C}_i$
instructed	expected	suggested	suggested
suggested	instructed	stated	instructed
well	stated	instructed	expected
usual	advertised	advertised	well
indicated	normal	well	stated

(d) *...half size down as [MASK] and...* (i=Clothing). Masked word: **suggested**.

Table A.2: Example MLM predictions of **E** and $\mathbf{C}_i$ using last layer representation. 'm' denotes model capacity. All models here use a data size of 100%.

Appendix B

Appendix for Chapter 5

B.1 Information on Attributes and Languages

We first provide information about all languages we use in our experiment in Table B.1. The information includes ISO 639-1 codes for all languages, the language family and the genus they belong to. In Table B.2, we present all morpho-syntactic attributes we experiment. For each attribute, we list all languages that have the attribute. We also provide a reverse list where we list by languages:

- **Arabic (ar)**: Aspect, Case, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Politeness, Voice
- **Bulgarian (bg)**: Aspect, Case, Comparison, Definiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Tense, Valency, Voice
- **Catalan (ca)**: Aspect, Case, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense
- **Chinese (zh)**: Aspect, Case, Number, Part of Speech, Person, Polarity, Valency, Voice
- **Croatian (hr)**: Animacy, Case, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency, Voice
- **Czech (cs)**: Animacy, Aspect, Case, Comparison, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency, Voice
- **Danish (da)**: Case, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Possession, Tense, Valency, Voice

- **Dutch (nl)**: Case, Comparison, Definiteness, Finiteness, Gender, Number, Part of Speech, Person, Tense, Valency
- **English (en)**: Case, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Tense, Valency
- **Estonian (et)**: Aspect, Case, Comparison, Finiteness, Mood, Number, Part of Speech, Person, Polarity, Tense, Valency, Voice
- **Finnish (fi)**: Case, Comparison, Finiteness, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency, Voice
- **French (fr)**: Aspect, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Tense, Valency
- **Galician (gl)**: Part of Speech, Polarity
- **German (de)**: Case, Comparison, Definiteness, Finiteness, Mood, Number, Part of Speech, Person, Polarity, Politeness, Possession, Tense, Valency
- **Greek (el)**: Aspect, Case, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Tense, Voice
- **Hebrew (he)**: Case, Definiteness, Finiteness, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency, Voice
- **Hindi (hi)**: Aspect, Case, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Politeness, Tense, Voice
- **Hungarian (hu)**: Case, Comparison, Definiteness, Finiteness, Mood, Number, Part of Speech, Person, Possession, Tense, Valency
- **Indonesian (id)**: Part of Speech, Polarity
- **Italian (it)**: Aspect, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Tense
- **Japanese (ja)**: Part of Speech
- **Korean (ko)**: Part of Speech
- **Polish (pl)**: Animacy, Aspect, Case, Comparison, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency, Voice

- **Portuguese (pt)**: Aspect, Case, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Tense
- **Romanian (ro)**: Aspect, Case, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency
- **Russian (ru)**: Animacy, Aspect, Case, Comparison, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Tense, Valency, Voice
- **Slovak (sk)**: Animacy, Aspect, Case, Comparison, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency, Voice
- **Slovenian (sl)**: Animacy, Aspect, Case, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Possession, Tense, Valency
- **Spanish (es)**: Aspect, Case, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Tense, Valency
- **Swedish (sv)**: Case, Comparison, Definiteness, Finiteness, Gender, Mood, Number, Part of Speech, Polarity, Tense, Voice
- **Turkish (tr)**: Aspect, Case, Mood, Number, Part of Speech, Person, Polarity, Politeness, Possession, Tense, Valency, Voice
- **Ukrainian (uk)**: Animacy, Aspect, Case, Comparison, Finiteness, Gender, Mood, Number, Part of Speech, Person, Polarity, Tense, Valency, Voice
- **Vietnamese (vi)**: Part of Speech, Polarity

Notice that in this list and in our study, we omit a language that has less than 100 instances labeled for a particular morphosyntactic category.

B.2 Additional Results for RQ1

We present the Pearson correlations between the average sig(ℓ) for all languages and their number of unique characters for each morphosyntactic category among all layers in Figure B.1. The results for type-token ratio (TTR) is presented in Figure B.2. Please note that in the figures, the last row is labeled as "All Words", representing the results obtained from using representations taken from the dataset with Part-of-Speech (PoS) attributes. Given that each lexical token

Language	**ISO 639-1**	**Family**	**Genus**
Arabic	ar	Afro-Asiatic	Semitic
Bulgarian	bg	Indo-European	Slavic
Catalan	ca	Indo-European	Romance
Chinese	zh	Sino-Tibetan	Chinese
Croatian	hr	Indo-European	Slavic
Czech	cs	Indo-European	Slavic
Danish	da	Indo-European	Germanic
Dutch	nl	Indo-European	Germanic
English	en	Indo-European	Germanic
Estonian	et	Uralic	Finnic
Finnish	fi	Uralic	Finnic
French	fr	Indo-European	Romance
Galician	gl	Indo-European	Romance
German	de	Indo-European	Germanic
Greek	el	Indo-European	Greek
Hebrew	he	Afro-Asiatic	Semitic
Hindi	hi	Indo-European	Indic
Hungarian	hu	Uralic	Ugric
Indonesian	id	Austronesian	Malayo-Sumbawan
Italian	it	Indo-European	Romance
Japanese	ja	Japanese	Japanese
Korean	ko	Korean	Korean
Polish	pl	Indo-European	Slavic
Portuguese	pt	Indo-European	Romance
Romanian	ro	Indo-European	Romance
Russian	ru	Indo-European	Slavic
Slovak	sk	Indo-European	Slavic
Slovenian	sl	Indo-European	Slavic
Spanish	es	Indo-European	Romance
Swedish	sv	Indo-European	Germanic
Turkish	tr	Altaic	Turkic
Ukrainian	uk	Indo-European	Slavic
Vietnamese	vi	Austro-Asiatic	Vietic

Table B.1: All languages used in this study along with the language family and genus they belong to. Family and genus information is from WALS.

in the dataset is associated with a PoS tag, this analysis encompasses the entire dataset, enabling us to observe and comprehend the overall trends captured in the representations.

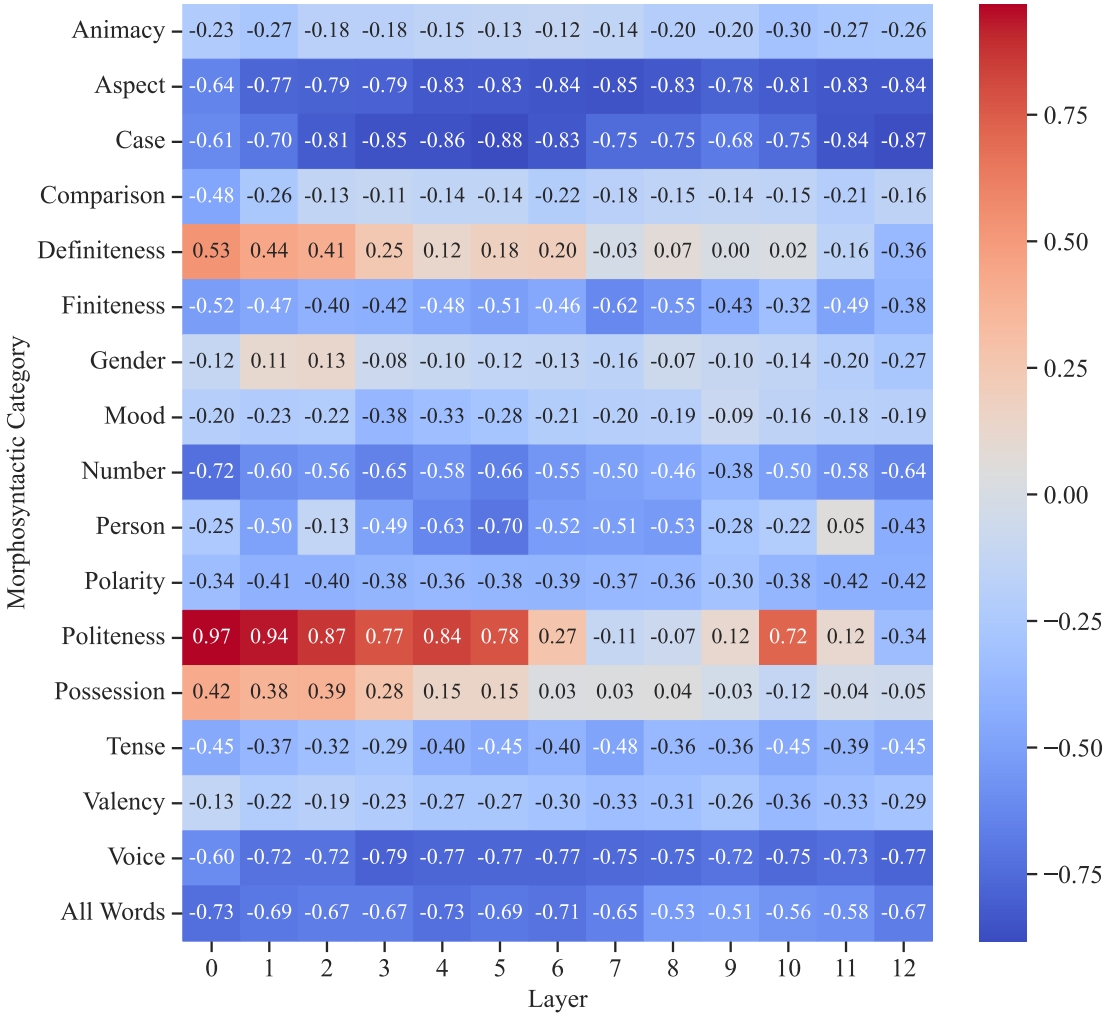


Figure B.1: Pearson correlation results between average $\mathrm{sig}(\ell)$ for all languages and their number of unique characters for each morphosyntactic category across all layers.

B.3 Additional Results for RQ3

We include in this section additional results for the performance prediction experiment.

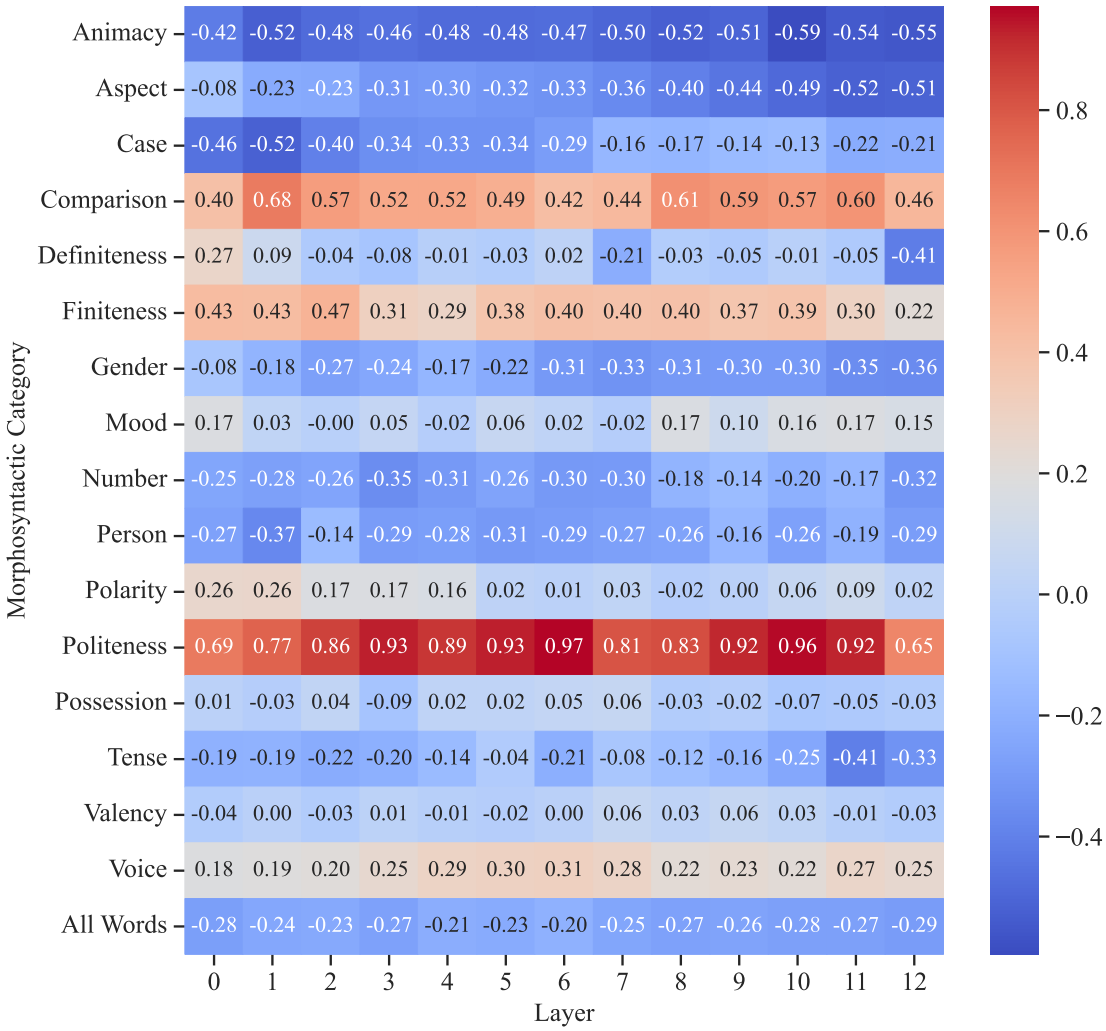


Figure B.2: Pearson correlation results between average $\mathrm{sig}(\ell)$ for all languages and their type-token ratio (TTR) for each morphosyntactic category across all layers.

B.3.1 Performance Prediction

We present the full correlation results for all layers for mBERT, XLM, XLM-R, and MMTE in Tables B.3 through B.6.

Property	**Language**
Animacy	bg, hr, cs, pl, ru, sk, sl, uk,
Aspect	ar, bg, ca, zh, hr, cs, et, fr, el, hi, hu, it, pl, pt, ro, ru, sk, sl, es, tr, uk
Case	ar, bg, ca, zh, hr, cs, da, nl, en, et, fi, fr, de, el, he, hi, hu, pl, pt, ro, ru, sk, sl, es, sv, tr, uk
Comparison	bg, hr, cs, da, nl, en, et, fi, fr, de, el, hu, it, pl, ro, ru, sk, sl, es, sv, uk
Definiteness	ar, bg, ca, hr, da, nl, en, fr, de, el, he, hu, it, pt, ro, sl, es, sv
Finiteness	ar, ca, hr, cs, da, nl, en, et, fi, fr, de, el, he, hi, hu, it, pl, pt, ro, ru, sk, sl, es, sv, uk
Gender	ar, bg, ca, hr, cs, da, nl, en, fr, el, hi, it, pl, pt, ro, ru, sk, sl, es, sv, uk
Mood	ar, bg, ca, hr, cs, da, en, et, fi, fr, de, el, he, hi, hu, it, pl, pt, ro, ru, sk, sl, es, sv, tr, uk
Number	ar, bg, ca, zh, hr, cs, da, nl, en, et, fi, fr, de, el, he, hi, hu, it, pl, pt, ro, ru, sk, sl, es, sv, tr, uk
Part of Speech	all 33 languages
Person	ar, bg, ca, zh, hr, cs, da, nl, en, et, fi, fr, de, el, he, hi, hu, it, pl, pt, ro, ru, sk, sl, es, tr, uk
Polarity	ar, bg, ca, zh, hr, cs, et, fi, fr, gl, de, he, hi, id, it, pl, pt, ro, ru, sk, sl, es, sv, tr, uk, vi
Politeness	ca, da, de, hi, es, tr
Possession	ar, ca, hr, cs, da, fi, de, he, hu, pl, ro, sk, sl, tr
Tense	bg, ca, hr, cs, da, nl, en, et, fi, fr, de, el, he, hi, hu, it, pl, pt, ro, ru, sk, sl, es, sv, tr, uk
Valency	bg, zh, hr, cs, da, nl, en, et, fi, fr, de, he, hu, pl, ro, ru, sk, sl, es, tr, uk
Voice	ar, bg, zh, hr, cs, da, et, fi, el, he, hi, pl, ru, sk, sv, tr, uk

Table B.2: All language properties (morphosyntactic categories) analyzed in this study along with languages that have the corresponding property.

Layer	Pair sentence		Structured prediction		Question answering			Sentence retrieval	
	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA-GoldP	BUCC	Tatoeba
	Acc.	Acc.	F1	F1	F1 / EM	F1 / EM	F1 / EM	F1	Acc.
0	0.01	0.71	0.26	0.47	0.33 / 0.15	0.03 / 0.15	0.61 / 0.20	0.64	-0.11
1	-0.03	0.72	0.23	0.43	0.25 / -0.00	-0.18 / -0.11	0.20 / -0.28	0.70	0.05
2	0.01	0.76	0.32	0.49	0.31 / 0.09	-0.15 / -0.04	0.36 / -0.16	0.68	0.05
3	0.18	0.71	0.33	0.49	0.41 / 0.13	-0.01 / 0.04	0.16 / -0.26	0.66	0.13
4	0.10	0.71	0.38	0.55	0.43 / 0.19	-0.01 / 0.10	0.41 / -0.12	0.65	0.12
5	0.16	0.69	0.33	0.51	0.49 / 0.25	0.06 / 0.15	0.45 / -0.03	0.59	0.11
6	0.24	0.74	0.41	0.59	0.54 / 0.28	0.12 / 0.20	0.54 / 0.03	0.66	0.18
7	0.16	0.70	0.29	0.41	0.43 / 0.21	0.02 / 0.11	0.35 / -0.07	0.69	0.10
8	0.09	0.56	0.14	0.20	0.32 / 0.12	-0.11 / -0.04	0.13 / -0.18	0.64	0.03
9	0.06	0.54	0.11	0.16	0.30 / 0.11	-0.16 / -0.07	0.12 / -0.16	0.72	-0.01
10	0.15	0.55	0.15	0.23	0.37 / 0.15	-0.02 / 0.04	0.14 / -0.13	0.71	0.01
11	0.18	0.58	0.24	0.32	0.42 / 0.19	0.02 / 0.09	0.17 / -0.13	0.67	0.03
12	0.36	0.67	0.36	0.46	0.60 / 0.35	0.23 / 0.31	0.41 / 0.05	0.72	0.15

Table B.3: Pearson correlations between $\mathrm{sig}(\ell)$ and XTREME benchmark performances of mBERT on various tasks.

Layer	Pair sentence		Structured prediction		Question answering			Sentence retrieval	
	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA-GoldP	BUCC	Tatoeba
	Acc.	Acc.	F1	F1	F1 / EM	F1 / EM	F1 / EM	F1	Acc.
0	-0.01	0.68	0.28	0.53	0.61 / 0.38	0.24 / 0.29	0.66 / 0.67	0.90	-0.02
1	-0.11	0.71	0.25	0.52	0.51 / 0.22	0.03 / 0.06	0.59 / 0.53	0.96	0.13
2	-0.07	0.74	0.34	0.57	0.54 / 0.27	0.05 / 0.11	0.65 / 0.60	0.97	0.12
3	0.09	0.70	0.37	0.55	0.65 / 0.36	0.23 / 0.22	0.42 / 0.36	0.98	0.21
4	0.05	0.70	0.39	0.59	0.68 / 0.40	0.21 / 0.27	0.58 / 0.54	0.94	0.20
5	0.09	0.67	0.34	0.54	0.72 / 0.46	0.27 / 0.31	0.50 / 0.48	0.93	0.18
6	0.17	0.72	0.41	0.63	0.75 / 0.48	0.32 / 0.35	0.67 / 0.65	0.94	0.27
7	0.08	0.68	0.29	0.46	0.69 / 0.45	0.26 / 0.28	0.53 / 0.51	0.95	0.19
8	-0.02	0.54	0.16	0.24	0.55 / 0.34	0.12 / 0.13	0.29 / 0.26	0.97	0.09
9	-0.04	0.51	0.12	0.20	0.53 / 0.32	0.09 / 0.11	0.25 / 0.23	0.92	0.06
10	0.07	0.52	0.17	0.26	0.61 / 0.38	0.23 / 0.22	0.29 / 0.28	0.92	0.08
11	0.10	0.56	0.25	0.34	0.65 / 0.40	0.25 / 0.26	0.33 / 0.31	0.97	0.11
12	0.30	0.65	0.36	0.46	0.81 / 0.56	0.46 / 0.48	0.43 / 0.43	0.96	0.24

Table B.4: Pearson correlations between $\mathrm{sig}(\ell)$ and XTREME benchmark performances of XLM on various tasks.

Layer	Pair sentence		Structured prediction		Question answering			Sentence retrieval	
	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA-GoldP	BUCC	Tatoeba
	Acc.	Acc.	F1	F1	F1 / EM	F1 / EM	F1 / EM	F1	Acc.
0	0.11	0.80	0.67	0.62	0.63 / 0.34	0.59 / 0.62	0.58 / 0.52	0.91	0.14
1	-0.03	0.80	0.61	0.57	0.51 / 0.14	0.39 / 0.40	0.61 / 0.47	0.84	0.21
2	0.02	0.83	0.67	0.64	0.58 / 0.24	0.46 / 0.50	0.63 / 0.47	0.85	0.23
3	0.13	0.78	0.66	0.62	0.55 / 0.19	0.43 / 0.45	0.37 / 0.16	0.85	0.19
4	0.15	0.80	0.73	0.68	0.71 / 0.39	0.56 / 0.61	0.59 / 0.46	0.90	0.31
5	0.17	0.78	0.69	0.63	0.70 / 0.40	0.61 / 0.63	0.51 / 0.43	0.93	0.28
6	0.24	0.82	0.73	0.70	0.73 / 0.42	0.63 / 0.64	0.68 / 0.62	0.89	0.35
7	0.15	0.78	0.62	0.53	0.57 / 0.26	0.52 / 0.55	0.53 / 0.46	0.87	0.25
8	0.04	0.66	0.47	0.33	0.36 / 0.05	0.32 / 0.37	0.30 / 0.23	0.88	0.13
9	0.03	0.63	0.44	0.30	0.34 / 0.03	0.29 / 0.35	0.26 / 0.19	0.85	0.10
10	0.13	0.64	0.49	0.36	0.41 / 0.10	0.37 / 0.41	0.29 / 0.24	0.86	0.11
11	0.19	0.67	0.56	0.43	0.52 / 0.20	0.45 / 0.49	0.36 / 0.34	0.85	0.15
12	0.36	0.75	0.66	0.55	0.73 / 0.45	0.64 / 0.68	0.46 / 0.46	0.83	0.28

Table B.5: Pearson correlations between $\mathrm{sig}(\ell)$ and XTREME benchmark performances of XLM-R on various tasks.

Layer	Pair sentence		Structured prediction		Question answering			Sentence retrieval	
	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA-GoldP	BUCC	Tatoeba
	Acc.	Acc.	F1	F1	F1 / EM	F1 / EM	F1 / EM	F1	Acc.
0	0.02	0.73	0.33	0.51	0.51 / 0.53	0.06 / -	0.92 / 0.35	0.59	-
1	-0.10	0.75	0.26	0.46	0.43 / 0.37	-0.15 / -	0.82 / 0.00	0.61	-
2	-0.06	0.78	0.38	0.53	0.48 / 0.44	-0.14 / -	0.93 / -0.04	0.58	-
3	0.02	0.73	0.36	0.51	0.54 / 0.42	0.07 / -	0.79 / -0.32	0.55	-
4	0.01	0.74	0.44	0.58	0.59 / 0.52	0.00 / -	0.91 / 0.06	0.58	-
5	0.08	0.71	0.39	0.54	0.64 / 0.57	0.07 / -	0.85 / 0.22	0.53	-
6	0.15	0.76	0.44	0.61	0.69 / 0.59	0.13 / -	0.91 / 0.31	0.58	-
7	0.07	0.73	0.35	0.43	0.59 / 0.57	0.09 / -	0.78 / 0.26	0.61	-
8	-0.02	0.59	0.21	0.23	0.46 / 0.44	0.00 / -	0.59 / 0.14	0.55	-
9	-0.04	0.56	0.19	0.21	0.44 / 0.44	-0.04 / -	0.54 / 0.16	0.66	-
10	0.05	0.58	0.20	0.27	0.51 / 0.46	0.10 / -	0.55 / 0.20	0.64	-
11	0.08	0.61	0.31	0.34	0.55 / 0.49	0.11 / -	0.54 / 0.28	0.57	-
12	0.21	0.69	0.40	0.46	0.72 / 0.61	0.28 / -	0.66 / 0.45	0.63	-

Table B.6: Pearson correlations between $\mathrm{sig}(\ell)$ and XTREME benchmark performances of MMTE on various tasks.

Appendix C

Appendix for Chapter 6

C.1 Dataset Details

This appendix provides a detailed overview of the datasets used in this study. We followed Wei et al. (2022a) and organized all tasks into the following task clusters:

- **Closed-book Question Answering (QA)** requires models to answer questions about the world without direct access to the answer-containing information.
- **Commonsense Reasoning** tests the capacity for physical or scientific reasoning infused with common sense.
- **Coreference Resolution** identifies expressions referring to the same entity within a given text.
- **Natural Language Inference (NLI)** focuses on the relationship between two sentences, typically evaluating if the second sentence is true, false, or possibly true based on the first sentence.
- **Paraphrase Detection** involves evaluating if two sentences have the same meaning. While it can be considered a form of bidirectional entailment, it remains distinct from NLI in academic contexts.
- **Reading Comprehension** assesses the ability to answer questions based on a given passage containing the necessary information.

- **Reading Comprehension with Commonsense** merges the tasks of reading comprehension and commonsense reasoning.
- **Sentiment Analysis** is a traditional NLP task that determines whether a text expresses a positive or negative sentiment.
- **Struct-to-Text** involves generating natural language descriptions from structured data.
- **Translation** is the task of translating text from one language to another.
- **Summarization** involves creating concise summaries from longer texts.
- **Unseen** clusters uses the original miscellaneous task cluster from Wei et al. (2022a) which includes:
    1. Conversational question-answering;
    2. Evaluating context-sentence word meanings;
    3. Linguistic acceptability;
    4. Math questions;
    5. Question classification.

We provide tasks contained in each cluster in Table C.1.

C.2 Additional Results

C.2.1 Results on Model Evaluation

We provide the results on all control models and the instruction-tuned Llama 2-SFT in Table C.3 and C.4 (for natural language understanding tasks) and Table C.5 (for natural language generation tasks). To further evaluate the validity of our instruction tuning, we also benchmark our models on two popular benchmark datasets: MMLU (Hendrycks et al., 2021) and BBH (Suzgun et al., 2022). We provide results in Table C.2. We can see that Llama 2-SFT outperforms Llama 2 on both of these benchmarks.

Task Cluster	**Dataset**	**Task Cluster**	**Dataset**
Natural language inference	ANLI CB MNLI QNLI SNLI WNLI RTE	Reading comprehension	BoolQ DROP MultiRC OBQA SQuADv1 SQuADv2
Commonsense reasoning	COPA HellaSwag PiQA StoryCloze	Sentiment analysis	IMDB Sentiment140 SST-2 Yelp
Closed-book QA	ARC NQ TriviaQA	Paraphrase detection	MRPC QQP Paws Wiki STS-B
Coreference resolution	DPR Winogrande WSC273	Reading comprehension w/ commonsense	CosmosQA ReCoRD
Struct to text	CommonGen DART E2ENLG WebNLG	Translation	En–Fr from WMT'14 WMT'16 En–Es from Paracrawl
Summarization	AESLC CNN-DM Gigaword MultiNews Newsroom Samsum XSum AG News Opinion Abstracts - Rotten Tomatoes Opinion Abstracts - iDebate Wikilingua English	Unseen	CoQA QuAC WiC TREC CoLA Math questions

Table C.1: Dataset details grouped by task clusters. For WMT'16, we include En–De, En–Tr, En–Cs, En–Fi, En–Ro, and En–Ru translation pairs. For all details about each dataset including the dataset size, please refer to Wei et al. (2022a).

	MMLU	**BBH**
Llama 2	41.25	32.82
Llama 2-SFT	47.81	37.49

Table C.2: Results for Llama 2 and Llama 2-SFT on MMLU and BBH. We use a 0-shot evaluation for MMLU to assess our models. For BBH, we follow the default evaluation protocol and use a 3-shot evaluation.

C.2.2 Results on Analysis

Here we provide additional results on our analysis. We provide the distribution of CKA similarities for all layers by tasks clusters in Figure C.1 and C.2. We also provide the t-SNE visualizations of representations in different layers of Llama 2 in Figure C.3 and C.4. Lastly, we provide the same visualizations for Llama 2-SFT in Figure C.5 and C.6.

Dataset	**Metric**	**Result**	
		Llama 2-SFT	Control Model
Natural Language Inference			
ANLI (r1)	Accuracy	51.87	54.45
ANLI (r2)	Accuracy	49.45	55.85
ANLI (r3)	Accuracy	47.48	54.14
CB	Accuracy	49.59	83.17
MNLI (matched)	Accuracy	87.25	88.64
MNLI (mismatched)	Accuracy	87.72	89.41
QNLI	Accuracy	83.00	86.46
SNLI	Accuracy	82.96	84.06
WNLI	Accuracy	71.22	69.64
RTE	Accuracy	81.52	81.21
Reading Comprehension			
BoolQ	Accuracy	83.53	88.18
DROP	F1	44.42	52.05
MultiRC	F1	72.19	73.92
OBQA	Accuracy	64.92	65.37
SQuADv1	F1	73.91	74.24
SQuADv2	F1	22.75	23.55
Commonsense Reasoning			
COPA	Accuracy	83.56	76.97
HellaSwag	Accuracy	71.43	73.49
PiQA	Accuracy	78.21	78.43
StoryCloze	Accuracy	85.81	84.82
Sentiment Analysis			
IMDB	Accuracy	72.06	74.54
Sentiment140	Accuracy	45.52	44.53
SST-2	Accuracy	79.14	79.03
Yelp	Accuracy	74.35	74.40

Table C.3: Performance metrics grouped by natural language understanding task clusters (part 1) for Llama 2-SFT and control models (Llama 2 model individually fine-tuned on each task). "Read. Comp. w/ Commonsense" denotes reading comprehension with commonsense.

Dataset	**Metric**	**Result**	
		Llama 2-SFT	Control Model
Closed-book QA			
ARC (Challenge)	Accuracy	59.09	52.83
ARC (Easy)	Accuracy	67.18	65.72
TriviaQA	Accuracy	59.00	59.26
NQ	Accuracy	28.79	31.18
Paraphrase Detection			
MRPC	Accuracy	78.35	84.73
QQP	Accuracy	84.91	87.37
PAWS Wiki	Accuracy	91.77	94.15
STS-B	Accuracy	47.46	51.20
Coreference Resolution			
DPR	Accuracy	85.12	72.53
Winogrande	Accuracy	69.68	69.93
WSC273	Accuracy	55.78	47.24
Read. Comp. w/ Commonsense			
CosmosQA	Accuracy	66.60	69.36
ReCoRD	Accuracy	85.13	85.78
Unseen			
CoQA	Accuracy	66.60	73.93
QuAC	Accuracy	18.29	33.99
WiC	Accuracy	56.47	70.77
TREC	Accuracy	57.05	80.25
CoLA	Accuracy	34.85	70.91
Math Questions	Accuracy	4.43	35.50

Table C.4: Performance metrics grouped by natural language understanding task clusters (part 2) for Llama 2-SFT and control models (Llama 2 model individually fine-tuned on each task). "Read. Comp. w/ Commonsense" denotes reading comprehension with commonsense.

Dataset	**Metric**	**Result**	
		Llama 2-SFT	Control Model
Struct-to-Text			
CommonGen	ROUGE-L	45.92	46.52
DART	ROUGE-L	55.46	57.28
E2ENLG	ROUGE-L	50.17	50.96
WebNLG	ROUGE-L	62.92	65.22
Translation			
WMT'14 En–Fr	BLEU	59.30	59.29
WMT'16 En–De	BLEU	56.84	57.45
WMT'16 En–Tr	BLEU	39.41	43.58
WMT'16 En–Cs	BLEU	46.92	47.21
WMT'16 En–Fi	BLEU	48.57	50.28
WMT'16 En–Ro	BLEU	56.03	57.70
WMT'16 En–Ru	BLEU	51.41	52.12
ParaCrawl En–Es	BLEU	54.76	56.39
Summarization			
AESLC	ROUGE-L	29.98	31.68
CNN-DM	ROUGE-L	17.38	19.59
Gigaword	ROUGE-L	28.69	30.22
MultiNews	ROUGE-L	15.17	16.61
Newsroom	ROUGE-L	18.95	22.43
Samsum	ROUGE-L	36.36	37.72
XSum	ROUGE-L	25.51	29.57
AG News	ROUGE-L	77.26	80.99
Opinion Abstracts - Rotten Tomatoes	ROUGE-L	19.36	21.70
Opinion Abstracts - iDebate	ROUGE-L	18.90	23.14
Wikilingua English	ROUGE-L	30.22	32.18

Table C.5: Performance metrics grouped by natural language generation task clusters for Llama 2-SFT and control models (Llama 2 model individually fine-tuned on each task).

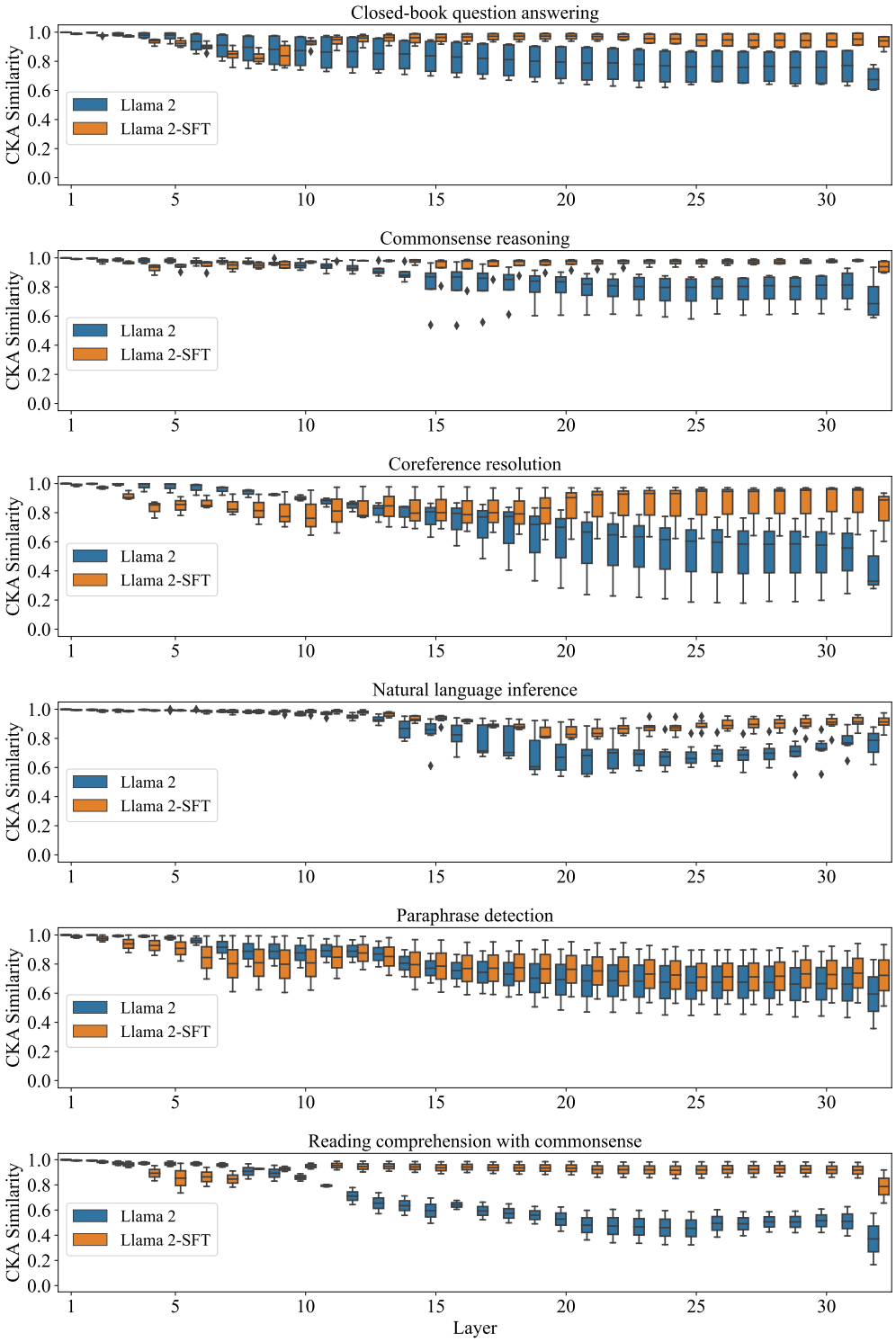


Figure C.1: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model, grouped by different task clusters.

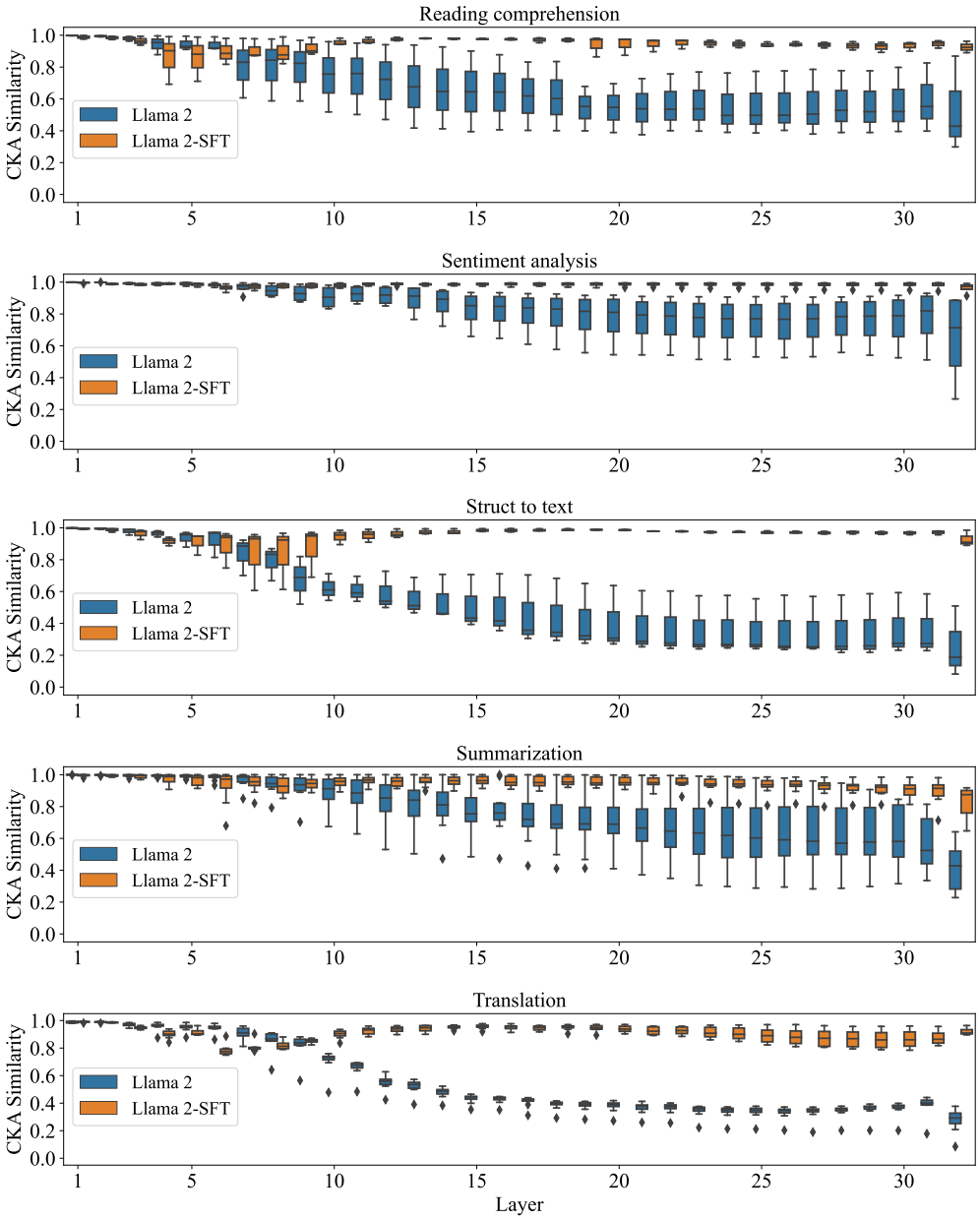


Figure C.2: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model, grouped by different task clusters.

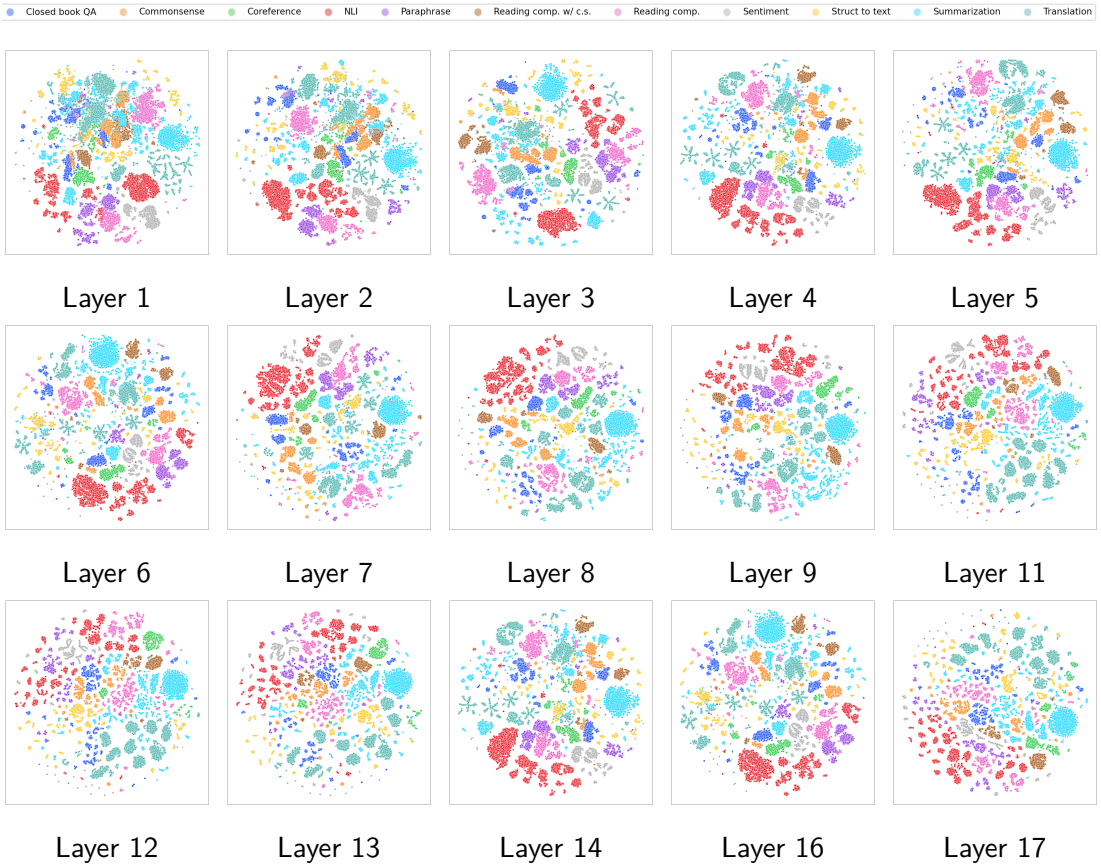


Figure C.3: t-SNE visualizations of the representations for each task cluster in different layers of the pre-trained Llama 2 model (part 1). Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer. "Reading comp." denotes reading comprehension tasks, and "reading comp. w/ c.s." denotes reading comprehension tasks with commonsense reasoning. We omit layers 10 and 15 as they are shown in the main chapter.

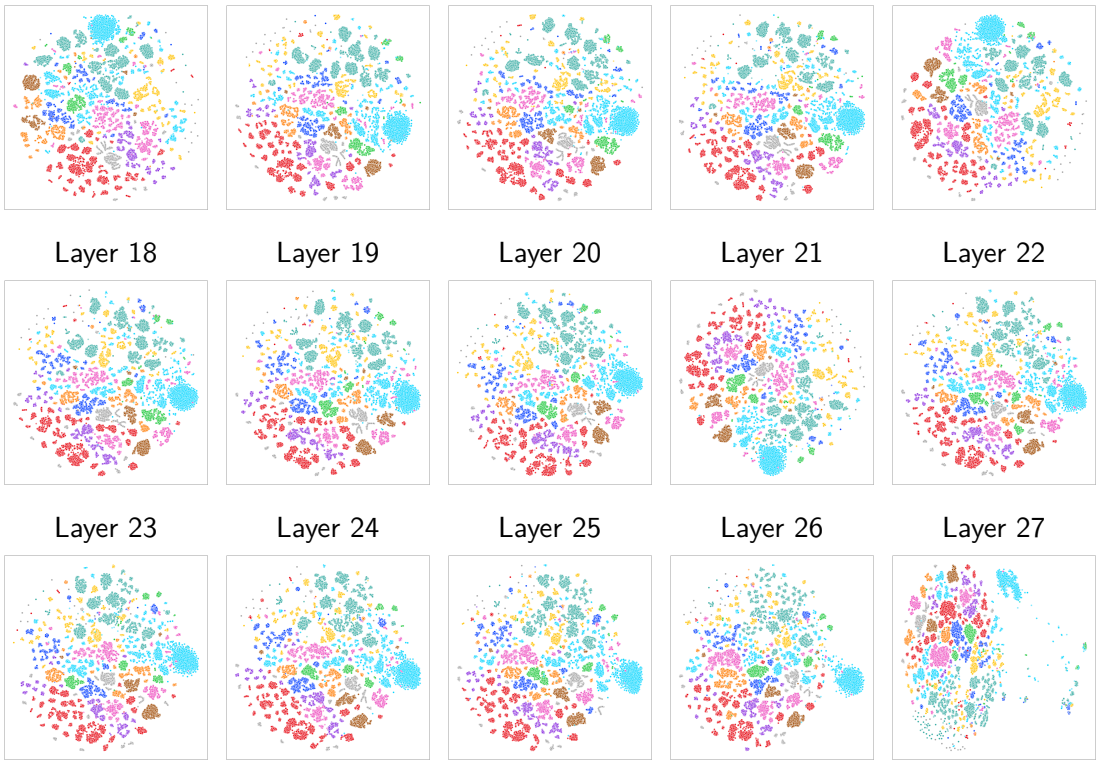


Figure C.4: t-SNE visualizations of the representations for each task cluster in different layers of the pre-trained Llama 2 model (part 2). Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer. "Reading comp." denotes reading comprehension tasks, and "reading comp. w/ c.s." denotes reading comprehension tasks with commonsense reasoning.

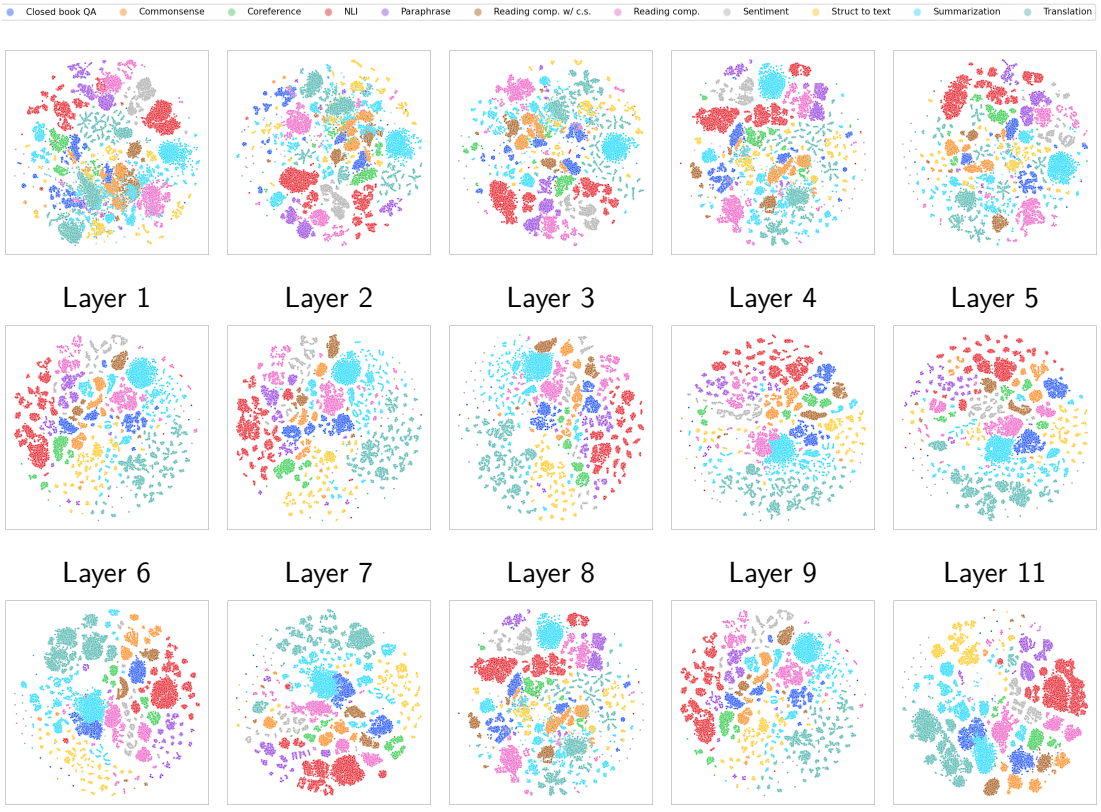


Figure C.5: t-SNE visualizations of the representations for each task cluster in different layers of the instruction-tuned Llama 2-SFT model (part 1). Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer. "Reading comp." denotes reading comprehension tasks, and "reading comp. w/ c.s." denotes reading comprehension tasks with commonsense reasoning. We omit layers 10 and 15 as they are shown in the main chapter.

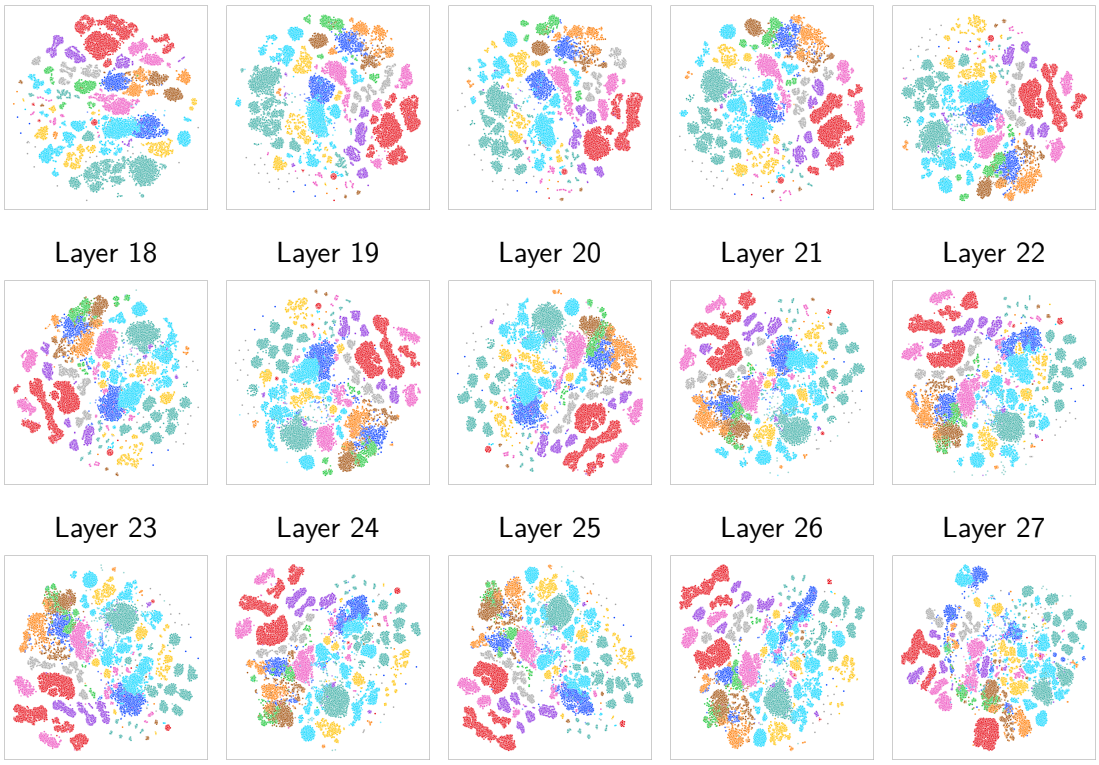


Figure C.6: t-SNE visualizations of the representations for each task cluster in different layers of the instruction-tuned Llama 2-SFT model (part 2). Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer. "Reading comp." denotes reading comprehension tasks, and "reading comp. w/ c.s." denotes reading comprehension tasks with commonsense reasoning.

Bibliography

Agarap, A. F. (2019). Deep learning using rectified linear units (relu).

Aghajanyan, A., Gupta, A., Shrivastava, A., Chen, X., Zettlemoyer, L., and Gupta, S. (2021). Muppet: Massive multi-task representations with pre-finetuning. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aharoni, R., Johnson, M., and Firat, O. (2019). Massively multilingual neural machine translation. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Alain, G. and Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *ArXiv*, abs/1610.01644.

Arivazhagan, N., Bapna, A., Firat, O., Lepikhin, D., Johnson, M., Krikun, M., Chen, M. X., Cao, Y., Foster, G. F., Cherry, C., Macherey, W., Chen, Z., and Wu, Y. (2019). Massively multilingual neural machine translation in the wild: Findings and challenges. *ArXiv preprint*, abs/1907.05019.

Artetxe, M., Ruder, S., and Yogatama, D. (2020). On the cross-lingual transferability of monolingual representations. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Artetxe, M. and Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback.

Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., and Glass, J. (2017a). What do neural machine translation models learn about morphology? In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Belinkov, Y., Màrquez, L., Sajjad, H., Durrani, N., Dalvi, F., and Glass, J. (2017b). Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In Kondrak, G. and Watanabe, T., editors, *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Zaenen, A. and van den Bosch, A., editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Caruana, R. (1997). Multitask learning. *Mach. Learn.*, 28(1):41–75.

Choenni, R. and Shutova, E. (2020). What does it mean to be language-agnostic? probing multilingual sentence encoders for typological properties. *CoRR*, abs/2009.12862.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. (2023). Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models.

Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., and Palomaki, J. (2020a). TyDi QA: A benchmark for information-seeking

question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020b). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Cohen, N., Kalinsky, O., Ziser, Y., and Moschitti, A. (2021). WikiSum: Coherent summarization dataset for efficient human-evaluation. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 212–219, Online. Association for Computational Linguistics.

Coleman, M. and Liau, T. L. (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Conneau, A. and Lample, G. (2019a). Cross-lingual language model pretraining. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Conneau, A. and Lample, G. (2019b). Cross-lingual language model pretraining. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., and Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Cortes, C., Mohri, M., and Rostamizadeh, A. (2012). Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res.*, 13(1):795–828.

de Vries, W., van Cranenburgh, A., and Nissim, M. (2020). What's so special about BERT's layers? a closer look at the NLP pipeline in monolingual and multilingual models. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350, Online. Association for Computational Linguistics.

Devlin, J. (2019). Multilingual BERT README.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Du, C., Sun, H., Wang, J., Qi, Q., and Liao, J. (2020). Adversarial and domain-aware BERT for cross-domain sentiment analysis. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028, Online. Association for Computational Linguistics.

Dubossarsky, H., Vulić, I., Reichart, R., and Korhonen, A. (2020). The secret is in the spectra: Predicting cross-lingual task performance with spectral similarity measures. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2377–2390, Online. Association for Computational Linguistics.

Fonseca, M., Ziser, Y., and Cohen, S. B. (2022). Factorizing content and budget decisions in abstractive summarization of long documents. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6341–6364, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Foroutan, N., Banaei, M., Lebret, R., Bosselut, A., and Aberer, K. (2022). Discovering language-neutral sub-networks in multilingual language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7575, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fujinuma, Y., Boyd-Graber, J., and Kann, K. (2022). Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1500–1512, Dublin, Ireland. Association for Computational Linguistics.

Gerz, D., Vulić, I., Ponti, E. M., Reichart, R., and Korhonen, A. (2018). On the relation between linguistic typology and (limitations of) multilingual language modeling. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 316–327, Brussels, Belgium. Association for Computational Linguistics.

Ginter, F., Hajič, J., Luotolahti, J., Straka, M., and Zeman, D. (2017). CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Giulianelli, M., Harding, J., Mohnert, F., Hupkes, D., and Zuidema, W. (2018). Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In Linzen, T., Chrupała, G., and Alishahi, A., editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of the 16th International Conference on Algorithmic Learning Theory*, ALT'05, page 63–77, Berlin, Heidelberg. Springer-Verlag.

Grusky, M., Naaman, M., and Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Hardoon, D. R., Szedmák, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 16(12):2639–2664.

Harshman, R. A. (1972). PARAFAC2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, 22:30–44.

Hazen, T. J., Dhuliawala, S., and Boies, D. (2019). Towards domain adaptation from limited data for question answering using deep neural networks. *arXiv preprint arXiv:1911.02655*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2021). Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hendrycks, D. and Gimpel, K. (2023). Gaussian error linear units (gelus).

Hewitt, J. and Liang, P. (2019). Designing and interpreting probes with control tasks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Hinkelmann, K. and Kempthorne, O. (2007). *Design and analysis of experiments, volume 1: Introduction to experimental design*, volume 1. John Wiley & Sons.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020). Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *ArXiv preprint*, abs/2003.11080.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

K, K., Wang, Z., Mayhew, S., and Roth, D. (2020). Cross-lingual ability of multilingual BERT: an empirical study. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kendall, M. G. (1948). *Rank correlation methods.* Griffin.

Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Kirov, C., Cotterell, R., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Mielke, S. J., McCarthy, A., Kübler, S., Yarowsky, D., Eisner, J., and Hulden, M. (2018). UniMorph 2.0: Universal Morphology. In Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kleindessner, M., Donini, M., Russell, C., and Zafar, M. B. (2023). Efficient fair pca for fair representation learning. In Ruiz, F., Dy, J., and van de Meent, J.-W., editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5250–5270. PMLR.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. E. (2019). Similarity of neural network representations revisited. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Kossaifi, J., Panagakis, Y., Anandkumar, A., and Pantic, M. (2019). Tensorly: Tensor learning in python. *Journal of Machine Learning Research*, 20(26):1–6.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Kudugunta, S., Bapna, A., Caswell, I., and Firat, O. (2019). Investigating multilingual NMT representations at scale. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Lekhtman, E., Ziser, Y., and Reichart, R. (2021). DILBERT: Customized pre-training for domain adaptation with category shift, with an application to aspect extraction. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 219–230, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020a). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lewis, P., Oguz, B., Rinott, R., Riedel, S., and Schwenk, H. (2020b). MLQA: Evaluating cross-lingual extractive question answering. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 7315–7330, Online. Association for Computational Linguistics.

Li, Y., Yosinski, J., Clune, J., Lipson, H., and Hopcroft, J. (2015). Convergent learning: Do different neural networks learn the same representations? In Storcheus, D., Rostamizadeh, A., and Kumar, S., editors, *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pages 196–212, Montreal, Canada. PMLR.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Re, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., WANG, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R. A., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. (2023). Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification, Outstanding Certification.

Liang, S., Dufter, P., and Schütze, H. (2021). Locating language-specific information in contextualized embeddings. *CoRR*, abs/2109.08040.

Libovický, J., Rosa, R., and Fraser, A. (2019). How language-neutral is multilingual bert? *ArXiv preprint*, abs/1911.03310.

Liu, C., Zhang, Q., Zhang, X., Singh, K., Saraf, Y., and Zweig, G. (2020). Multilingual graphemic hybrid ASR with massive data augmentation. In Beermann, D., Besacier, L., Sakti, S., and Soria, C., editors, *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 46–52, Marseille, France. European Language Resources association.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M.,

Zettlemoyer, L., and Stoyanov, V. (2019a). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Long, Q., Luo, T., Wang, W., and Pan, S. (2022). Domain confused contrastive learning for unsupervised domain adaptation. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2982–2995, Seattle, United States. Association for Computational Linguistics.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., and Roberts, A. (2023). The flan collection: Designing data and methods for effective instruction tuning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Magar, I. and Schwartz, R. (2022). Data contamination: From memorization to exploitation. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Mann, H. B. (1945). Nonparametric tests against trend. *Econometrica: Journal of the econometric society*, pages 245–259.

Maurer, A., Pontil, M., and Romera-Paredes, B. (2016). The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32.

McCarthy, A. D., Silfverberg, M., Cotterell, R., Hulden, M., and Yarowsky, D. (2018). Marrying Universal Dependencies and Universal Morphology. In de Marneffe, M.-C., Lynn, T., and Schuster, S., editors, *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Merchant, A., Rahimtoroghi, E., Pavlick, E., and Tenney, I. (2020). What happens to BERT embeddings during fine-tuning? In Alishahi, A., Belinkov, Y., Chrupała, G., Hupkes, D., Pinter, Y., and Sajjad, H., editors, *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Mishra, S., Khashabi, D., Baral, C., and Hajishirzi, H. (2022). Cross-task generalization via natural language crowdsourcing instructions. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Morcos, A. S., Raghu, M., and Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5732–5741.

Muller, B., Elazar, Y., Sagot, B., and Seddah, D. (2021). First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231, Online. Association for Computational Linguistics.

Ni, J., Li, J., and McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Nivre, J., Zeman, D., Ginter, F., and Tyers, F. (2017a). Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Nivre, J., Zeman, D., Ginter, F., and Tyers, F. (2017b). Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Olfat, M. and Aswani, A. (2019). Convex formulations for fair principal component analysis. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

OpenAI et al. (2024). GPT-4 technical report.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., and Ji, H. (2017). Cross-lingual name tagging and linking for 282 languages. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Pimentel, T., Valvoda, J., Maudslay, R. H., Zmigrod, R., Williams, A., and Cotterell, R. (2020). Information-theoretic probing for linguistic structure. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Pitler, E. and Nenkova, A. (2008). Revisiting readability: A unified framework for predicting text quality. In Lapata, M. and Ng, H. T., editors, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

Plank, B. (2016). What to do about non-standard (or non-canonical) language in NLP. In Dipper, S., Neubarth, F., and Zinsmeister, H., editors, *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19-21, 2016*, volume 16 of *Bochumer Linguistische Arbeitsberichte*.

Qiu, Y., Zhao, Z., Ziser, Y., Korhonen, A., Ponti, E., and Cohen, S. B. (2024). Spectral editing of activations for large language model alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Qiu, Y., Ziser, Y., Korhonen, A., Ponti, E., and Cohen, S. (2023). Detecting and mitigating hallucinations in multilingual summarisation. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8914–8932, Singapore. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. (2017). SVCCA: singular vector canonical correlation analysis for deep learning dynamics and

interpretability. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6076–6085.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rocktäschel, T., Huber, T., Weidlich, M., and Leser, U. (2013). WBI-NER: The impact of domain-specific features on the performance of identifying and classifying mentions of drugs. In Manandhar, S. and Yuret, D., editors, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 356–363, Atlanta, Georgia, USA. Association for Computational Linguistics.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N. V., Datta, D., Chang, J., Jiang, M. T., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Févry, T., Fries, J. A., Teehan, R., Scao, T. L., Biderman, S., Gao, L., Wolf, T., and Rush, A. M. (2022). Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Saphra, N. and Lopez, A. (2019). Understanding learning dynamics of language models with SVCCA. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

Volume 1 (Long and Short Papers), pages 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

Shakeri, S., Nogueira dos Santos, C., Zhu, H., Ng, P., Nan, F., Wang, Z., Nallapati, R., and Xiang, B. (2020). End-to-end synthetic data generation for domain adaptation of question answering systems. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.

Shang, J., Liu, L., Gu, X., Ren, X., Ren, T., and Han, J. (2018). Learning named entity tagger using domain-specific dictionary. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.

Shao, S., Ziser, Y., and Cohen, S. B. (2023a). Erasure of Unaligned Attributes from Neural Representations. *Transactions of the Association for Computational Linguistics*, 11:488–510.

Shao, S., Ziser, Y., and Cohen, S. B. (2023b). Gold doesn't always glitter: Spectral removal of linear and nonlinear guarded attribute information. In Vlachos, A. and Augenstein, I., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1611–1622, Dubrovnik, Croatia. Association for Computational Linguistics.

Singh, J., McCann, B., Socher, R., and Xiong, C. (2019). BERT is not an interlingua and the bias of tokenization. In Cherry, C., Durrett, G., Foster, G., Haffari, R., Khadivi, S., Peng, N., Ren, X., and Swayamdipta, S., editors, *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 47–55, Hong Kong, China. Association for Computational Linguistics.

Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Stanczak, K., Ponti, E., Torroba Hennigen, L., Cotterell, R., and Augenstein, I. (2022). Same neurons, different languages: Probing morphosyntax in multilingual pre-trained models. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1589–1598, Seattle, United States. Association for Computational Linguistics.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., , and Wei, J. (2022). Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Torroba Hennigen, L., Williams, A., and Cotterell, R. (2020). Intrinsic probing through dimension selection. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 197–216, Online. Association for Computational Linguistics.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams,

A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., and Korhonen, A. (2020). Probing pretrained language models for lexical semantics. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupała, G., and Alishahi, A., editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, J., Ding, K., Hong, L., Liu, H., and Caverlee, J. (2020). Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1101–1110.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Improving text embeddings with large language models.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022a). Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022b). Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proc. of EMNLP*.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yang, Y., Zhang, Y., Tar, C., and Baldridge, J. (2019). PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Yue, Z., Kratzwald, B., and Feuerriegel, S. (2021). Contrastive domain adaptation for question answering using limited text corpora. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9575–9593,

Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhang, C., Zhong, H., Zhang, K., Chai, C., Wang, R., Zhuang, X., Bai, T., Jiantao, Q., Cao, L., Fan, J., Yuan, Y., Wang, G., and He, C. (2025). Harnessing diversity for important data selection in pretraining large language models. In *The Thirteenth International Conference on Learning Representations*.

Zhang, K. and Bowman, S. (2018). Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In Linzen, T., Chrupała, G., and Alishahi, A., editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

Zhang, S., Zhang, D., Zhong, H., and Wang, G. (2020). A multiclassification model of sentiment for e-commerce reviews. *IEEE Access*, 8:189513–189526.

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., and Du, M. (2024a). Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2).

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2025). A survey of large language models.

Zhao, Z., Ziser, Y., and Cohen, S. (2022). Understanding domain learning in language models through subpopulation analysis. In Bastings, J., Belinkov, Y., Elazar, Y., Hupkes, D., Saphra, N., and Wiegreffe, S., editors, *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 192–209, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Zhao, Z., Ziser, Y., and Cohen, S. B. (2024b). Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*,

pages 15195–15214, Miami, Florida, USA. Association for Computational Linguistics.

Zhao, Z., Ziser, Y., Webber, B., and Cohen, S. (2023). A joint matrix factorization analysis of multilingual representations. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12764–12783, Singapore. Association for Computational Linguistics.

Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., and Ma, Y. (2024). Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Ziser, Y. and Reichart, R. (2017). Neural structural correspondence learning for domain adaptation. In Levy, R. and Specia, L., editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410, Vancouver, Canada. Association for Computational Linguistics.

Ziser, Y. and Reichart, R. (2018a). Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 238–249, Brussels, Belgium. Association for Computational Linguistics.

Ziser, Y. and Reichart, R. (2018b). Pivot based language modeling for improved neural domain adaptation. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1241–1251, New Orleans, Louisiana. Association for Computational Linguistics.

Zweigenbaum, P., Sharoff, S., and Rapp, R. (2017). Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In Sharoff, S., Zweigenbaum, P., and Rapp, R., editors, *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 60–67, Vancouver, Canada. Association for Computational Linguistics.