UNIVERSITY OF COPENHAGEN
DEPARTMENT OF COMPUTER SCIENCE

Ph.D. Thesis

Phillip Rust

Visual and Multilingual Representations

Towards Inclusive and Trustworthy Language Processing

Advisor: Anders Søgaard

This thesis has been submitted to the Ph.D. School of The Faculty of Science, University of Copenhagen on May 5, 2025.

Abstract

Language models are playing an increasingly central role in communication, information access, and decision-making worldwide. While their success indicates the potential for immense positive impact, it also creates an urgent need to ensure these systems are both inclusive—providing equitable access, treatment, and performance across a diverse range of users and linguistic contexts—and trustworthy, demonstrating consistent, fair, secure, and interpretable behavior in both operation and impact. However, the prevailing NLP paradigm, centered on text and heavily reliant on tokenization, often conflicts with these goals. It tends to exclude or poorly represent the majority of the world's linguistic diversity, struggles with non-digitized inputs such as historical documents, and can perpetuate bias or fail unpredictably, limiting equitable access and undermining trust. This thesis aims to address such challenges by investigating two directions: visual language representation learning, whereby models process language in various forms directly as pixel data to bypass limitations of tokenization, and multilingual language models, which aim to broaden coverage and performance across languages. We first show that these visual language models overcome tokenization bottlenecks when processing digital text, handling diverse scripts, orthographic variation, and linguistic noise more effectively. This approach also supports the direct analysis of non-digital texts, such as scanned historical documents, without relying on fragile OCR pipelines. Additionally, it extends naturally to under-served non-written forms of language: we present state-of-the-art, privacy-aware methods for sign language translation, addressing critical challenges of data scarcity and signer privacy. Complementary to this visual paradigm that naturally lends itself to multilingualism, the thesis offers an analysis of massively multilingual language models, highlighting trade-offs between core trustworthiness goals such as privacy, fairness across languages, and the ability to identify influential training data. Together, this thesis shows that visual and multilingual language representations can help in building language processing systems that are more inclusive and trustworthy, aligning with the broader goal of serving a diverse global population more effectively.

Resumé

Sprogmodeller spiller en stadig mere central rolle i kommunikation, informationsadgang og beslutningstagning på verdensplan. Selvom deres succes indikerer potentialet for enorm positiv indvirkning, skaber det også et presserende behov for at sikre, at disse systemer er både inkluderende—hvilket sikrer lige adgang, behandling og ydeevne på tværs af en bred vifte af brugere og sproglige kontekster—og pålidelige, idet de udviser konsekvent, fair, sikker og transparent adfærd i både drift og effekt. Imidlertid er det dominerende NLP-paradigme, som er centreret omkring tekst og stærkt afhængigt af tokenisering, ofte i modstrid med disse mål. Det har en tendens til at udelukke eller dårligt repræsentere størstedelen af verdens sproglige mangfoldighed, har svært ved at håndtere ikke-digitaliserede input såsom historiske dokumenter, og kan videreføre bias eller fejle uforudsigeligt, hvilket begrænser lige adgang og underminerer tilliden. Denne afhandling sigter mod at adressere sådanne udfordringer ved at undersøge to retninger: visuel sprogrepræsentationslæring, hvorved modeller behandler sprog i forskellige former direkte som pixeldata for at omgå begrænsningerne ved tokenisering, og flersprogede sprogmodeller, som sigter mod at udvide dækning og ydeevne på tværs af sprog. Vi viser, at disse visuelle sprogmodeller overvinder barrierer ved tokenisering, når de behandler digital tekst, og håndterer diverse skriftsystemer, ortografisk variation og sproglig støj mere effektivt. Denne tilgang understøtter også direkte analyse af ikke-digitale tekster, såsom scannede historiske dokumenter, uden at være afhængig af skrøbelige OCR-pipelines. Derudover udvides den naturligt til underbetjente ikke-skriftlige sprogformer: vi præsenterer state-of-the-art, privatlivsbevidste metoder til tegnsprogsoversættelse, der adresserer kritiske udfordringer som dataknaphed og tegnsprogsbrugere privatliv. Som et supplement til dette visuelle paradigme, der naturligt egner sig til flersprogethed, tilbyder afhandlingen en analyse af massivt flersprogede sprogmodeller, der fremhæver afvejninger mellem centrale mål for pålidelighed såsom privatliv, retfærdighed på tværs af sprog og evnen til at identificere indflydelsesrige træningsdata. Samlet set viser denne afhandling, at visuelle og flersprogede sprogrepræsentationer kan bidrage til at udvikle sprogbehandlingssystemer, der er mere inkluderende og pålidelige, i tråd med det overordnede mål om at betjene en mangfoldig global befolkning mere effektivt.

Acknowledgements

A Ph.D. journey is rarely smooth, and mine was no exception—but thanks to the support of many wonderful people, it became not only manageable, but truly meaningful and rewarding. I want to express my heartfelt gratitude to everyone who helped turn this experience into a great adventure.

I first would like to thank my advisor, Anders Søgaard, for welcoming me into this wonderful research group as a Ph.D. student, letting me pursue my interests freely, and supporting me with so much patience, optimism, and inspiration. Thank you also to Desmond Elliott, whose hands-on guidance, constructive feedback, and countless discussions helped me think critically, shape my research vision, and appreciate the value of biscuit breaks.

I sincerely thank my Ph.D. committee members—Serge Belongie, Mirella Lapata, and Goran Glavaš—for their thoughtful feedback, engaging discussions, and the time they generously committed to evaluating my work.

To all my brilliant collaborators—Jonas Lotz, Emanuele Bugliarello, Liz Salesky, Miryam de Lhoneux, Desmond Elliott, Nadav Borenstein, Isabelle Augenstein, Bowen Shi, Skyler Wang, Necati Cihan Camgöz, Jean Maillard, and Anders Søgaard—it has been a true pleasure working with you on the projects presented here. Our discussions, conference experiences, and late-night Overleaf sessions taught me so much about research and the joy of shared inquiry. Thank you for your insights, patience, and good humor.

Next, I'd like to thank my dear CoAStaL'ers: Mostafa Abdou, Rahul Aralikatte, Marcell Bollmann, Emanuele Bugliarello, Stephanie Brandl, Laura Cabello, Yong Cao, Ilias Chalkidis, Ruixiang Cui, Ruchira Dhar, Constanza Fierro, Stella Frank, Nicolas Garneau, Ana Valeria Gonzalez, Victor Hansen, Mareike Hartmann, Daniel Hershcovich, Antonia Karamolegkou, Yova Kementchedjhieva, İlker Kesen, Ali Al-Laith, Seolhwa Lee, Heather Lent, Miryam de Lhoneux, Jiaang Li, Wenyan Li, Jonas Lotz, Katerina Margatina, Lukas Nielsen, Ninell Oldenburg, Tommaso Pasini, Qiwei Peng, Rita Ramos, Vinit Ravishankar, Israfel Salazar, Danae Sánchez, Alice Schiavone, Monorama Swain, Yifei Yuan, Anna van Zee, Li Zhou, and Ingo Ziegler. Whether in the lab, at conferences, or on unforgettable trips, the energy of this group is unmatched. Thank you for all the shared experiences, conversations, and making Copenhagen feel like home from the beginning. I am immensely proud to have shared this chapter of my life with you.

At FAIR, thank you to Jean Maillard and the Seamless Communication team for hosting me as an intern in Menlo Park and providing guidance throughout. And, of course, thank you to my amazing intern cohort—Emanuele Aiello, Sihoon Choi, Yassir Fathullah, Ivona Najdenkoska, Jonas Schult, Yi-Lin Sung, Neha Verma, and Felix Wimbauer. All of you have made this internship a highlight of my Ph.D. Thank you also to Kevin Heffernan, Alex Mourachko, Tuan Tran, and the rest of the LCM team at FAIR for making my part-time role so engaging.

At Amazon, thank you to Marius Cotescu, Trevor Wood, and the ConvSpeech team for hosting me in Cambridge and teaching me the ways of applied science. Special appreciation for my co-interns, Sonal Sannigrahi and Armand Stricker, for the shared coffee/tea breaks and weekend wanderings around Cambridge and London.

I'd also like to thank Ivan Vulić, Anna Korhonen, and all the Ph.D. students at Cambridge LTL for being so welcoming during my brief visit. The research chats and personal conversations were both enriching and fun—and if not for this thesis, I would have loved to stay longer.

Thank you to my master's thesis supervisor, Jonas Pfeiffer, for the invaluable mentorship before the Ph.D. and making sure I know what I'm getting myself into. I'm also grateful to Harumi Kuno at Hewlett Packard Labs and Gözde Gül Şahin at TU Darmstadt for first showing me what a researcher even does.

Thank you to all the other Copenhagen NLP people, conference and summer school buddies, tokenization connoisseurs, and anyone I've inevitably missed in writing this. All of you have helped create memories I will cherish forever.

My deepest gratitude goes to my parents, my sister, and my grandparents for their unwavering love and support over the years. Your constant encouragement and belief in me have nurtured the intellectual curiosity, ambition, and perseverance that sustained me throughout this journey. I also want to thank my friends in Germany for the remote moral support, timely distractions, and occasional visits that recharged me when I needed it most. And thank you to Antonia—for being my biggest cheerleader, patient through distance, and a daily reminder of what this is all about. You're the main reason I crossed the finish line in good spirits, and I couldn't be more grateful.

A final thanks goes to all the cats who blessed me with their presence in the making of this thesis—meow the treats be with you!

Contents

List of Figures

List of Tables

List of Publications

This is an article-based thesis. The articles are identical in content as they appear here and in the original publications, except for minor changes such as the correction of typos and reformatting of tables, figures, references, and equations. The following articles are included as chapters in the thesis, listed in the order of their appearance within the document:

1. **Phillip Rust**, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda. (notable top 5%)

2. Jonas Lotz, Elizabeth Salesky, **Phillip Rust**, and Desmond Elliott. 2023. Text rendering strategies for pixel language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10155–10172, Singapore. Association for Computational Linguistics.

3. Nadav Borenstein, **Phillip Rust**, Desmond Elliott, and Isabelle Augenstein. 2023b. PHD: Pixel-based language modeling of historical documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 87–107, Singapore. Association for Computational Linguistics.

4. **Phillip Rust** and Anders Søgaard. 2023. Differential privacy, linguistic fairness, and training data influence: Impossibility and possibility theorems for multilingual language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29354–29387. PMLR.

5. **Phillip Rust**, Bowen Shi, Skyler Wang, Necati Cihan Camgoz, and Jean Maillard. 2024. Towards privacy-aware sign language translation at scale. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8624–8641, Bangkok, Thailand. Association for Computational Linguistics.

Below is a list of articles, in chronological order, that I co-authored over the course of my Ph.D., but not included as part of this thesis ($*$ denotes equal first-author contribution):

1. Laura Cabello Piqueras*, Constanza Fierro*, Jonas F. Lotz*, **Phillip Rust***, Joen Rommedahl, Jeppe Klok Due, Christian Igel, Desmond Elliott, Carsten B. Pedersen, Israfel Salazar, and Anders Søgaard. 2022. Date recognition in historical parish records. In *Frontiers in Handwriting Recognition*, pages 49–64, Cham. Springer International Publishing.

2. Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarello, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, **Phillip Rust**, and Anders Søgaard. 2022a. Challenges and strategies in cross-cultural NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6997–7013, Dublin, Ireland. Association for Computational Linguistics.

3. Jeppe Klok Due, Marianne Giørtz Pedersen, Sussie Antonsen, Joen Rommedahl, Esben Agerbo, Preben Bo Mortensen, Henrik Toft Sørensen, Jonas Færch Lotz, Laura Cabello Piqueras, Constanza Fierro, Antonia Karamolegkou, Christian Igel, **Phillip Rust**, Anders Søgaard, and Carsten Bøcker Pedersen. 2024. Towards more comprehensive nationwide familial aggregation studies in denmark: The danish civil registration system versus the lite danish multi-generation register. *Scandinavian Journal of Public Health*, 52(4):528–538. PMID: 37036022.

4. Antonia Karamolegkou, **Phillip Rust**, Ruixiang Cui, Yong Cao, Anders Søgaard, and Daniel Hershcovich. 2024. Vision-language models under cultural and inclusive considerations. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, pages 53–66, Bangkok, Thailand. ACL.

5. Antonia Karamolegkou, Malvina Nikandrou, Georgios Pantazopoulos, Danae Sanchez Villegas, **Phillip Rust**, Ruchira Dhar, Daniel Hershcovich, and Anders Søgaard. 2025b. Evaluating multimodal language models as visual assistants for visually impaired users. *arXiv preprint (under review)*.

6. Antonia Karamolegkou, Oliver Eberle, **Phillip Rust**, Carina Kauf, and Anders Søgaard. 2025a. Trick or Neat: Adversarial ambiguity and language model evaluation. *under review*.

Chapter 1

Introduction

Not long ago, *artificial intelligence (AI)* was still largely confined to science fiction books and a small academic field. The term first originated in the 1950s, but for most of its history, progress in AI and its subfield *natural language processing (NLP)* was slow, and systems relied heavily on rules and handcrafted feature vectors defined by human experts.

The landscape began to change with the introduction of hardware accelerators for machine learning, as exemplified by Krizhevsky et al. (2012)'s AlexNet model for image recognition. Combined with architectural advances and access to large labeled datasets like ImageNet (Deng et al., 2009), training deeper and wider neural networks suddenly became feasible. This kicked off a new wave of research in *deep learning*, first in computer vision and soon after spilling into NLP.

In NLP, this shift enabled researchers to move away from designing feature templates and linguistic pipelines. Instead, models began to learn *representations*—vector-based encodings of words, sentences, or entire documents that capture semantic and syntactic properties—directly from raw text. Early examples include distributed word embeddings like Word2Vec (Mikolov et al., 2013a,b) and GloVe (Pennington et al., 2014), which encoded words in a way that reflected their usage in context. This was the beginning of a broader move toward *representation learning*: the idea that instead of telling a model what to look for, we should give it the tools to figure that out on its own, guided by data. Sequence models like LSTMs and GRUs pushed this further (Hochreiter and Schmidhuber, 1997; Sutskever et al., 2014; Cho et al., 2014), allowing models to process and generate natural language in more coherent and context-sensitive ways.

This trend continued for a couple of years until, with the advent of transformer models (Vaswani et al., 2017) and large-scale pretraining (Radford et al., 2018; Devlin et al., 2019) around 2017/2018, the paradigm had entirely shifted from crafting task-specific features by hand to relying on the magic of linear algebra and calculus deployed at scale. Enabled by massive parallelization, models

could now learn deeply contextualized representations from web-scale text data through simple language modeling pretext tasks like cloze-style reconstruction and next-token prediction. What once required a team of linguists and feature engineers could now be done end-to-end with enough data and compute. This transformation is also succinctly captured by Turing Award laureate Rich Sutton's essay *The Bitter Lesson*,[1] which argues that the most effective methods in AI have consistently been those that leverage computation to learn general solutions, rather than relying on human knowledge and intuition.

Fast forward to today, this scaling trend has not slowed down, and NLP has entered an era of large language models (LLMs). Most notably, since the late-2022 LLM release of ChatGPT (OpenAI, 2022), these technologies are no longer confined to academic research laboratories; they are deployed across an unprecedented range of societal domains. LLMs now actively shape human interaction with technology (Phang et al., 2025), influence financial transactions (Li et al., 2023e), assist in healthcare diagnostics and delivery (Meng et al., 2024), automate decision-making in areas like loan approval and hiring (Fan, 2024), and help protect critical infrastructure (Yigit et al., 2025). They mediate our access to information, generate content, and facilitate communication for billions of users (Bommasani et al., 2021). This proliferation brings great potential for positive transformation, promising to, for instance, augment human capabilities and creativity (De Silva and Halloluwa, 2025), drive economic progress (Agrawal et al., 2019; Aghion et al., 2018), improve overall well-being (Stade et al., 2024), and accelerate scientific discovery (Wang et al., 2023a).

However, as Uncle Ben wisely reminds us in Spider-Man, *with great power comes great responsibility.* This sentiment is particularly relevant as we consider the increasing capabilities of AI systems and the societal responsibilities they entail. As these technologies are rolled out into the real world, we must continually ask ourselves: Who is empowered by these systems—and who is excluded? What are the risks for users and those whose data trained the models? Do we fundamentally understand how these models work? Can we rely on them to be accurate, no matter what inputs we feed them?

These questions, centered around the *inclusivity* and *trustworthiness* of language models, are open research problems. This thesis makes contributions to address some of the associated challenges, with a particular focus on studying *visual* and *multilingual* language representations as facilitators.

[1] http://www.incompleteideas.net/IncIdeas/BitterLesson.html

1.1 Structure of the Thesis

The thesis is organized as follows. In the remainder of this chapter, we first provide relevant technical background on representation learning (§ 1.2.1), then discuss current challenges related to inclusivity (§ 1.2.2) and trustworthiness (§ 1.2.3) in NLP and AI more broadly. We conclude the chapter by summarizing our scientific contributions (§ 1.3).

The subsequent chapters present the individual publications:

- § 2 Language Modelling with Pixels (Rust et al., 2023)
- § 3 Text Rendering Strategies for Pixel Language Models (Lotz et al., 2023)
- § 4 Pixel-Based Language Modeling of Historical Documents (Borenstein et al., 2023b)
- § 5 Differential Privacy, Linguistic Fairness, and Training Data Influence in Multilingual Language Models (Rust and Søgaard, 2023)
- § 6 Towards Privacy-Aware Sign Language Translation at Scale (Rust et al., 2024)

We conclude (in § 7) by discussing the thesis' contributions and limitations in the context of the current state of the field and highlighting avenues for future work.

1.2 Background and Challenges

This section provides relevant background to situate the work and contributions presented in the subsequent chapters. It begins with a brief overview of representation learning, followed by a discussion of challenges related to inclusivity and trustworthiness in NLP and AI.

1.2.1 Representation Learning

As mentioned in the opening section, *representation learning* refers to the idea that models can automatically extract useful features from raw data, rather than relying on handcrafted features. Formally, it involves learning a functional mapping $f : \mathcal{X} \rightarrow \mathcal{Z}$ from a data space $\mathcal{X}$ to an abstract representation space $\mathcal{Z}$ (Torralba et al., 2024). Illustrated in Figure 1.1, this mapping is typically implemented as a neural network referred to as an encoder.

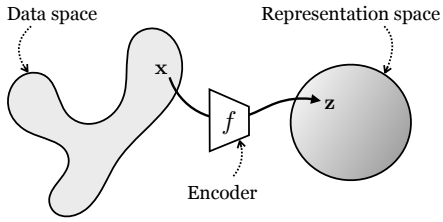


Figure 1.1: Schematic overview of representation learning. Figure reproduced from Torralba et al. (2024), licensed under CC-BY-NC-ND.

For example, suppose $\mathcal{X}$ is the set of all cat images. A sample $\mathbf{x} \in \mathcal{X}$ could then be a high-dimensional vector containing the pixel values of a photograph of a sleeping tabby. Its corresponding feature embedding $\mathbf{z} \in \mathcal{Z}$ would typically be a lower-dimensional (i.e., compressed) vector representation that captures salient aspects of the input, enabling downstream tasks such as predicting the cat breed (image classification) or locating the cat within the image (object detection).

When this encoder is a deep neural network, the mapping f is realized through a sequence of transformations, typically organized into layers

$$\mathbf{z} = f_L \circ f_{L-1} \circ \ldots \circ f_1(\mathbf{x}),$$

where each consecutive layer f_l maps its input representation to a more abstract one. In our example, early layers may extract edges and textures, while deeper layers form higher-level concepts such as animals or objects (Zeiler and Fergus, 2014; Zhou et al., 2015). These abstract features allow simple models, such as a linear classifier $h : \mathcal{Z} \to \mathcal{Y}$, to perform downstream tasks by operating directly on the learned representations. In the image classification setup, the final output $\mathbf{y} = h(f(\mathbf{x}))$ could predict the cat breed, perhaps leveraging the fact that certain breeds are easily distinguished by features such as fur color, texture, or ear shape.

Self-supervised representation learning The representation learning process can be guided in different ways. Traditional *supervised* approaches train the encoder f and the downstream classifier h together using a labeled dataset where each input $\mathbf{x}$ has a corresponding ground-truth label $\mathbf{y}$ (e.g., cat images labeled with “tabby” or “siamese”). Though effective (Krizhevsky et al., 2012; He et al., 2016; Dosovitskiy et al., 2021), this approach requires large amounts of often expensive labeled data, which makes scaling difficult.

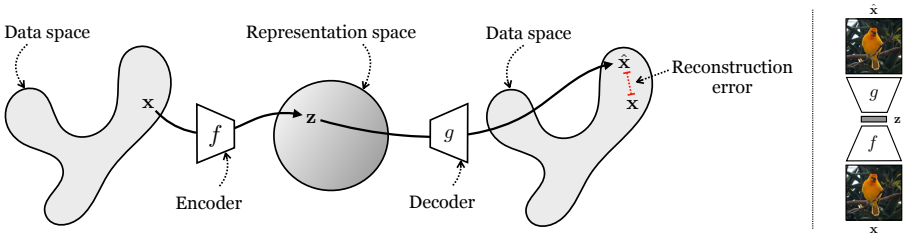


Figure 1.2: Schematic overview of an autoencoder. Figure reproduced from Torralba et al. (2024), licensed under CC-BY-NC-ND.

Instead, most state-of-the-art methods nowadays leverage *self-supervised learning (SSL)*, where the encoder is, at least initially,[2] trained without explicitly human-provided labels. One classic SSL approach is the autoencoder (AE; Rumelhart et al., 1986; Ballard, 1987), illustrated in Figure 1.2. An autoencoder combines the encoder f with a decoder g that aims to reconstruct the original input $\mathbf{x}$ from the compressed representation z, i.e., $\hat{\mathbf{x}} = g(f(x)) \approx \mathbf{x}$. The objective is typically to minimize a reconstruction loss, e.g. the squared error $||\hat{\mathbf{x}} - \mathbf{x}||_2^2$. For the purpose of downstream tasks such as classification, the decoder g can be discarded after training, leaving the encoder f as the desired representation learner.

A highly effective recent SSL method that combines the autoencoder framework with the idea of an *imputation pretext task*—learning to predict missing data with the aim of learning better representations as a by-product—is the masked autoencoder (MAE) (Devlin et al., 2019; He et al., 2022; Feichtenhofer et al., 2022). MAEs randomly mask a portion of the input (e.g., image patches, spatio-temporal video patches, or text tokens) and train the encoder-decoder network to reconstruct the original, unmasked input $\mathbf{x}$. This forces the encoder f to learn a rich semantic understanding of the visible parts to infer the missing content. Another classic variant of AEs is the denoising autoencoder (DAE), which applies corruptions rather than masks to its inputs (Vincent et al., 2008, 2010; Lewis et al., 2020). Variants of DAEs are found in state-of-the-art generative modeling approaches such as diffusion models (Ho et al., 2020), and are starting to make a comeback in representation learning (Chen et al., 2025).

In addition to reconstruction-based methods, *contrastive learning* offers another major SSL paradigm, learning representations that are invariant to certain data

[2]The encoder parameters are sometimes overwritten during a subsequent supervised finetuning stage (transfer learning), as described in the following paragraphs.

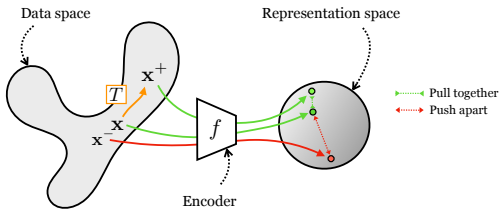


Figure 1.3: Schematic overview of contrastive learning. Figure reproduced from Torralba et al. (2024), licensed under CC-BY-NC-ND.

augmentations or transformations by pulling representations of related views (positive pairs) together while pushing unrelated views (negative pairs) apart (Hadsell et al., 2006; Chen et al., 2020; He et al., 2020). This idea is illustrated in Figure 1.3. Although contrastive learning is often used in SSL, it is not limited to this setting; state-of-the-art multimodal approaches such as CLIP (Radford et al., 2021a), VideoCLIP (Xu et al., 2021), CoCa (Yu et al., 2022b), and SigLIP (Zhai et al., 2023) leverage contrastive objectives with supervision from highly scalable, weakly paired data, such as image-caption pairs commonly found on the web.

This thesis extensively features SSL techniques: we train image- and video-based MAEs (§ 2, 3, 4, 6) and video–text contrastive learners (§ 6).

Transfer learning The *pretrained* representations resulting from SSL often form the basis for *transfer learning*: the encoder f, hopefully having learned generalizable features from a large (usually general-purpose) dataset, is reused and adapted *(finetuned)* to solve new downstream tasks with significantly less task-specific labeled data. This framework has largely been popularized in the context of NLP (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019).

It may often be preferable (e.g., for efficiency or modularity reasons) not to finetune the full encoder, as this entails computing gradients for, and overwriting, all of its parameters (Pfeiffer et al., 2023). Popular alternatives in such cases are parameter-efficient finetuning (PEFT) techniques such as low-rank adaptation (LoRA; Hu et al., 2022), SSL approaches that are explicitly optimized not to require further finetuning (frozen features) such as DINOv2 (Oquab et al., 2024), and in-context learning (prompting) approaches for LLMs, as introduced with GPT-3 (Brown et al., 2020).

We employ transfer learning throughout all chapters of this thesis, primarily relying on full finetuning (§ 2, 3, 4, 5, 6) and frozen features (§ 6).

A note on terminology

We briefly clarify some ambiguous terms before moving on:

- *The process of representation* vs. *a vector representation*: representation can refer to the overarching method or process for encoding data into a numerical format suitable for neural networks (e.g., tokenization, rendering text as pixels); or it can refer to the randomly initialized or learned vector representation(s) (also used interchangeably with *embedding* and *feature*). The *representation learning* process naturally subsumes these two terms.
- *Text* vs. *language*: text refers to written language; language is not necessarily written (e.g., speech and sign language). *Language models* often have text-based inputs/outputs, but learn to represent language in the abstract sense.
- *Visual* vs. *pixel-based*: *pixel-based* denotes the specific input format (images, video frames), while *visual* refers to the broader sensory modality; as the visual methods discussed herein rely exclusively on pixel data, we often use *visual language representation(s)* as the main term and may use *visual* and *pixel-based* interchangeably in this specific context.

Text representation Digital text, in its raw symbolic form, must first be converted into numerical sequences suitable for neural network processing. The conventional way to do this is through *tokenization*, which refers to the process of segmenting a piece of text into a sequence of discrete vocabulary units, called *tokens*. Each token in the vocabulary maps to a designated embedding vector in a lookup table. Table 1.1 illustrates segmentations produced by different tokenization strategies, which we discuss below. Taking the subword-level segmentation, for example, would give an initial embedding sequence $\mathbf{x} \in \mathbb{R}^{14\times d}$, where d is the model's embedding dimension (e.g., $d = 768$).

Early approaches typically used *word-level* tokenization, learning distinct embeddings for each unique word in the vocabulary (Mikolov et al., 2013a,b; Pennington et al., 2014). The first sequence-to-sequence models similarly relied on word-level embeddings, often limited to the top-k most frequent words in the corpus due to computational constraints (Sutskever et al., 2014; Cho et al.,

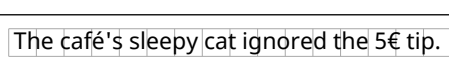
Unit	Tokenized or rendered sentence	Length
Words	The café's sleepy cat ignored the 5€ tip.	9
Characters	T h e c a f é ' s s l e e p y c a t i g n o r e d t h e 5 € t i p .	41
Bytes (UTF-8 Hex)	54 68 65 20 63 61 66 C3 A9 27 73 20 73 6C 65 65 70 79 20 63 61 74 20 69 67 6E 6F 72 65 64 20 74 68 65 20 35 E2 82 AC 20 74 69 70 2E	44
Subwords	The café 's sleepy cat ignored the 5€ tip.	14
Pixels		16*

Table 1.1: Example of segmentations from different tokenization strategies (top) and image rendered using the visual text representation strategy from § 2 (Rust et al., 2023). ∗: Number of square image patches from grid slicing.

2014). Generally, word-level tokenization has two primary limitations. First, the vocabulary either becomes excessively large or lacks coverage. Large vocabularies make the standard softmax computation over all possible output tokens computationally intractable during prediction, which has led to approximations like hierarchical softmax or negative sampling (Mikolov et al., 2013a). Second, the Zipfian nature of word distributions in language (Zipf, 1935, 1949) leads to data sparsity: most words are rare, making it difficult to learn meaningful representations for them through limited exposure during training.

Other early language models operated at the *character-level* (Ling et al., 2015b,a; Costa-jussà and Fonollosa, 2016), reducing vocabulary size and sparsity while improving coverage, but with drawbacks such as long sequences (see Table 1.1) and reduced performance. Similarly, mapping text directly to its underlying UTF-8 byte representation offers a method with a fixed, small vocabulary (256 units). This inherently avoids out-of-vocabulary items (open vocabulary) and mitigates sparsity. However, such *byte-level* representation also results in long sequences (see Table 1.1), often requiring specialized architectural strategies for length reduction (Yu et al., 2023; Pagnoni et al., 2024), and has traditionally yielded lower performance than subword methods, which we discuss next, although recent work shows competitive results (Pagnoni et al., 2024).

To mitigate the drawbacks of these rule-based approaches, data-driven *subword tokenization* strategies were proposed. These algorithms learn to segment words into parts (subwords)—in the extreme case, all the way back down to characters or bytes—effectively balancing sparsity, coverage, sequence lengths, and vocabulary size. In particular, byte-pair encoding (BPE; Sennrich et al., 2016; Kudo and

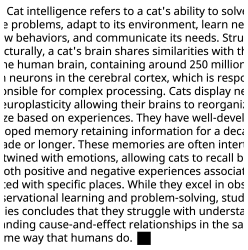
Richardson, 2018), originally proposed in the context of data compression (Gage, 1994), has become the de facto standard tokenization algorithm. Virtually all language models nowadays, from BERT (Devlin et al., 2019) over GPT variants (Radford et al., 2019; Brown et al., 2020; OpenAI et al., 2024), to Llama models (Touvron et al., 2023a,b; Grattafiori et al., 2024), and DeepSeek (DeepSeek-AI et al., 2025) build on BPE or variants thereof. BPE builds a token vocabulary through iterative merge operations. It starts from individual characters or bytes and, at every step, combines the most frequent pair of units in the training corpus into a new subword unit, effectively capturing common character sequences, until a desired vocabulary size is reached. Common vocabulary sizes in models' subword tokenizers range from around 30K in monolingual models (Devlin et al., 2019) up to around 256K in multilingual models (Conneau et al., 2020a; Xue et al., 2021; Gemma Team et al., 2025).

Once a tokenizer is trained, it enables the learning of contextualized text representations through pretraining objectives. Common strategies include masked language modeling (MLM) and next-token prediction, also referred to as (causal) language modeling (LM). MLM was introduced with BERT, which can be considered a variant of an MAE: the model learns to reconstruct randomly masked tokens within a sequence. In fact, He et al. (2022)'s popular MAE approach in computer vision was inspired by the success of MLM in BERT. Next-token prediction was introduced by Bengio et al. (2003) and popularized for pretraining with OpenAI's GPT series. As the name suggests, the learning objective is to causally predict the next token from the past, one token at a time.[3]

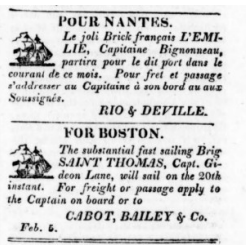
Despite the effectiveness of subword tokenization combined with these pretraining methods, the token-based paradigm has crucial limitations (discussed further in § 1.2.2; § 1.2.3; § 2; § 3; § 4). Issues related to vocabulary coverage for morphologically rich or low-resource languages, handling of visual aspects like layout or font, brittleness to noise, and the fundamental mismatch for non-written language have motivated exploring alternative paradigms.

Visual language representation Addressing such limitations inherent in token-based methods, this thesis explores *visual language representation*. The key idea of this alternative paradigm is to rely on pixel-based views of language data, which are then processed with image and video encoders.

[3] In transformers, this process is efficiently parallelized during training using a causal masking strategy and teacher-forcing (Williams and Zipser, 1989).


Cat intelligence refers to a cat's ability to solv
e problems, adapt to its environment, learn ne
w behaviors, and communicate its needs. Stru
cturally, a cat's brain shares similarities with th
he human brain, containing around 250 millior
n neurons in the cerebral cortex, which is respc
onsible for complex processing. Cats display ne
uroplasticity allowing their brains to reorgani
ze based on experiences. They have well-devel
oped memory retaining information for a dec
ade or longer. These memories are often inter
twined with emotions, allowing cats to recall b
oth positive and negative experiences associat
ed with specific places. While they excel in obs
ervational learning and problem-solving, stud
ies concludes that they struggle with understa
nding cause-and-effect relationships in the sa
me way that humans do. ■

(a) Digital text (§ 2,3) [4]

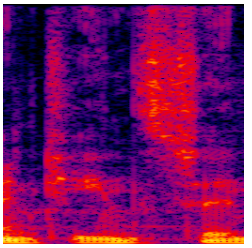

POUR NANTES.
Le joli Brick français L'EMILIE, Capitaine Bignonneau, partira pour le dit port dans le courant de ce mois. Pour fret et passage s'adresser au Capitaine à son bord au aux Soussignés.
RIO & DEVILLE.
FOR BOSTON.
The substantial fast sailing Brig SAINT THOMAS, Capt. Gideon Lane, will sail on the 20th instant. For freight or passage apply to the Captain on board or to
CABOT, BAILEY & Co.
Feb. 6.

(b) Printed text (§ 4) [5]

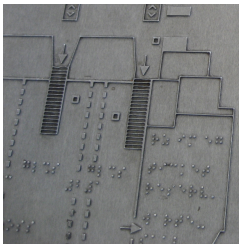

Soon the rabbit noticed Alice, as she stood looking curiously about her: and at once put in a quick angry tone, 'why, Mary Ann! what are you doing out here? Go home this moment, and look on my dressing-table for my gloves and nosegay, and fetch them here, as quick as you can run, do you hear?' and Alice was so much frightened that she ran off at once, without

(c) Handwritten text [6]

(d) Sign language (§ 6) *[7]



(e) Spectrogram [8]



(f) Braille [9]

Figure 1.4: Examples of language in the visual modality. ∗: video.

As shown in Figure 1.4, this approach naturally unifies the representation of language across different visual forms: it subsumes pixel-based representations of written language—digitally rendered as images (§ 2, 3) or scanned from physical documents (§ 4)—and extends seamlessly to inherently visual languages such as sign language, learned directly from video recordings (§ 6). By operating on raw visual data (pixels), visual language representation bypasses many issues related to tokenization granularity, vocabulary coverage, and the handling of multilingual or visually complex scripts, offering a more universal approach to learning from language data. Earlier approaches for visual text representation learning include Broscheit (2018)'s and Salesky et al. (2021)'s works on machine translation. Conveniently, the computer vision and NLP communities are increasingly converging on the same architectures and methods for representation learning. In particular,

[4] From Wikipedia, CC BY-SA 4.0, https://en.wikipedia.org/wiki/Cat, rendered as image (§ 2).
[5] From the "Caribbean Newspapers, 1718–1876" database (https://www.readex.com/products/caribbean-newspapers-series-1-1718-1876-american-antiquarian-society) used in § 4.
[6] From Public Domain (Carroll, 1865b).
[7] From DailyMoth-70h, CC BY-NC 4.0 (§ 6; Rust et al., 2024).
[8] From Public Domain, https://en.m.wikipedia.org/wiki/File:Spectrogram-19thC.png.
[9] From Geogast, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=28931878.

variants of the transformer architecture (Vaswani et al., 2017) and SSL methods such as MAE and contrastive learning are widely used for both language and image processing.[10] Our proposed methods for visual language representation learning build on these joint advances.

Summary Representation learning is a powerful tool at the core of many recent breakthroughs in AI. SSL frameworks such as MAE, contrastive learning, and next-token prediction are conceptually simple. Yet, when applied at massive scale (e.g., state-of-the-art large language models consume over 10 trillion tokens), they have paved the way for the immense success of today's models.[11]

Building on the concept of *visual language representation* introduced above, central contributions of this thesis lie in applying this paradigm to address challenges in inclusivity and trustworthiness within NLP and AI. We demonstrate how processing language visually—via rendered images for digital text (§ 2, 3), scans for non-digitized text (§ 4), and video for sign language (§ 6)—avoids many limitations of traditional token-based methods. We now turn to our discussion of such inclusivity and trustworthiness challenges.

1.2.2 Inclusivity Challenges in NLP

Cambridge dictionary defines inclusivity as *"the fact of including all types of people, things or ideas and treating them all fairly and equally."*[12] As discussed in the opening statement, NLP technology's influence on society has been growing rapidly and will continue to shape our lives in the years to come. Ensuring they serve everyone equitably, regardless of linguistic, cultural, or sociodemographic background, therefore becomes an ethical and practical necessity. Failure to do so risks perpetuating and even amplifying existing societal inequalities such as the growing digital divide (Lythreatis et al., 2022).

Historically, NLP as a field has been heavily skewed towards English. Factors contributing to this Anglocentrism include the early availability of large digital English-language datasets (like the Penn Treebank (Marcus et al., 1993) or Brown Corpus (Francis and Kucera, 1979)), the concentration of research funding and

[10] Notably, this convergence on the same tools has also facilitated multimodal (e.g., vision-and-language) processing (Bordes et al., 2024).

[11] Beyond such self-supervised pretraining, current LLMs also employ *post-training*, including strategies such as instruction finetuning (Wei et al., 2022) and reinforcement learning from human feedback (RLHF; Ouyang et al., 2022). However, it is widely understood that a capable pretrained base model is a prerequisite for these techniques to be effective (DeepSeek-AI et al., 2025).

[12] https://dictionary.cambridge.org/dictionary/english/inclusivity

institutions in English-speaking regions, and the initial focus on computational challenges solvable with existing English resources (Bender, 2011; Bird, 2020).

However, this focus on the English language has led to a significant disparity, leaving the vast majority of the world's 7000+ spoken languages under-served by modern NLP technologies (Joshi et al., 2020b; Ritchie et al., 2024). Towards closing this gap, there has been a collective push in recent years to include languages beyond English (Ruder, 2020; Costa-jussà et al., 2022), even proposing drastic measures such as to place a temporary ban on English NLP research (Søgaard, 2022).

A key ingredient in this effort to scale beyond English has been the development of massively multilingual (large) language models (MLLMs), such as multilingual BERT (mBERT; Devlin et al., 2019), XLM-R (Conneau et al., 2020a), mT5 (Xue et al., 2021), and BLOOM (Scao et al., 2022), which are jointly trained on large datasets comprising text in the order of 100 languages. By facilitating *cross-lingual transfer* (Wu and Dredze, 2019; Pires et al., 2019), the ability to perform tasks in one language (often a low-resource one) after being trained primarily on others (often high-resource ones), these models have leapfrogged the state of the art for non-English languages. Recent advances in this direction have largely been driven by model and compute scaling and the introduction of larger, higher-quality multilingual datasets. Examples include Aya (Singh et al., 2024; Üstün et al., 2024), which focuses on instruction tuning data, and Glot500 (Imani et al., 2023), which targets 500 languages. Despite these advances, we are still far from adequately covering 1000+ languages, and several crucial challenges persist.

Data scarcity Firstly, *data scarcity* remains the greatest barrier for the vast majority of languages (the "long tail") (Ritchie et al., 2024). These *low-resource languages* are often limited by their small digital footprint and lack of stakeholders, despite having (hundreds of) millions of users (Joshi et al., 2020b). This includes, but is not limited to, many African (Nekoto et al., 2020; Adebara and Abdul-Mageed, 2022), Southeast Asian (Aji et al., 2022, 2023), and Indian (Khanuja et al., 2023) languages. In particular, training data in such languages lack not only in volume, but also in quality (Kreutzer et al., 2022; Tatariya et al., 2024b), and it is evident that relying on positive transfer within MLLMs to fill these data gaps is not an adequate long-term strategy (Rajaee and Monz, 2024). As such, claims of "training an MLLM for [X] languages" may also be misleading; in reality, these [X] languages are not represented nearly equitably. Community and research

initiatives such as Masakhane,[13] IndoNLP,[14] AI4Bharat, [15] and AmericasNLP (Mager et al., 2024) are actively fighting this issue, but its scale is immense. Joshi et al. (2020b) state that for especially low-resource languages *"it will be a monumentous, probably impossible effort to lift them up in the digital space."*

Data scarcity also affects the crucial model evaluation process. Standardized benchmarks exist for far fewer languages than those included in MLLMs, resulting in overconfident claims about language diversity (Ploeger et al., 2024), and often forcing reliance on indirect evaluation through zero-shot cross-lingual transfer from English or on machine translation-based metrics, which may not accurately reflect real-world utility (Artetxe et al., 2020b; Choenni et al., 2024a).

We address data scarcity in sign language translation (§ 6) by designing a visual language representation learning framework that enables training on unannotated videos, rather than being limited to videos with paired translations, and by releasing a new American Sign Language (ASL) benchmark dataset.

Representational bottlenecks A commonly known limitation of MLLMs is the *curse of multilinguality* (Conneau et al., 2020a): as model capacity is shared across an increasing number of languages, performance on individual languages, especially low-resource ones, can degrade compared to bilingual or monolingual models. While this phenomenon has been investigated (Rust et al., 2021; Chang et al., 2024) and mitigated (Pfeiffer et al., 2020, 2022; Choenni et al., 2024b) in recent MLLMs, it will likely continue to present a hurdle in scaling to 1000+ languages (presuming adequate data coverage).

Fundamental representational limitations of *tokenization* also pose a major challenge for language inclusion. As explained in § 1.2.1, conventional (large) language models represent text using a finite vocabulary of tokens, typically learned through BPE (Sennrich et al., 2016; Kudo and Richardson, 2018). Although effective for handling unknown words, reducing sparsity, and controlling vocabulary size (Sennrich et al., 2016; Huck et al., 2017; Kudo, 2018), these algorithms become prohibitive when attempting to scale to thousands of languages. The core issue is a trade-off we refer to as the *vocabulary bottleneck* (§ 2; Rust et al., 2023): a fixed vocabulary forces languages to compete for limited space (akin to the curse of multilinguality), while expanding the vocabulary to accommodate all languages makes the softmax computation during prediction intractable. While alternatives

[13] https://www.masakhane.io/
[14] https://indonlp.github.io/
[15] https://ai4bharat.iitm.ac.in/

like byte-level (Xue et al., 2022) and certain character-level approaches (Clark et al., 2022) address vocabulary explosion and sparsity, they significantly increase sequence lengths, leading to scaling issues for training and inference. It is also important to note that—precisely since these data-driven tokenizers are guided by information-theoretic compression principles (Gage, 1994; Sennrich et al., 2016) and are trained over imbalanced data—they learn to represent high-resource languages far more efficiently than low-resource ones. This inherent bias further disadvantages users of underrepresented languages, as exemplified by findings for ChatGPT (Petrov et al., 2023): users in some languages encounter sequence lengths over an order of magnitude longer, leading to more than double the latency, over 2.5× the cost, and reduced service quality from hitting context limits.

We propose visual language representation methods (in § 2, 3) to overcome the vocabulary bottleneck and facilitate more efficient and equitable representation of many non-Latin writing systems.

Multimodality True inclusivity in NLP also requires embracing the *multimodal* nature of human language. Current practice in NLP largely assumes that language is primarily text-based and digitally available. However, out of the world's 293 writing systems,[16] only 168 are currently supported by Unicode and readily available for text-based processing;[17] the remainder, many of which are living minority scripts,[18] are effectively invisible to tokenization-based models. Consequently, speakers of these languages are excluded from the benefits of modern NLP.

Beyond unsupported writing systems, many languages and cultures experience a broader *digitization lag*, where substantial portions of linguistic and cultural knowledge exist primarily in non-digital formats, such as scanned manuscripts, handwritten records, or printed documents (Mager et al., 2018; Ignat et al., 2022). Making this information accessible requires robust optical character recognition (OCR) systems capable of handling a wide variety of scripts, layouts, and degraded image conditions (Agarwal and Anastasopoulos, 2024). Crucially, OCR must be tightly integrated with NLP pipelines in order to not only extract but also *understand* the recovered text—a problem this thesis tackles (§ 4).

Inclusivity also demands attention to languages that are predominantly spoken rather than written. Many languages either lack an established orthography or exhibit significant dialectal variation not captured by standardized text (Mager

[16] https://www.worldswritingsystems.org/
[17] https://www.unicode.org/versions/Unicode16.0.0/
[18] https://linguistics.berkeley.edu/sei/scripts-not-encoded.html

et al., 2018; Aji et al., 2022). Developing NLP technologies for these languages relies on automatic speech recognition (ASR) and text-to-speech (TTS) systems, which in turn require large amounts of transcribed audio data and models robust to acoustic variability (Seamless Communication et al., 2023). Yet, such resources are often scarce, reinforcing the digital marginalization of these communities.

Another severely under-served modality is *sign language*. Over 300 distinct sign languages are used by millions of people in d/Deaf communities worldwide.[19] Sign languages are visual-manual; meaning is conveyed via co-articulated features, including hands (e.g., spatial orientation, position, shape, and movement), body posture, gaze, mouth, and facial expressions (Stokoe, 1980). As such, fully supporting them in NLP requires models capable of processing and generating spatio-temporally structured visual information. On top of these modeling challenges, sign languages suffer from data scarcity, requiring further investment from the research community to collect dataset resources for training, benchmarking, and linguistic analysis (Yin et al., 2021).

It is worth mentioning that vector quantization techniques (van den Oord et al., 2017; Yan et al., 2021) offer ways to discretize images, speech, and videos into token representations, technically making these modalities compatible with token-based language models. These approaches are successfully used in early-fusion multimodal LLMs such as Gemini (Gemini Team et al., 2024) or Chameleon (Chameleon Team, 2025). However, they are also lossy (and linguistically uninformed) transformations that tend to degrade performance on perception and understanding tasks, compared to learning from the raw signal (Du et al., 2024; Qu et al., 2024). In sign language processing, many approaches also rely on gloss annotations (Camgöz et al., 2018; Zhang et al., 2023), a form of transliteration into written labels. However, glosses are incomplete, inaccurate, and costly to annotate (Müller et al., 2023b). Overall, today's tokenization-based language models, by their very design, struggle to adequately capture the full multimodal nature of language.[20]

We propose to unify language representation learning across modalities through *visual representation* (illustrated previously in Figure 1.4), avoiding lossy conversions into tokens while addressing representational bottlenecks. This thesis covers methods for OCR-free processing of non-digital text (in § 4), showing promise to also address the broader digitization lag in the future, and methods for sign language processing (in § 6).

[19] https://www.un.org/en/observances/sign-languages-day

[20] We briefly further discuss opportunities for multimodal LLMs in § 7.2.

Culture and sociodemographics Lastly, inclusivity in NLP goes far beyond language coverage and must take into account *cultural and sociodemographic dimensions*. This includes, for example, teaching models to navigate diverse cultural norms and communication styles (Hershcovich et al., 2022a), and identifying and mitigating inequalities related to gender, race, ethnicity, religion, age, disability, and socioeconomic status (Blodgett et al., 2020; Bender et al., 2021; Karamolegkou et al., 2024; Kirk et al., 2024). While these topics are not the direct focus of this thesis, they remain critical concerns in the broader development of inclusive NLP technologies, and they have become highly active areas of research in recent years (Dev et al., 2023; Soni et al., 2024; Faleńska et al., 2024). These concerns also closely relate to topics of bias and fairness in NLP, which we discuss further in the next section.

Summary Although much progress has been made, achieving genuine inclusivity in NLP will continue to require efforts across data curation, evaluation methodologies, model architecture, representation learning, multimodal integration, and cultural and sociodemographic dimensions.

This thesis makes several contributions towards mitigating these challenges. In § 2 and § 3, we develop methods for visual language representation learning of digital text data to facilitate scaling across languages and writing systems (model architecture, representation learning, multimodal integration). In § 4, we extend this methodology to non-digitized historical text data, making progress towards OCR-free historical document understanding and addressing the broader digitization lag affecting many languages and populations. In § 5, we study MLLMs with a focus on understanding potential synergies and tensions between multilinguality (which directly affects inclusivity) and specific dimensions of trustworthiness. In § 6, we learn visual language representations for American Sign Language, overcoming data scarcity issues to achieve state-of-the-art sign language translation performance. We also curate a new benchmark dataset for sign language translation.

1.2.3 Trustworthiness Challenges in NLP

In recent years, regulatory bodies and organizations, including the EU Commission, OECD, and NIST, have increasingly advocated for the development of *trustworthy AI*. However, defining *trustworthiness* in the context of AI systems remains challenging (Søgaard, 2025), with no clear consensus on its precise

meaning or requirements. We focus on several key desiderata that consistently appear across the various taxonomies proposed (Smuha, 2019; Brundage et al., 2020; Li et al., 2023a; NIST, 2023; Newman, 2023; Makridis et al., 2024; OECD, 2024) and discuss challenges associated with each of them, particularly in the context of NLP and representation learning.

Fairness Fairness demands that systems treat individuals and groups equitably, avoiding the reinforcement or amplification of existing societal inequalities (Mehrabi et al., 2021). This implies that NLP models should perform consistently and provide similar quality of service across different socio-demographic groups (e.g., defined by race, gender identity, age, disability, sexual orientation, religion, nationality, socioeconomic status, or cultural identity) and, particularly relevant in the context of this thesis, across different languages and dialects (Blodgett et al., 2020).

Challenges in fairness largely stem from biases deeply embedded within the data used for training and the design of the models themselves. Large-scale datasets scraped from the web often reflect historical and societal biases, which models readily learn and perpetuate (Bender et al., 2021). These biases manifest in various ways, from stereotypical associations in word embeddings (Bolukbasi et al., 2016) and language models (Nadeem et al., 2021) to disparate performance in downstream applications like hate speech detection or machine translation, where models may perform significantly worse for certain demographic groups or language varieties (Sap et al., 2019). Furthermore, as discussed in § 1.2.2, even fundamental components like tokenizers can introduce inequities, creating less efficient representations for lower-resource languages and disadvantaging their speakers (Petrov et al., 2023).

Researchers have explored various mitigation strategies, often categorized by their intervention stage: pre-processing, in-training, intra-processing, or post-processing (Gallegos et al., 2024). These include curating more balanced datasets or augmenting data (e.g., via counterfactual examples), developing debiasing algorithms that adjust representations or model outputs (e.g., via projection methods, adversarial learning, or contrastive learning), modifying training objectives or using reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022), applying parameter-efficient fine-tuning or modular approaches like adapters (Fatemi et al., 2023; Kumar et al., 2023), modifying decoding strategies (Chung et al., 2023; Kim et al., 2023), and post-hoc rewriting (Dhingra et al., 2023).

While these techniques show promise, they often involve trade-offs with model accuracy or other desiderata (Gonen and Goldberg, 2019). Furthermore, defining and measuring fairness is itself difficult, context-dependent, and subject to ongoing debate, with no single universally applicable definition (Verma and Rubin, 2018). Fully understanding the sources and propagation of bias, especially within LLMs, and developing effective, robust, and universally applicable mitigation strategies that address fairness across diverse linguistic and cultural contexts remain critical open challenges (Blodgett et al., 2020; Gallegos et al., 2024). We focus our efforts related to fairness on performance disparities across languages and writing systems (§ 2, 5).

Privacy Privacy requires the protection of sensitive information related to individuals whose data is used to train or interact with AI/NLP systems (Smuha, 2019). This includes preventing unauthorized disclosure, inference, or re-identification of personal data, aligning with regulatory frameworks like GDPR (European Parliament and Council of the European Union, 2016).

The pretraining paradigm for language models has exacerbated privacy challenges in NLP, as web-scale training data may inadvertently contain personally identifiable information (PII) or other sensitive details (Barrett et al., 2023). Furthermore, LLMs have been shown to *memorize* portions of their training data, sometimes verbatim (Karamolegkou et al., 2023; Biderman et al., 2023). These memorized data can be unintentionally leaked or deliberately extracted through carefully crafted prompts, posing significant risks, especially when models are finetuned on private or proprietary datasets (Carlini et al., 2021). Beyond direct leakage, models are also vulnerable to various inference attacks, including membership inference (determining if specific data was used in training) and attribute inference (inferring sensitive user attributes) (Shokri et al., 2017; Yeom et al., 2018; Mattern et al., 2023). These risks extend to multimodal settings, such as processing sign language videos, which inherently contain sensitive biometric information, as we explore in § 6.

To mitigate privacy risks, *data minimization* and *anonymization* techniques aim to remove or obfuscate sensitive information before training (Senavirathne and Torra, 2020; Lee et al., 2021; Yang et al., 2022c). *Differential privacy (DP)* offers formal guarantees by adding calibrated noise during training or inference, limiting what can be learned about any single data point (Dwork, 2006; Abadi et al., 2016). *Federated learning* allows models to be trained collaboratively on decentralized data without sharing raw user data (Kairouz et al., 2021; Fan et al.,

2023). Other techniques include secure multi-party computation, homomorphic encryption, and newer methods like *machine unlearning* to retroactively remove data influence (Knott et al., 2021; Lee et al., 2022; Xu et al., 2023).

Despite these mitigations, significant challenges remain. A fundamental trade-off often exists between the strength of privacy guarantees and model utility (performance) (Li and Li, 2009), due to which privacy techniques are often still avoided in practice. Moreover, robustly preventing all forms of memorization and leakage in large models is an ongoing struggle (and may not be fully avoidable (Brown et al., 2021)), and effectively auditing privacy remains difficult (Carlini et al., 2021; Panda et al., 2025). Developing privacy-preserving techniques that are scalable, maintain high utility, and address the nuances of language data, therefore, continues to be an active area of research. We study these challenges in the context of differentially-private multilingual text encoders (§ 5) and privacy-aware sign language translation (§ 6).

Robustness and generalization Robustness and generalization require that models maintain reliable and accurate performance even when faced with diverse, unforeseen, or challenging conditions (Hendrycks et al., 2022). This includes generalizing to data drawn from distributions different from the training data (*out-of-distribution, OOD* generalization) (Yang et al., 2023), handling natural variations and noise in inputs—such as the *orthographic variation* common in real-world text or historical documents (Al Sharou et al., 2021)—and resisting deliberate attempts to cause failure (*adversarial robustness*) (Carlini et al., 2019; Goyal et al., 2023).

Models often struggle with OOD generalization (Yang et al., 2023); and while large-scale pretraining of language models provides a foundation and has shown to improve OOD robustness (Hendrycks et al., 2020; Tänzer et al., 2022), generalization to truly novel distributions depends heavily on the breadth of pretraining data and the effectiveness of RLHF post-training (Chu et al., 2025). Models can also be brittle to natural noise and specific perturbations; for instance, standard subword tokenization can make models vulnerable to orthographic noise (Sun et al., 2020). Furthermore, models are susceptible to adversarial examples (Goodfellow et al., 2015)—even subtle typos can degrade performance (Eger et al., 2019; Gan et al., 2024)—and often rely on spurious correlations rather than robust understanding (Geirhos et al., 2020). Robustness challenges are also amplified in multimodal settings, for example, when processing scanned documents where OCR errors can propagate and degrade downstream understanding, or when

translating speech content transcribed through ASR (Zhao and Calapodescu, 2022). We tackle these robustness challenges related to OCR-induced noise in § 4.

Towards improving robustness, adversarial training (Madry et al., 2018; Morris et al., 2020; Ziegler et al., 2022) and data augmentation (Rebuffi et al., 2021; Wei and Zou, 2019) are commonly used. For LLMs, instruction finetuning and RLHF have been critical for improving robustness against certain harmful or adversarial inputs (Kumar et al., 2025), although models are still easily fooled or jailbroken (Zou et al., 2023). Other techniques include robust optimization (Sagawa et al., 2020; Foret et al., 2021) and methods that specifically target OOD such as domain adaptation (Ramponi and Plank, 2020) or OOD detection (Liu et al., 2024). The development of benchmarks to evaluate robustness (Eger and Benz, 2020; Yuan et al., 2023; Calderon et al., 2024) and auditing through red teaming efforts (Dinan et al., 2019; Perez et al., 2022; Ganguli et al., 2022) also indirectly help build more robust models. Robustness to orthographic variation and noise has also been shown to be better in character- or byte-level models than in subword-based ones (Tay et al., 2021; Xue et al., 2022), albeit at the cost of long sequences and degraded performance. We show (in § 2 and 4) that visual language representations can be a compelling alternative to these methods, as first suggested by Salesky et al. (2021). To improve robustness in multimodal systems, recent work increasingly aims to replace cascaded systems (e.g., using OCR or ASR as the first step) by end-to-end approaches (Kim et al., 2022; Seamless Communication et al., 2023; Gemini Team et al., 2024). Our research on OCR-free historical document processing via visual language representations (in § 4) also fits in this line of work. Overall, it is evident that robustness is an active research area with many open questions (Hendrycks et al., 2022).

Explainability and transparency Another crucial aspect of trustworthiness is explainability (or interpretability) and transparency, which refers to the ability to understand how and why an AI model arrives at its outputs (Danilevsky et al., 2020; Søgaard, 2021; Zhao et al., 2024). This ability is vital for debugging, identifying and mitigating biases, ensuring accountability in high-stakes decisions (e.g., in healthcare or finance), building user trust, and gaining insights into the model's learned representations and potential failure modes.

However, due to their large number of parameters, neural networks (especially LLMs) are effectively opaque "black boxes" (Vaassen, 2022; Goetze, 2022; Søgaard, 2023). As such, tracing their outputs, through many layers of computation, back to specific inputs or internal states in a human-understandable way is highly

challenging. In particular, there is often a tension between generating explanations that are *plausible* (convincing to humans) and those that are *faithful* (accurately reflecting the model's actual reasoning); often, optimizing for plausibility can mask the true, potentially flawed, reasoning (Lipton, 2018; Jacovi and Goldberg, 2020; Lyu et al., 2024). Evaluating the quality and faithfulness of explanations itself remains an open problem.

Numerous techniques have been developed to help explain or interpret model behavior. *Post-hoc explanation* methods analyze trained models. Common approaches include *feature attribution*, which assigns importance scores to input features (e.g., words or tokens) using methods like LIME, SHAP, LRP, or gradient-based techniques (Ribeiro et al., 2016; Lundberg and Lee, 2017; Bach et al., 2015; Sundararajan et al., 2017), and visualization of internal states like *attention maps* in transformers (Vig, 2019), although the reliability of attention as explanation is debated (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). As we demonstrate in § 2, such feature attribution methods can offer insights even for models operating on non-standard inputs like pixels (in the context of text understanding). Another category of post-hoc methods focuses on *instance-based explanations*, aiming to understand the impact of specific training data points on model predictions or behavior. Techniques like *influence functions* (Koh and Liang, 2017) estimate the effect of up- or down-weighting individual training examples, helping to identify influential points, potentially problematic data, or sources of specific predictions—an aspect we explore extensively in § 5 in the context of instance-interpretability of multilingual LMs. Most recently, the field of *mechanistic interpretability* seeks to reverse-engineer the specific algorithms and circuits learned within models (Olah et al., 2020; Sharkey et al., 2025), and disentangle concepts stored within the model using methods such as sparse autoencoders (SAEs; Bricken et al., 2023; Huben et al., 2024). Other approaches include building *inherently interpretable models* (often simpler, like linear models or rule lists) or generating natural language explanations (Nori et al., 2019; Zhao et al., 2024).

Overall, however, post-hoc methods can lack faithfulness or stability (Jacovi and Goldberg, 2020), influence functions can be computationally expensive to apply at scale (Grosse et al., 2023), and mechanistic interpretability is still nascent and extremely challenging for LLMs (Sharkey et al., 2025). Bridging these gaps and developing methods that produce scalable, faithful, and useful explanations for the largest models remains an open problem.

Accountability and auditability Accountability deals with mechanisms for assigning responsibility for AI system outcomes and providing redress, while auditability refers to the ability to examine system processes, data, and performance against benchmarks or requirements (Novelli et al., 2024; Li and Goel, 2025). The desideratum is to ensure that AI systems operate within defined ethical and legal boundaries, and that responsibility can be traced when issues arise.

Key challenges include the lack of transparency in current models, as discussed in the previous paragraph, which obscures the models' decision-making pathways (Li and Goel, 2025). The distributed nature of how AI is developed, deployed, and used (via developers, data providers, deployers, users, etc.) can also diffuse responsibility (Novelli et al., 2024). Furthermore, a lack of standardized auditing procedures and documentation formats hinders consistent oversight (Ojewale et al., 2025).

Mechanisms to improve accountability and auditability include rigorous documentation practices like *model cards* (Mitchell et al., 2019) and *datasheets* (Gebru et al., 2021), comprehensive logging of model predictions and data provenance (Longpre et al., 2024), version control for models and datasets (Mökander et al., 2024), and the establishment of clear internal governance structures and external audit frameworks (Bommasani et al., 2024; Longpre et al., 2025). While accountability and auditability are not among the dimensions we study in this thesis, they do represent important dimensions of trustworthy AI, and comprehensively addressing the highlighted challenges remains an ongoing effort in the rapidly evolving landscape around language models.

Safety and security Model safety requires preventing unintended harm resulting from the model, and security aims to maintain integrity against deliberate attacks (Hendrycks et al., 2022; Shah et al., 2025). Challenges with respect to these criteria arise because models can, for example, make harmful *mistakes* due to incomplete knowledge or flawed reasoning (hallucinations) (Huang et al., 2025). They can also be deliberately instructed by users for *misuse*, including for example generating disinformation and aiding cyberattacks (Pan et al., 2023). Models might also become *misaligned*, pursuing goals contrary to developer intent, potentially through deceptive behavior (Greenblatt et al., 2024). Security vulnerabilities, such as insecure access controls or successful jailbreaking attempts, can exacerbate misuse and misalignment risks (Zou et al., 2023). Mitigating these diverse risks requires a layered defense strategy.

Techniques improving model *robustness* (discussed previously) contribute to

security by hardening models against certain adversarial inputs (Ziegler et al., 2022). This is something we demonstrate in the case of orthographic text attacks (in § 2), where, as a consequence of improved robustness, the model may avoid making harmful misclassifications. *Alignment techniques*, including safety-focused training and RLHF, aim to prevent models from generating harmful content and refuse unsafe requests (Ouyang et al., 2022; Bai et al., 2022). Proactive identification of dangerous capabilities and testing via *red teaming* help uncover vulnerabilities (Ganguli et al., 2022). System-level security measures like strict access controls, monitoring for irregular behavior, and guardrails for input and output filtering provide further layers of protection against both misuse and potential harm from misaligned models (Inan et al., 2023; Rebedea et al., 2023). Furthermore, (mechanistic) *interpretability* methods, as also discussed earlier, can aid in understanding and potentially detecting unsafe or misaligned model internals (Bereska and Gavves, 2024). Overall, as model capabilities continue to rapidly advance, safety and security practice also needs to continue to improve accordingly (Shah et al., 2025).

The space in between A major challenge beyond these individual criteria lies in the inter-dependencies and trade-offs between them (Ovalle et al., 2024). In particular, enhancing one dimension often negatively impacts another; however, we need to uphold all of the above principles to achieve *trustworthy AI*. For example, strong privacy guarantees or robustness tend to reduce model accuracy (Li and Li, 2009; Raghunathan et al., 2020). And while security occasionally benefits from robustness (not necessarily vice-versa) (Ziegler et al., 2022), privacy and group fairness can in many cases be at odds (Cummings et al., 2019; Hansen et al., 2024) (although there is also counter-evidence (Matzken et al., 2023; de Oliveira et al., 2024)), and empirical fairness and explainability are orthogonal in certain NLP tasks (Brandl et al., 2024). Likewise, maximizing fairness according to one definition might conflict with another (Binns, 2020). These examples of the not yet fully understood relationships between trustworthiness criteria are *pairwise* interactions; further, Ruder et al. (2022); Cresswell (2025) find that interactions between *more than two* dimensions are even less frequently studied, leaving the joint space largely unexplored. Further understanding these interactions, quantifying the trade-offs, and developing methods that co-optimize multiple objectives is a critical but still nascent area of research, which we explore in § 5 of this thesis in the context of multilingual (token-based) language models.

Summary In sum, trustworthiness encompasses a wide spectrum of desiderata, each presenting unique technical and ethical challenges. As highlighted in this section, broad research efforts are underway to address these dimensions individually, and the (often conflicting) interactions between these goals are just beginning to be explored more deeply. Bridging the gap between these high-level principles and practical implementations remains an ongoing challenge. This thesis makes contributions towards addressing specific facets of these challenges: exploring alternative visual language representations to directly mitigate robustness issues related to orthographic noise and linguistic variation (§ 2) and indirectly improve robustness by avoiding OCR-induced noise in document processing (§ 4); investigating the interplay between instantiations of privacy, linguistic fairness, transparency, and performance in multilingual language models (§ 5), and developing a privacy-aware methodology sign language processing at scale (§ 6). While these represent targeted steps within a vast research area, they contribute to the development of more trustworthy NLP technologies.

1.3 Scientific Contributions

The overarching scientific goal of this thesis is to advance the development of more inclusive and trustworthy language processing systems. Our contributions lie in addressing specific challenges towards this goal, many of which trace back to limitations of how current NLP models represent text (§ 1.2.2; 1.2.3). We explore visual language representations as an alternative to token-based representations, aiming to overcome these limitations, and we also study the implications of visual and multilingual language representations more broadly for AI trustworthiness desiderata. We now detail the contributions of the individual publications included in this thesis, first through the lens of inclusivity, and then through the lens of trustworthiness.

1.3.1 Addressing Inclusivity Challenges Through Visual Language Representations

- Chapter 2 (Rust et al., 2023) proposes and investigates the learning of visual language representations for processing of written language in the form of digital text. We introduce the pixel-based encoder of language (PIXEL), a vision transformer (ViT; Dosovitskiy et al., 2021) operating directly on images of text obtained through a controlled rendering process,

and pretrain PIXEL on the English Wikipedia and BookCorpus via masked autoencoding (He et al., 2022). We then finetune and evaluate PIXEL on a wide range of syntactic and semantic natural language processing tasks over a typologically diverse set of languages and writing systems, demonstrating how the visual representations learned by PIXEL can handle a wider range of linguistic diversity, including previously unseen writing systems, low-resource languages, and mixed-language text (code-switching), without having to predefine a finite vocabulary of discrete input units (tokens)—effectively overcoming the *vocabulary bottleneck* of token-based models. More broadly, this paper lays the foundation for visual text processing as an alternative paradigm to models relying on subword tokenization.

- Chapter 3 (Lotz et al., 2023) further explores these visual text representations, comparing the continuous text rendering strategy from Rust et al. (2023) with slightly less flexible but more structured rendering strategies. We find that rendering text as bigrams, i.e. pairs of consecutive characters, dramatically reduces the complexity of the model's image patch space, resulting in more efficient PIXEL models that consistently perform better at sentence-level tasks, making them an even more viable alternative to token-based encoders for both high- and low-resource languages. Structured rendering also enables scaling down the model from 86M to 22M parameters (the latter can be pretrained with ¼ the compute cost) without losing performance over the original (continuous-rendering) PIXEL model on benchmarks like GLUE (Wang et al., 2018) and Universal Dependencies (Zeman et al., 2022; Nivre et al., 2020). These leaner models are much more accessible to people with limited computational resources—another crucial consideration for inclusivity (Khanuja et al., 2023).

- Chapter 4 (Borenstein et al., 2023b) studies the effectiveness of visual language representations for non-digitized text, in particular in the context of historical documents. To this end, we pretrain a PIXEL model on scans of real historical documents and synthetic scans where text is rendered as images in a way that closely resembles the layout and style of historical documents. Through finetuning and evaluation on natural language understanding and question answering tasks, we find that this fully pixel-based approach is a viable alternative to cascaded approaches relying on optical character recognition (OCR), followed by token-based text processing. Our results show that visual language encoders can effectively deal with the

high levels of noise typically present in historical documents, showing promise for an OCR-free future in (historical) document processing. These results are also encouraging beyond the domain of historical documents: visual language representations could analogously enable processing for those written languages that are not yet supported by Unicode or for which digitization efforts are still lagging behind.

- Chapter 6 (Rust et al., 2024) shows how visual language representations naturally extend to sign languages, which are a natively visual form of communication and historically under-served by the NLP community (Yin et al., 2021). We propose a privacy-aware two-stage framework for sign language translation at scale and develop a state-of-the-art method for sign language translation from American Sign Language (ASL) to English, outperforming the prior state-of-the-art by over 3 BLEU points (Papineni et al., 2002) in the zero-shot and finetuned settings—while addressing privacy risks. Our method, termed SSVP-SLT, leverages self-supervised pretraining on unannotated videos (i.e., learning of visual representations of sign language) to overcome the critical scarcity of labeled sign language data. We also ablate various components of our method, helping the research community better understand how to effectively use self-supervised pretraining for sign language video (given that our work is the first to perform larger-scale pretraining in this domain). We also release a hand-curated ASL benchmark dataset, Dailymoth-70h, comprising ~75 hours of video with time-aligned English translations, to facilitate the controlled evaluation of ASL models. Overall, this work presents a step towards closing the large performance gap between neural machine translation (NMT) systems for spoken and sign languages, contributing to more inclusive multimodal NLP and translation technology.

1.3.2 Addressing Trustworthiness Challenges Through Visual and Multilingual Language Representations

- Chapter 2 (Rust et al., 2023) goes beyond standard downstream performance when comparing visual and token-based text encoders, studying their *robustness* to orthographic noise. We perform a series of adversarial attacks in which texts are orthographically perturbed across different levels of degradation during finetuning, finding that PIXEL models offer superior robustness to these types of noise than their token-based counterparts. Given

that orthographic noise can vary strongly across data domains, these results yield insights into how reliable the tested models may be when deployed in the real world. We also employ a feature attribution explainability method (Chefer et al., 2021) to learn what parts of the image are relevant to PIXEL when classifying samples, qualitatively confirming the *plausability* of the model's predictions, even in the presence of strong noise.

- Chapter 4 (Borenstein et al., 2023b) investigates the feasibility of visual language representations in eliminating OCR-induced noise as a way to strengthen the *robustness* of historical document processing systems. To this end, we introduce two visual versions of the popular GLUE benchmark (Wang et al., 2018)—one rendered with random font augmentation and degradations commonly found in historical documents and one without such interventions—and compare our historical PIXEL model against both OCR-free and OCR-based (cascaded) baselines on these datasets. We find that, while the cascaded approaches are still preferable in terms of overall performance, the OCR-free approaches, including our model, are substantially more robust to the introduced noise and degradations, supporting the idea that visual language representations can help build more robust models that can be relied upon in real-world settings.

- Chapter 5 (Rust and Søgaard, 2023) aims to address the research gap resulting from the fact that most research in trustworthy NLP/AI focuses on individual trustworthiness objectives in isolation, rather than considering the inter-dependencies between them. We, therefore, directly study the interactions between common instantiations of three trustworthiness criteria (differential *privacy*, linguistic *fairness*, instance-*interpretability*) in the context of multilingual language models. In particular, we aim to learn in which ways *multilinguality* can facilitate or hinder trustworthiness along these criteria. This question is complementary to our work on visual language representations, which naturally lend themselves to multilingual settings by overcoming the vocabulary bottleneck. We first show, theoretically, that differential privacy and instance-interpretability (the ability to post-hoc identify influential training examples) are fundamentally at odds, so we can at best Pareto-optimize for them. We also show that multilinguality can help jointly satisfy differential privacy and linguistic fairness, although difficult to achieve in practice. We then extensively finetune, evaluate, and analyze the representations of XLM-R (Conneau et al., 2020a) and mBERT (Devlin

et al., 2019), two popular massively multilingual encoder LMs at the time, across two common NLP tasks and a typologically diverse set of languages to further explore these trade-offs empirically. We also introduce the influence uniformity (InfU) measure, which quantifies instance-interpretability in a multilingual setting. Our results elucidate the three-way and four-way interactions between the different objectives, showing that we can optimize for certain combinations of objectives but cannot simultaneously achieve strong performance, privacy, multilinguality, and instance-interpretability. Our improved understanding of these interactions can help inform the development of alternative methods that achieve practical trade-offs and push the Pareto frontier towards more overall trustworthy language models.

- Chapter 6 (Rust et al., 2024) explores the *privacy-utility trade-off* in sign language translation more practically. In particular, a central goal of this work is to make sign language processing more scalable through self-supervised video pretraining. However, we argue that increased scale also bears privacy risks due to biometric information present in sign language data, necessitating a framework that considers both scalability and privacy preservation. Given the lack of advanced anonymization tools for sign language data, we adopt facial blurring as a practical privacy-preserving method, despite its known limitations in obfuscating linguistic cues conveyed through facial expressions. To mitigate the resulting loss in utility, our framework learns visual sign language representations from *anonymized* videos during pretraining and allows for optional de-anonymization during finetuning (with signer consent). We show that this approach achieves state-of-the-art translation performance by a substantial margin, even when anonymity is preserved throughout the large-scale pretraining phase. Through careful ablation, we then isolate the effects of anonymization and demonstrate that while best performance is achieved with full visual access, the performance degradation due to anonymization can largely be recovered during finetuning. To facilitate future research on privacy-aware sign language processing, we also release two versions of our previously mentioned DailyMoth-70h dataset—one with anonymized and one with deanonymized videos—allowing for controlled comparisons. Overall, this chapter contributes to the development of privacy-aware sign language processing systems and demonstrates that it is possible to design systems that are both high-performing *and* aligned with privacy considerations around biometric data.

Chapter 2

Language Modelling with Pixels

The work presented in this chapter is based on a paper that has been published as: **Phillip Rust**, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda.

Abstract

Language models are defined over a finite set of inputs, which creates a *vocabulary bottleneck* when we attempt to scale the number of supported languages. Tackling this bottleneck results in a trade-off between what can be represented in the embedding matrix and computational issues in the output layer. This paper introduces PIXEL, the **Pix**el-based **E**ncoder of **L**anguage, which suffers from neither of these issues. PIXEL is a pretrained language model that renders text as images, making it possible to transfer representations across languages based on orthographic similarity or the co-activation of pixels. PIXEL is trained to reconstruct the pixels of masked patches instead of predicting a distribution over tokens. We pretrain the 86M parameter PIXEL model on the same English data as BERT and evaluate on syntactic and semantic tasks in typologically diverse languages, including various non-Latin scripts. We find that PIXEL substantially outperforms BERT on syntactic and semantic processing tasks on scripts that are not found in the pretraining data, but PIXEL is slightly weaker than BERT when working with Latin scripts. Furthermore, we find that PIXEL is more robust than BERT to orthographic attacks and linguistic code-switching, further confirming the benefits of modelling language with pixels.

 xplip/pixel  Team-PIXEL

2.1 Introduction

Natural language processing has rapidly progressed in recent years due to a combination of self-supervised representation learning, i.e. pretrained language models (PLMs) like BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), and XLM-R (Conneau et al., 2020a); large unlabelled datasets; such as C4 (Raffel et al., 2020), The Pile (Gao et al., 2020); and large-scale computing power (Hirschberg and Manning, 2015). Despite this progress, these models only cover a fraction of the world's languages, with large inequalities in performance (Pires et al., 2019; Lauscher et al., 2020), and the majority of languages are falling behind English (Joshi et al., 2020b; Bugliarello et al., 2022). Even within English, these models struggle when tasked with processing noisy inputs (Sun et al., 2020; Eger and Benz, 2020). In this paper, we show how to effectively support *thousands* of written languages in a single model while being robust to variations caused by character-level noise.

Language models typically support a finite vocabulary of categorical inputs, e.g. characters, subwords or even words, and much effort has been devoted to vocabulary construction (Wan, 2022). On one end of the spectrum, a vocabulary over words has three problems: (i) it is not possible to encode out-of-vocabulary words because they lack an entry in a closed vocabulary, e.g. "doxing", (ii) there are too many parameters in the word embedding layer, and relatedly, (iii) the normalising constant for the softmax activation in the output layer is too expensive to compute. On the other end of the spectrum, vocabularies over bytes or characters are much smaller, which leads to increased sequence lengths (Keren et al., 2022). In practice, most current models operate over inputs smaller than words but larger than characters: subword units (Sennrich et al., 2016; Kudo, 2018). Subwords prevent the problem of extremely large embedding and output layers, and support open vocabulary processing. While this is a practical solution in a monolingual context and for some languages like English, dealing with many languages with a variety of scripts will either result in a very large vocabulary or a trade-off over what is represented within a fixed number of subwords (see § 2.5). Taken together, given a language model with a finite vocabulary, there is a bottleneck in two locations: at the level of the encoding of the inputs and at the level of estimating the probability distribution over the vocabulary. We call this the *vocabulary bottleneck*. A language model that can handle thousands of languages needs to deal with this problem.

We propose to rethink language modelling as a visual recognition task, remov-

ing the need for a finite vocabulary. Our proposal is inspired by Salesky et al. (2021), who showed how to train a machine translation model with "visual text representations" in the encoder instead of subwords. Our **Pix**el-based **E**ncoder of **L**anguage (**PIXEL**) is built on the Masked Autoencoding Visual Transformer (ViT-MAE; He et al., 2022). ViT-MAE is a Transformer-based encoder-decoder trained to reconstruct the pixels in masked image patches. PIXEL does not have a vocabulary embedding layer; instead, text is rendered as a sequence of fixed-sized patches, which are processed using a Vision Transformer encoder (Dosovitskiy et al., 2021). PIXEL also does not have an expensive output layer when it reconstructs the pixels of the masked patches. In effect, PIXEL provides a solution to the vocabulary bottleneck without needing the prohibitively long sequences of character-based models.

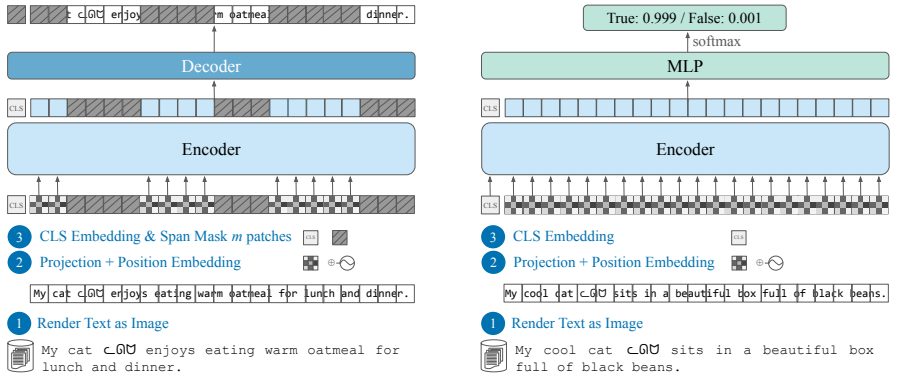
PIXEL is pretrained on the same data as BERT, given our computational resources. This means that it has encountered only ~0.05% non-English text (Blevins and Zettlemoyer, 2022).[1] We evaluate PIXEL on a range of syntactic and semantic tasks in 32 typologically diverse languages across 14 scripts, showing that it can rapidly adapt to new languages and unseen scripts. PIXEL is also evaluated on its ability to handle noisy text caused by orthographic attacks, where pixel-based encoding is a clear improvement over subword-based vocabularies. In lexical code-switching experiments, PIXEL performs on-par with BERT and sometimes outperforms the multilingually pretrained MBERT.

PIXEL is a new type of language model that can theoretically support any language that can be typeset by a modern computer. We make the implementation, the pretrained model including intermediate training checkpoints, and the finetuned models freely available for the community.

2.2 Approach

The Pixel-based Encoder of Language, PIXEL, consists of three major components: a text renderer, which draws text as an image; an encoder, which encodes the unmasked regions of the image; and a decoder, which reconstructs the masked regions at the pixel level. Figure 2.1 provides an illustration.

[1] We do not claim that a language model designed to support thousands of languages should be pretrained only on English text. We expect that pretraining on an appropriate choice of another language or multilingually may provide more remarkable results. PIXEL represents an initial effort at smaller scale.



(a) PIXEL pretraining (b) PIXEL finetuning

Figure 2.1: Overview of PIXEL's architecture. Following He et al. (2022), we use a masked autoencoder with a ViT architecture and a lightweight decoder for pretraining (left). At finetuning time (right), the decoder is replaced by a task-specific classification head that sits on top of the encoder.

2.2.1 Text Renderer

The key component of PIXEL is a text renderer that takes one or more pieces of text and renders them onto a blank RGB image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$. We set height $H = 16$ and width $W = 8464$ and choose $C = 3$ RGB input channels, which is equivalent to a square colour image with a 368×368 resolution and corresponds to a sequence of 529 image patches of size 16×16 pixels.[2] Figure 2.2 shows examples of text inputs rendered by the text renderer. The renderer supports (a) colour emoji and hieroglyphs scripts, (b) left-to-right and right-to-left writing systems, and (c) text that requires ligatures. Analogous to BERT, a sequence can either contain a single paragraph of text or a text pair; we use black 16×16 patches to serve as separators and end-of-sequence (EOS) markers. Blank (white) patches after the end-of-sequence marker are treated as padding by PIXEL, where no attention scores or losses are computed. Sequences longer than the maximum length are either truncated or split into multiple sequences. Further technical details about the renderer are provided in Appendix 2.7.4.

[2]We chose a sequence length of 529 so that the memory requirements at maximum length are approx. equal to those of BERT. Forward and backward passes of the transformer layers at equal length are also equally fast.

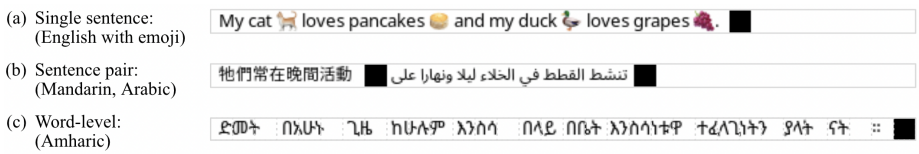


Figure 2.2: Illustrative examples of our rendered text. PIXEL natively supports most writing systems, colour emoji (a), and complex text layouts such as right-to-left writing and ligatures (b). Black patches serve as separators and end-of-sequence markers. Blank patches to the right of the end-of-sequence marker are treated as sequence padding. For word-level tasks, horizontal spacing can be added between words (c) so that every patch can be assigned to exactly one word (dotted lines indicate patch boundaries for demonstration).

2.2.2 Architecture

PIXEL-base is a 112M parameter ViT-MAE architecture (He et al., 2022) with a 12-layer ViT encoder (Dosovitskiy et al., 2021) and an 8-layer Transformer decoder (Vaswani et al., 2017). The encoder has 86M parameters and the decoder has 26M parameters, respectively. The 8-layer decoder is not used for downstream tasks. We give an overview of the architecture below, with more details in Appendix 2.7.5. We did not train larger PIXEL variants for lack of computational resources.

Patch Embeddings The images produced by the text renderer (§ 2.2.1) are patch-wise linearly projected to obtain a sequence of patch embeddings with a 16 × 16 pixel resolution, to which fixed sinusoidal position embeddings are added.[3]

Patch Span Masking Instead of the random masking procedure used in ViT-MAE or block-wise masking in BEiT (Bao et al., 2022), PIXEL uses span masking with a 25% masking ratio as outlined in Algorithm 1, which masks spans of up to $S = 6$ consecutive image patches with a dynamic number of unmasked patches left between them. The idea behind the span masking approach, inspired by T5 (Raffel et al., 2020) and SpanBERT (Joshi et al., 2020a), is that it masks more meaningful units of text (full words or phrases) than random masking where the model more often has to fill in (parts of) individual characters, thereby

[3]This is a fast operation that does not require the large text embedding layer found in subword-based models, saving parameters which could in theory be re-allocated to the self-attention stack. We refer to Xue et al. (2022) for a discussion regarding benefits and drawbacks of re-allocation of embedding layer weights.

Algorithm 1 PIXEL Span Masking

Input: #Image patches N, masking ratio R, maximum masked span length S, span length cumulative weights $W = \{w_1, \ldots, w_S\}$
Output: Masked patches $\mathcal{M}$
$\mathcal{M} \leftarrow \emptyset$
repeat
  $s \leftarrow$ randchoice($\{1, \ldots, S\}, W$) ▷ $S = 6$, $\mathbb{E}(s) = 3.1$
  $l \leftarrow$ randint$(0, \max(0, N - s))$
  $r \leftarrow l + s$
  if $\mathcal{M} \cap \{l - s, \ldots, l - 1\} = \emptyset$ **and** $\mathcal{M} \cap \{r + 1, \ldots, r + s\} = \emptyset$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{l, \ldots, r\}$
  end if
until $|\mathcal{M}| > R \cdot N$ ▷ $R = 0.25$
return $\mathcal{M}$

encouraging PIXEL to model a higher level of abstraction. In practice, span masking was slightly more effective than random masking in early prototypes of PIXEL. This effect may be less noticeable at higher masking ratios (such as the 75% used in ViT-MAE), when random masking would more often mask consecutive patches. We found a 25% masking ratio to work well for PIXEL-base, which is in line with recent findings for BERT-type models of similar size (Wettig et al., 2023). We mask spans of $s \in \{1, 2, 3, 4\}$ patches in length, each with 20% probability, and spans of $s \in \{5, 6\}$ patches with 10% probability each, so $\mathbb{E}(s) = 3.1$.

Encoder Following ViT-MAE (He et al., 2022), the PIXEL encoder only processes unmasked patches (i.e., ≈ 396 "visible" patches at 25% masking) rather than on a sequence including mask tokens, which not only reduces memory requirements and increases training speed, but also has the advantage of not creating a mismatch between pretraining and finetuning. This mismatch would occur when training the encoder with inserted mask tokens because they are not inserted during finetuning (He et al., 2022). We also prepend the special `CLS` embedding to the unmasked patches.[4] The resulting `CLS` and unmasked patches are processed by a 12-layer Transformer encoder to produce a sequence of encoder output representations.

Decoder The PIXEL decoder first projects the encoder outputs into the same space as the decoder model's hidden size. It then inserts learnable mask embed-

[4] In pretraining, no loss is computed for the `CLS` embedding, but it can be used for finetuning.

dings at the masked positions; these are what PIXEL tries to reconstruct at the pixel level. Fixed sinusoidal position embeddings (Vaswani et al., 2017) are added to inject order information. After processing this sequence via 8 Transformer layers, a linear projection yields patch logits. Note that the decoder does not have to compute an expensive softmax over a subword vocabulary and circumvents the question of whether to tie the subword embedding weights. PIXEL is trained with a normalised mean squared error (MSE) pixel reconstruction loss, measuring the discrepancy between normalised target image patches and reconstructed patches. This loss is only computed for *masked, non-blank (text)* patches.

2.2.3 Pretraining

PIXEL-base is pretrained on a rendered version of the English Wikipedia and the Bookcorpus (Zhu et al., 2015), which is roughly equivalent to the BERT pretraining data.[5] For better compute efficiency, we concatenate paragraphs until the maximum sequence length is reached, albeit not across document and book boundaries. Wikipedia has 2B words rendered into 11.4M examples and the Bookcorpus has 1.1B words rendered into 5.4M examples; in total ~3.1B words (BERT used 3.3B) rendered into 16.8M examples.[6] PIXEL is pretrained for 1M steps with batch size 256 (i.e. ~16 epochs) using the AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with a linear warmup over the first 50k steps to a peak learning rate of $1.5e{-}4$ and a cosine decay to a minimum learning rate of $1e{-}5$. Pretraining took 8 days on 8×40GB Nvidia A100 GPUs. We show the loss curve and additional pretraining details in Appendix 2.7.5. We stored PIXEL checkpoints every 10k steps and make them available alongside the fully trained model on the HuggingFace Hub (Wolf et al., 2020), which we hope will be useful to analyze training dynamics of PIXEL models (Sellam et al., 2022). Figure 2.5 in Appendix 2.7.2 shows, for three unseen examples, how PIXEL learns to model language over the course of pretraining.

2.2.4 Finetuning

PIXEL can be finetuned for downstream NLP tasks in a similar fashion to BERT-like encoders by simply replacing the PIXEL decoder with a suitable classification head. By truncating or interpolating the sinusoidal position embeddings, we can

[5] We use a similar Wikipedia dump Devlin et al. (2019) used for BERT (February 1, 2018) and a slightly newer version of the Bookcorpus available at 🤗 `datasets/bookcorpusopen`.

[6] This rendering is quite compact; see Appendix 2.7.4.

finetune with sequences shorter or longer than 529 patches, respectively. The latter, in particular, is common in computer vision applications to finetune on higher resolution images (Touvron et al., 2019; Kolesnikov et al., 2020; Dosovitskiy et al., 2021; He et al., 2022). For most common NLP tasks, we can typically finetune with sequences shorter than 529 to accelerate training while retaining performance. To demonstrate that PIXEL supports a variety of downstream tasks, we conduct finetuning experiments in four settings as follows:

Word Classification For word-level tasks like part-of-speech (POS) tagging and named entity recognition (NER), we render each word at the start of a new image patch so that we can create a bijective mapping between words and patches (see Figure 2.2 for an example).[7] To finetune PIXEL on these images, we add a linear classifier with dropout. We assign the label of a word only to its first corresponding image patch and compute a cross-entropy loss with softmax.

Dependency Parsing For dependency parsing, we render text as above but obtain word-level representations by mean pooling over all corresponding image patches of a word and employ a biaffine parsing head (Dozat and Manning, 2017), following the implementation from Glavaš and Vulić (2021).

Sequence Classification For sequence-level tasks, e.g. in GLUE (Wang et al., 2018), we render text as in pretraining. For sentence-pair tasks like natural language inference (NLI) we separate the sentences with a black patch. We finetune with different strategies, including training a classifier on top of (1) the CLS embedding, (2) the mean-pooled or max-pooled representations of all patches, (3) a multi-head attention block. Although we did not notice significant performance differences between them in our experiments, we mainly used option (1), which is exactly the same as in BERT, and (2), which has been shown to work well for image classification (Liang et al., 2022).

Extractive Question Answering (QA) For extractive QA datasets like SQuAD (Rajpurkar et al., 2016), we render the question and context like in sequence-pair tasks above and, same as Devlin et al. (2019), use a sliding window approach to extract answers for examples exceeding the maximum sequence length. We use a

[7] This particular formulation assumes that word boundaries are available. We note that subword-based and character-based models also make this assumption. For further discussion on the implications, see Appendix 2.7.6.

linear classifier to predict the start and end patches of the span containing the answer. Appendix 2.7.4 explains how we obtain the mapping between characters and rendered text.

2.3 Experiments

We finetune PIXEL on common NLP tasks and evaluate its syntactic and semantic processing capabilities in English, as well as its adaptability to unseen languages. Table 2.8 (Appendix 2.7.6) describes the languages used in these experiments, and our language and data selection is also motivated below.

2.3.1 Tasks and Languages

Syntactic Tasks We evaluate PIXEL on part-of-speech (POS) tagging and dependency parsing using data from Universal Dependencies v2.10 treebanks (Nivre et al., 2020; Zeman et al., 2022) for a set of typologically diverse languages that captures a large variety of unseen scripts[8]: Arabic (ARA), Coptic (COP), English (ENG), Hindi (HIN), Japanese (JPN), Korean (KOR), Tamil (TAM), Vietnamese (VIE), Chinese (ZHO).[9] We compare how well PIXEL transfers to these languages compared to BERT. Note that BERT does not support all of these writing systems. However, both models have been trained on the same data. This comparison allows us to gauge the extent to which PIXEL can overcome the script barrier and vocabulary bottleneck of subword-based models.

Semantic Tasks We evaluate both monolingual (ENG) and cross-lingual *word-level* understanding on MasakhaNER (Adelani et al., 2021), a named entity recognition (NER) benchmark for 10 African languages (AMH, HAU, IBO, KIN, LUG, LUO, PCM, SWA, WOL, YOR), which also includes a copy of the ConLL-2003 dataset (ENG; Tjong Kim Sang and De Meulder, 2003). For monolingual ENG *sentence-level* understanding, we rely on GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016). Finally, we evaluate cross-lingual sentence-level understanding on TyDiQA-GoldP (Clark et al., 2020) in the *in-language multitask* setting where we train on the combined gold data in all 9 target languages (ARA, BEN, ENG, FIN, IND, KOR, RUS, SWA, TEL) at once, and on two additional larger monolingual extractive question answering (QA) corpora: KorQuAD 1.0

[8] By unseen, we mean not present in the pretraining data.
[9] Table 2.10 in Appendix 2.7.6 gives an overview of the treebanks we use.

(KOR; Lim et al., 2019) and JaQuAD (JPN; So et al., 2022).

2.3.2 Baselines and Finetuning protocols

We compare results to BERT-base, which is trained on the same data.[10] We do not compare to newer monolingual English models like RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2020), or DeBERTa (He et al., 2021b,a) because these models have been pretrained longer on much larger corpora.[11] Likewise, we do not compare against models trained on massively multilingual corpora. However, to contextualise the performance of PIXEL in cross-lingual settings, we report results for MBERT and, if results are available, for CANINE (Clark et al., 2022). For BERT, we use the standard finetuning protocols used by Devlin et al. (2019) and the same biaffine classifier for parsing as for PIXEL. We list finetuning details for all tasks in Appendix 2.7.6.

2.3.3 Results

Syntactic Tasks We present results for POS tagging and dependency parsing in Table 2.1. While BERT is slightly better than PIXEL in the monolingual setting (ENG), PIXEL clearly outperforms BERT in the remaining languages. On the lower end, the accuracy gap in favor of PIXEL in ARA and VIE, both languages covered by BERT's vocabulary, is relatively small (~1%). On the higher end, in COP, where BERT has an out-of-vocabulary ([UNK]) token ratio of 93%, the gap is ~70% for both tasks. There is a strong correlation[12] between the proportion of [UNK]s (shown in Table 2.1 on the right) and the performance gap, which shows that PIXEL overcomes BERT's vocabulary bottleneck. These results are further analysed in Appendix 2.7.9.

Semantic Tasks We present results for NER in Table 2.2, for GLUE in Table 2.3, for QA in Table 2.4. We also conduct experiments on XNLI in the *translate-train-all* setting, which we present in Table 2.16 in Appendix 2.7.9, for brevity. We find that BERT consistently achieves higher performance than PIXEL in its pretraining language ENG. Likewise, it often outperforms on languages using the Latin

[10] We use BERT weights from 🤗 `bert-base-cased`.

[11] We do not intend to claim state-of-the-art performance, but to demonstrate that PIXEL can overcome the vocabulary bottleneck and to provide a starting point for further research on pixel-based encoding of language.

[12] Pearson correlation $r = 0.9$, $p < 0.001$ for POS tagging, $r = 0.95$, $p < 0.0001$ for dependency parsing.

	$\lvert\theta\rvert$	ENG	ARA	COP	HIN	JPN	KOR	TAM	VIE	ZHO
POS Tagging (Accuracy)										
BERT	110M	**97.2**	95.4	26.5	86.4	87.9	60.0	45.4	84.5	58.6
PIXEL	86M	96.7	**95.7**	**96.0**	**96.3**	**97.2**	**94.2**	**81.0**	**85.7**	**92.8**
Dependency Parsing (LAS)										
BERT	110M	**90.6**	**77.7**	13.0	75.9	73.8	30.2	15.2	49.4	28.8
PIXEL	86M	88.7	77.3	**83.5**	**89.2**	**90.7**	**78.5**	**52.6**	**50.5**	**73.7**

	[UNK]%	Fertility
ENG	0	1.2
ARA	1.8	3.7
COP	93.6	1.0
HIN	32.6	2.7
JPN	45.5	1.5
KOR	84.7	1.0
TAM	82.3	1.3
VIE	4.5	2.5
ZHO	73.2	1.5

Table 2.1: Results for PIXEL and BERT finetuned for POS tagging and dependency parsing on various Universal Dependencies treebanks. We report test set results averaged over 5 runs each. $|\theta|$ denotes the number of model parameters. The table on the right shows BERT's proportion of [UNK]s as a measure of (inverse) vocabulary coverage and fertility (i.e., number of subwords per tokenized word; Ács, 2019; Rust et al., 2021) as a measure of over-segmentation in respective UD treebanks.

	#L	$\lvert\theta\rvert$	ENG	AMH	HAU	IBO	KIN	LUG	LUO	PCM	SWA	WOL	YOR
MBERT*	104	179M	92.2	0	87.3	85.3	72.6	79.3	73.5	86.4	87.5	62.2	80.0
CANINE-C + ng*	104	167M	89.8	50.0	88.0	85.0	72.8	79.6	74.2	88.7	83.7	66.5	79.1
CANINE-C*	104	127M	79.8	44.6	76.1	75.6	58.3	69.4	63.4	66.6	72.7	60.7	67.9
BERT	1	110M	**92.9**	0	**86.6**	**83.5**	**72.0**	**78.4**	**73.2**	**87.0**	**83.3**	**62.2**	**73.8**
PIXEL	1	86M	89.5	**47.7**	82.4	79.9	64.2	76.5	66.6	78.7	79.8	59.7	70.7

Table 2.2: Results for PIXEL and BERT finetuned for NER on MasakhaNER. We report test set F_1 scores averaged over 5 runs each. BERT outperforms PIXEL in all of the languages that use Latin script, whereas PIXEL does better on AMH, whose script is not covered by BERT's vocabulary. The performance gap is smaller for languages heavier in diacritics, e.g. YOR. It is larger for languages closer to English such as Naija Pidgin (PCM), an English-based creole. #L denotes the number of pretraining languages, +ng denotes CANINE's n-gram extension, and * indicates results taken from Clark et al. (2022) for additional context.

writing system; for instance, in NER where all languages besides AMH use Latin script, in QA for FIN, IND, and SWA. Although BERT has more trainable parameters, this finding indicates that a PIXEL model pretrained for the same number of steps as BERT is slightly worse at semantic tasks, and it may require longer pretraining or an additional inductive bias to close the performance gap. Similarly, character-based models also tend to underperform subword-based models on NER (Keren et al., 2022), here seen by the CANINE-C results. Since

the addition of n-gram embeddings improves the performance of CANINE-C, likely due to boosting entity memorisation capabilities (Clark et al., 2022), we hypothesize that PIXEL may benefit from equivalent enhancements.

For languages where BERT only partially covers the script, such as KOR, JPN and TEL in QA, PIXEL consistently outperforms BERT, sometimes by large amounts (e.g. , +63 F_1 points better on KorQuAD). In the extreme case where BERT has no coverage of the script whatsoever, seen in NER for AMH, BERT fails completely (0 F_1) while PIXEL outperforms the larger, multilingually trained CANINE and performs competitively with its n-gram variant. In other words, PIXEL also overcomes the vocabulary bottleneck of subword-based PLMs in semantics-driven tasks. Note that although BERT was trained on English, its vocabulary has a high coverage of the Arabic script, explaining its good performance in ARA and URD.[13]

While the same may apply to languages like BEN and RUS in QA, where one may otherwise expect PIXEL to outperform BERT, there is an external factor at play; in the standard QA task formulation used by BERT, answer spans are extracted by predicting start and end tokens. We adopt this procedure in PIXEL for simplicity. However, an image patch will often overlap two words at variable positions, so the answer may actually start or end mid-patch. By only predicting on a full-patch level, and extracting the entire content of the patch, PIXEL will sometimes extract leading and trailing characters that should not be part of the answer, which degrades the F_1 score—even though the model may have correctly identified the span. Languages not using whitespace to delimit words are particularly affected, which also explains why PIXEL is only slightly better than BERT in JPN.

Generally, and in particular when transferring to unseen scripts, we find that PIXEL performs best when finetuning on larger corpora. An example of this behaviour can be seen in QA, where PIXEL performs significantly better on KorQuAD (60k examples) than the KOR subset of TyDi (1.6k examples). While large corpora may often not be available when dealing with unseen scripts, we hypothesize that multilingual pretraining will alleviate the need for long finetuning, while potentially being even more conducive to *positive transfer* (Conneau et al., 2020a; Chau et al., 2020; Pfeiffer et al., 2021) by not being vocabulary-bottlenecked.

[13]Arabic is lexically sparse (Antoun et al., 2020; Al-Sallab et al., 2017), so the characters can be covered in the vocabulary. However, it is morphologically complex, which leads to over-segmentation, as the fertility of 3.7 in Table 2.1 shows. This over-segmentation is not necessarily problematic in our selection of tasks (Keren et al., 2022), e.g. due to the sliding window in QA, but can be a disadvantage

	$\lvert\theta\rvert$	MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	WNLI	Avg
		393k	364k	105k	67k	8.6k	5.8k	3.7k	2.5k	635	
BERT	110M	**84.0 / 84.2**	**87.6**	**91.0**	**92.6**	**60.3**	**88.8**	**90.2**	**69.5**	51.8	**80.0**
PIXEL	86M	78.1 / 78.9	84.5	87.8	89.6	38.4	81.1	88.2	60.5	**53.8**	74.1

Table 2.3: Results for PIXEL and BERT finetuned on GLUE. We report *validation* set performance averaged over 5 runs. The metrics are F_1 score for QQP and MRPC, Matthew's correlation for CoLA, Spearman's ρ for STS-B, and accuracy for the remaining datasets. PIXEL achieves non-trivial performance scores on GLUE, indicating *pixel-based encoders can learn higher-level semantic tasks*, but performs worse overall than BERT, so it may require (a) more pretraining steps than subword-tokenized PLMs or (b) additional inductive bias to acquire the same level of monolingual abstraction.

	#L	$\lvert\theta\rvert$	TyDiQA-GoldP										SQuAD	KorQuAD	JaQuAD
			ENG	ARA	BEN	FIN	IND	KOR	RUS	SWA	TEL	Avg	ENG	KOR	JPN
MBERT	104	179M	75.6	78.1	74.7	75.5	84.3	64.8	74.9	83.1	81.6	77.1	88.6	90.0	76.4
BERT	1	110M	**68.5**	**58.0**	**43.2**	**58.3**	**67.1**	12.4	**53.2**	**71.3**	48.2	51.5	**88.2**	14.9	28.8
PIXEL	1	86M	59.6	57.3	36.3	57.1	63.6	**26.1**	50.5	65.9	**61.7**	**52.3**	81.4	**78.0**	**34.1**

Table 2.4: Results for PIXEL and BERT finetuned on extractive QA datasets. We report validation set F_1 scores averaged over 5 runs each. Average (Avg) scores for TyDiQA-GoldP exclude ENG as customary (Clark et al., 2020). While BERT clearly outperforms PIXEL in ENG, PIXEL is much better in KOR, TEL, and JPN—a consequence of the vocabulary bottleneck in BERT—thereby gaining an edge on average. In some languages, answer span extraction adversely affects results (see § 2.3.3).

2.4 Robustness to Orthographic Attacks and Code-Switching

Informal text, commonly found on social media, often contains orthographic noise such as typos and other variations (Baldwin et al., 2015; van Esch et al., 2019; Caswell et al., 2020). Previous work has demonstrated the vulnerability of pretrained language models to character-level adversarial attacks and noise (Sun et al., 2020; Eger and Benz, 2020), with text normalization typically required to maintain performance (Pruthi et al., 2019; Keller et al., 2021). To evaluate PIXEL's robustness to textual noise and variation, and inspired by the robustness tests of Salesky et al. (2021), we experiment with the *Zeroé* benchmark (Eger and

in others (Rust et al., 2021).

	POS Tagging		Named Entity Recognition		
	SPA-ENG	HIN-ENG	SPA-ENG	HIN-ENG	MSA-EA
MBERT	97.1	86.3	64.0	72.6	65.4
BERT	**96.9**	87.0	**61.1**	**74.5**	59.4
PIXEL	96.8	**88.2**	61.0	73.0	**63.7**

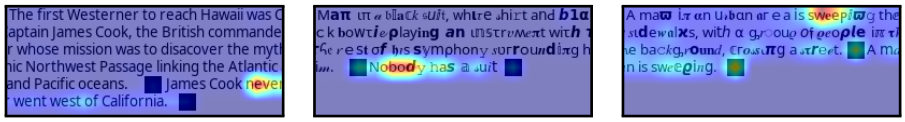
Table 2.5: Code-switching results on LinCE.

Benz, 2020; Keller et al., 2021), which covers a variety of low-level orthographic attacks as illustrated in Table 2.13. We replace their version of visual attacks with the Unicode Technical Standard #39 set of visually-confusable characters.[14] We apply *Zeroé* attacks during finetuning and evaluation of two English downstream tasks, POS tagging and NLI (Bowman et al., 2015), where we expect models to rely on different levels of abstraction.

Figure 2.8 and 2.9 in Appendix 2.7.7 compare PIXEL and BERT across three levels of token-level noise for POS tagging and NLI. There is little impact on POS tagging performance with either model from most low-level attacks, with the exception of visually-confusable character substitutions (CONFUSABLE); here PIXEL expectedly maintains performance above 92% as it generalizes across orthographic similarities, but BERT drops to 38%. For NLI, both models are negatively affected, but PIXEL exhibits less degradation than BERT with higher proportions of noise, with the impact varying across the types of attacks, which each affect subword tokenization differently. Figure 2.3 shows relevancy heatmaps (Chefer et al., 2021) for SNLI predictions made with and without CONFUSABLE substitutions. The heatmaps are similarly clear with and without noise, providing qualitative evidence that PIXEL is indeed robust to the noise. The illustrated robustness may be dependent upon finetuning, however; we find that PIXEL can struggle in zero-shot applications when text is rendered differently from observed during pretraining (see Appendix 2.7.4 on using different fonts). Future work could explore the impact of data augmentation during pretraining on PIXEL's robustness and ability to transfer across scripts. Furthermore, it would be interesting to investigate how the choice of font influences the search space during reconstruction of masked patches (Bland et al., 2022).

In addition to robustness to orthographic noise, dealing with character-level substitutions is important for effectively modelling different morphological forms. There are also many types of higher-level token, phrase or sequence-level

[14] https://util.unicode.org/UnicodeJsps/confusables.jsp



(a) 0%, *contradiction* (b) 80%, *contradiction* (c) 80%, *entailment*

Figure 2.3: Visual explanations of correct PIXEL predictions (for classes *contradiction* and *entailment*) for NLI examples with 0% and 80% CONFUSABLE substitutions using method by Chefer et al. (2021), providing qualitative evidence for PIXEL's robustness to character-level noise and the interpretability of its predictions. Red heatmap regions represent high relevancy.

variations such as code-switching—when a speaker alternates between two or more languages in the same utterance, while being grammatically consistent in each language (Joshi, 1982)—or the lexical substitutions in social media text. We evaluate PIXEL on the LinCE benchmark (Aguilar et al., 2020), which includes core tasks and downstream applications for linguistic code-switching. PIXEL is finetuned on POS Tagging and NER in Spanish-English, Hindi-English and Modern Standard Arabic-Egyptian Arabic. Table 2.5 shows that PIXEL and BERT perform similarly on SPA-ENG tasks, with BERT outperforming PIXEL on NER for (romanised) HIN-ENG. On the other tasks, PIXEL performs better than BERT and even outperforms MBERT on HIN-ENG POS tagging. The gap between MBERT and PIXEL is larger on Arabic scripts, which were extensively seen by MBERT during pretraining.

2.5 Related Work

The question of vocabulary construction is an open problem in NLP, especially in a multilingual context.[15] The most widely used language models, e.g. BERT, RoBERTa, T5, GPT-2 *inter alia*, rely on different tokenizers, such as WordPiece (Devlin et al., 2019), Byte-Pair Encoding (BPE; Sennrich et al., 2016) and Unigram LM (Kudo, 2018). There is an established ecosystem around subword tokenizers, such as the SentencePiece (Kudo and Richardson, 2018) and HuggingFace Tokenizers.

In a monolingual context and for some languages like English, vocabularies of subwords are a good tradeoff between vocabularies of characters and vocabularies

[15] See Mielke et al. (2021) for a recent, comprehensive survey on open-vocabulary modeling and tokenization.

of words. When representing a large number of languages in multilingual PLMs like mBERT and XLM-R, adequately representing the vocabulary of each individual language would be computationally prohibitive. The tokenization then becomes a bottleneck when trying to scale up to a large number of languages (Conneau et al., 2020a; Rust et al., 2021), which manifests itself in degraded cross-lingual performance to languages and language families that are underrepresented in the data used for training multilingual PLMs. There are large inequalities in the performance of these models across typologically diverse languages (Wu and Dredze, 2020; Lauscher et al., 2020). This issue is further exacerbated by tokenizations out-of-the-box not being compatible across languages (Maronikolakis et al., 2021). Language imbalance and poor character coverage in the vocabulary can also decrease downstream performance (Zhang et al., 2022). To some extent, these problems can be attenuated through techniques such as subword mapping (Vernikos and Popescu-Belis, 2021), transliteration (Moosa et al., 2023), leveraging lexical overlap (Patil et al., 2022), vocabulary clustering and reallocation (Chung et al., 2020), continued or language-adaptive pretraining (Ebrahimi and Kann, 2021), adaptation via bilingual lexica (Wang et al., 2022), and embedding matrix adaptation (Artetxe et al., 2020a). However, these are post-hoc workarounds to expand model vocabularies after training. They do not provide a direct solution to the vocabulary bottleneck problem.

Some subword-based algorithms can also produce undesirable segmentations for morphologically rich languages (Klein and Tsarfaty, 2020; Amrhein and Sennrich, 2021), so dedicated morphologically-aware tokenizers have been developed (e.g. Smit et al. (2014)), but this process often requires expert-level knowledge and may only work for individual languages.

Due to the limitations of subword vocabularies in multilingual language modelling, some works have used vocabularies over characters (Lee et al., 2017; Ma et al., 2020, *inter alia*) or bytes (Wang et al., 2020a; Wei et al., 2021). These provide benefits over purely subword-based models in terms of robustness and most of them are readily applicable in a multilingual context,[16] but they typically come at the cost of increased sequence lengths or latency. Also, such models cannot exploit orthographic similarities between characters across and within scripts and do not account for the fact that meaning of language may be carried visually, such as in writing systems that are (partially) logographic, like Chinese, in ancient hieroglyphs, or when using emoji.

[16]Character-aware models are not directly applicable to languages that do not use whitespace to delimit sentences (Tay et al., 2021), for example.

Finally, some works have developed pixel-based approaches. Broscheit (2018) embedded images of Chinese glyphs but still relied on a fixed vocabulary. Wu et al. (2019) combined character-level images and embeddings for a variety of Chinese tasks. Radford et al. (2021a) trained a linear probe for CLIP, which also incorporates a tokenizer, on a rendered version of SST-2 (Socher et al., 2013). Other works have trained pixel-based models that removed the need for a fixed vocabulary: Sun et al. (2019) trained a convolutional sentiment classifier on pixels. Mansimov et al. (2020) used images of text for in-image MT. Salesky et al. (2021) employed a convolutional embedder for a Transformer-based MT system with a subword-based decoder. Our method differs from these in that it provides a general-purpose language encoder that completely removes the need for a vocabulary.

2.6 Conclusion

This paper introduced PIXEL, a pretrained language model that renders text as images, which allows it to represent any written language that can be typeset using its text renderer. PIXEL was pretrained on the predominantly English Wikipedia and Bookcorpus datasets, and evaluated on part-of-speech tagging, dependency parsing, question answering, and language understanding tasks. The results demonstrate that PIXEL readily transfers to unseen scripts, as shown by its performance on 14 scripts across 32 languages. PIXEL currently lags behind BERT when processing languages with a Latin script, including English; however, PIXEL is more robust than BERT against low-level orthographic attacks and performs competitively to BERT and MBERT on linguistic code-switching tasks. Overall, these results show that pixel-based representations are a strong backbone for cross-lingual and cross-script transfer learning. The limitations of this work are discussed in Appendix 2.7.10.

In future work, we will investigate inductive biases and additional objectives that can better capture long-range dependencies in PIXEL models. We hope that this will help overcome the limits of PIXEL in semantic processing. We also plan to pretrain PIXEL on multilingual text with a view to further improving its cross-script and cross-lingual abilities. This will also allow us to more fairly compare pixel-based models against larger subword-based and tokenization-free *multilingual* models. Finally, we will also develop new rendering and finetuning formulations that are better tailored to pixel-based models, e.g. for improving downstream question answering.

Acknowledgments

We thank Ákos Kádár, Barbara Plank, and Kris Cao for their comments on an earlier draft. We also thank Davide Rigoni, Rita Ramos, Stella Frank, and members of the CoAStaL and LAMP groups for discussions. Miryam de Lhoneux is funded by the Swedish Research Council (grant 2020-00437). Phillip Rust is funded by the Novo Nordisk Foundation (grant NNF 20SA0066568). Jonas F. Lotz is funded by the ROCKWOOL Foundation (grant 1242). Emanuele Bugliarello is supported by funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199. Elizabeth Salesky is supported by the Apple Scholars in AI/ML fellowship. Desmond Elliott is partially supported by the Innovation Foundation (grant 0176-00013B) and the Novo Nordisk Foundation (grant NNF 20SA0066568). This work was supported by a research grant (VIL53122) from VILLUM FONDEN. The computing power was generously supported by EuroHPC grants 2010PA5869, 2021D02-068, and 2021D05-141, and with Cloud TPUs from Google's TPU Research Cloud (TRC).

2.7 Appendix

2.7.1 Abstract Reconstructions

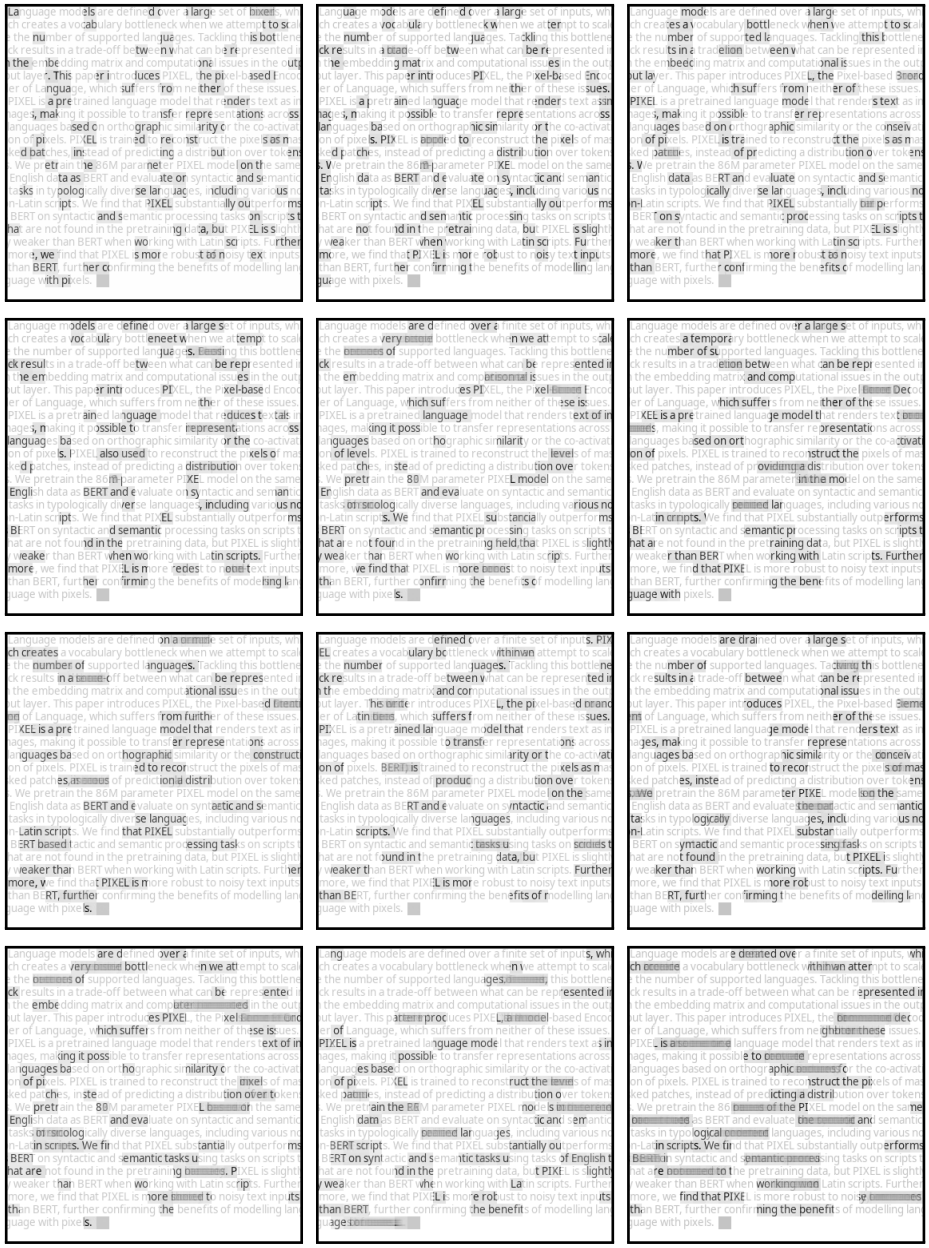
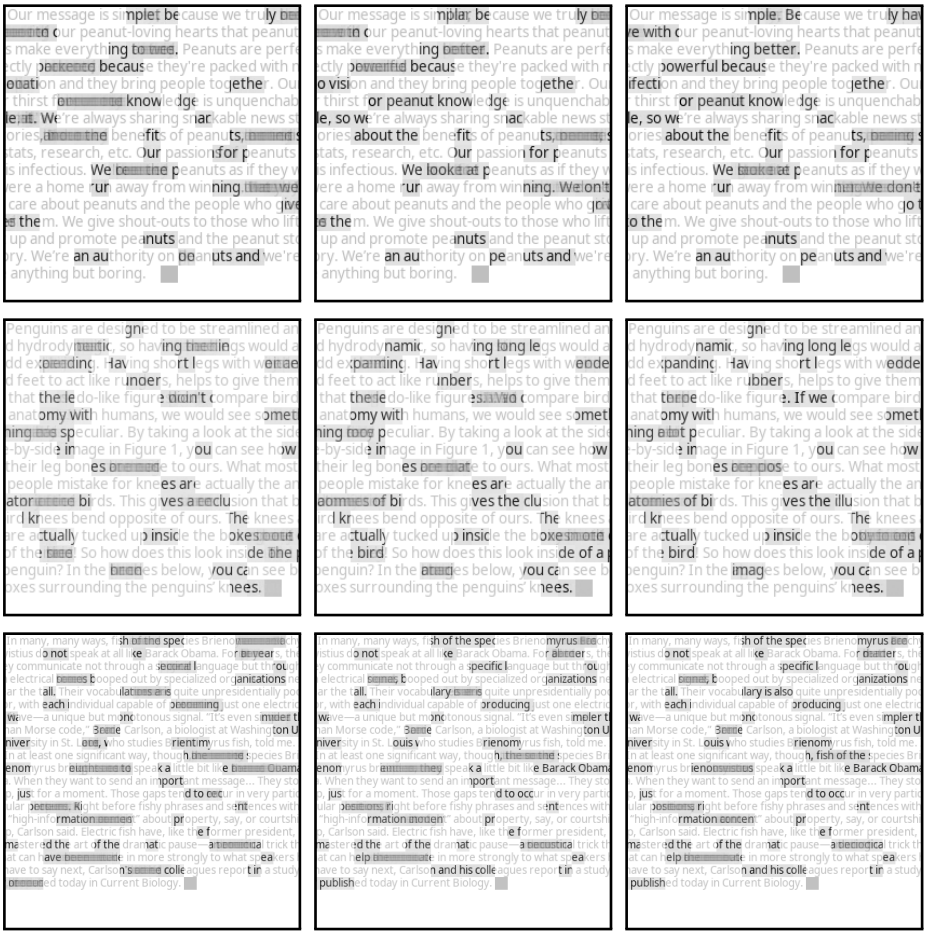


Figure 2.4: PIXEL image reconstructions of the abstract with different span masks.

2.7.2 Web Text Reconstructions



(a) 100k steps (b) 500k steps (c) 1M steps

Figure 2.5: PIXEL image reconstructions after 100k, 500k, and 1M steps of pretraining. We overlay the masked original image with the model's predictions. Images are wrapped into squares and resized for visualization purposes only. The texts were not part of the training data. We see that the fully trained PIXEL (1M) predicts masked spans more clearly and accurately. For longer spans with a larger possible prediction space, multiple predictions may appear together creating blurred text.

Reconstructions of three sources of text[17] [18] [19] after 100K, 500K and 1M pretraining steps. The figure also shows how PIXEL (visually) expresses uncertainty, e.g. for reconstructions of long spans where the space of possible outputs is much larger than for short spans, and how it captures long-range dependencies. In the third row, we can for instance see that PIXEL uses context from the beginning of a sequence (*Barack Obama*) to correctly fill in a gap later in the sequence, and vice-versa (*Brienomyrus*).

2.7.3 Code

PIXEL is implemented in PyTorch (Paszke et al., 2019) and built on HuggingFace transformers (Wolf et al., 2020). We make our code available at `xplip/pixel`. Our pretrained PIXEL model, including a large number of intermediate checkpoints, is available at `Team-PIXEL/pixel-base` and our finetuned models, including multiple seeds each, are available through the model hub.

2.7.4 Text Renderer Details

Rendering backend We experimented with different text rendering backends. Following Salesky et al. (2021), our first implementation was based on PyGame,[20] which PIXEL was also pretrained with. Later on, we switched to a backend based on Pango (Taylor, 2004) and Cairographics,[21] which has native support for complex text layouts, making it possible to specify fallback fonts, and has faster rendering speed. Without fallback fonts, we would be limited to a maximum number of $2^{16} - 1$ glyphs that can fit into a single OpenType or TrueType font file due to a technical limitation.[22] By leveraging fallback fonts, we can theoretically cover all Unicode codepoints, including emojis.

Fonts We rely on the Google Noto Sans fonts collection,[23] which covers the majority of Unicode codepoints and is actively growing.[24]. Note, however, that PIXEL is compatible with any font and can therefore encode anything that can be

[17] https://www.nationalpeanutboard.org/peanut-info/our-message.htm
[18] https://www.penguinsinternational.org/2019/07/10/do-penguins-have-knees-and-other-frequently-asked-questions/
[19] https://www.theatlantic.com/science/archive/2021/05/electric-fish-pause/618993/
[20] https://www.pygame.org/
[21] https://www.cairographics.org/
[22] See https://en.wikipedia.org/wiki/Unicode_font for an explanation.
[23] https://fonts.google.com/noto
[24] See https://notofonts.github.io/overview/ for an overview of Noto's Unicode coverage.

typeset on a computer screen. We used a font size of 8 at 120 DPI for pretraining with PyGame, which was selected manually to fit most scripts into a rendered height of 16px. It can, however, also be adjusted at finetuning time. For finetuning with PangoCairo, we use a font size of $8 \cdot (120/72) \approx 13.33$ which yields roughly the same outputs as the PyGame renderer. Due to how glyphs are shaped by the two backends, the outputs of the two renderers do not *exactly* match. Because we did not employ data augmentation to make PIXEL robust to such changes in font size, we recommend using the PyGame renderer it was pretrained with for *zero-shot* applications with PIXEL. When finetuning, this minor mismatch in rendering outputs is easily overcome by PIXEL, so we generally recommend using the PangoCairo renderer.

Characters versus glyphs For extractive QA, it is necessary to obtain a mapping between the characters in the context paragraph and where they appear on the rendered image. Obtaining this mapping is not straightforward due to how text is rendered. The *shaping* step in the rendering pipeline converts characters into glyphs.[25] In ligatures, as common for instance in Arabic, a glyph is composed of multiple characters. Likewise, an emoji often consists of a base codepoint and a modifier codepoint (e.g. to change the emoji skin colour) which are represented by a single glyph. For accents, on the other hand, one character might yield multiple glyphs.[26] In practice, the renderer therefore uses grapheme clusters, whose logical boundaries in the rendered image we can map to the input characters.[27] For simplicity, we assign each codepoint of a grapheme cluster to the logical horizontal offset at which the cluster starts on the rendered image. Future work may investigate alternative mapping strategies.

RGB rendering PIXEL supports RGB rendering which may be useful to accurately represent colour emoji and for multimodal applications in the future. However, 24-bit RGB rendering is slightly slower than 8-bit grayscale rendering (see Table 2.6 below) for text written in Latin script, which is why we made RGB rendering an optional setting. In our pretraining and finetuning experiments we rendered text in grayscale, and we generally recommend doing so when not working with coloured inputs.

[25] See https://docs.gtk.org/Pango/pango_rendering.html for an overview of the rendering pipeline.

[26] https://docs.gtk.org/Pango/pango_fonts.html#glyphs

[27] https://unicode.org/reports/tr29/#Grapheme_Cluster_Boundaries

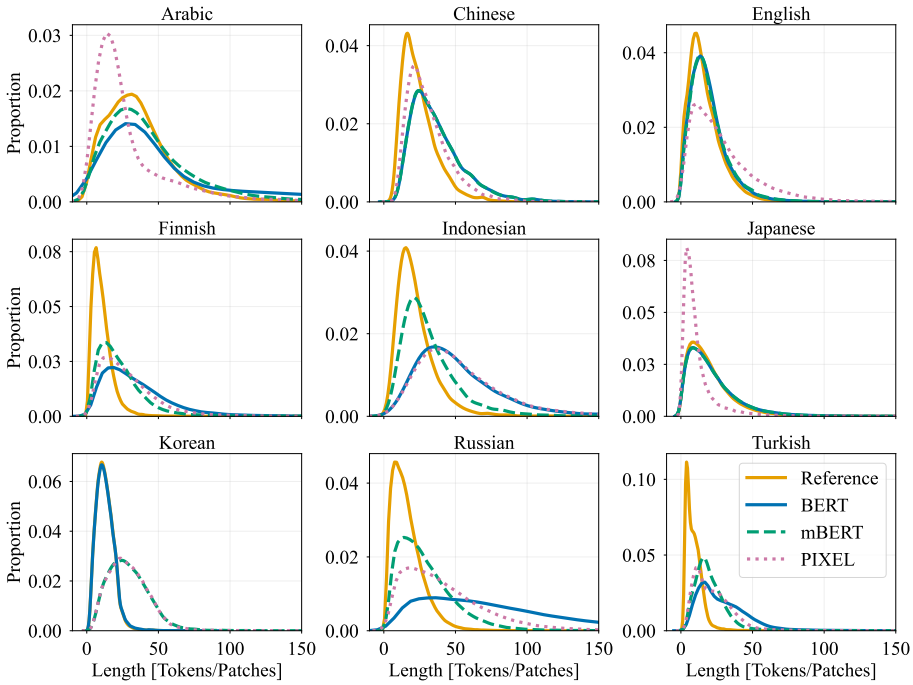


Figure 2.6: Distributions of sentence lengths from monolingual UD corpora after tokenizing by BERT and MBERT and rendering by PIXEL, compared to the reference by UD treebank annotators.

Right-to-left scripts PIXEL's renderer natively supports right-to-left (RTL) writing. In the default setting, the base text direction (which for instance determines on which side of a sentence punctuation marks are placed) is inferred automatically by the rendering backend based on the first "strong directional" character in a given paragraph.[28] The mirroring of RTL characters is also handled automatically according to their Unicode bidi attributes. Optionally, the base text direction can be set manually, which is useful when working on monolingual data, e.g. in Arabic or Hebrew, as the renderer does not have to go through the direction check. In § 2.7.10, we describe limitations of how we currently handle RTL writing.

[28] See https://unicode.org/reports/tr9/ for an overview of the Unicode bidi algorithm.

Processor	Batched	Throughput [ex / s]	
		ENG	ZHO
Renderer (Grayscale)	✗	3944.1	6309.0
Renderer (RGB)	✗	3615.1	6849.5
Tokenizer (Rust)	✓	**19128.9**	**18550.5**
	✗	4782.9	5684.4
Tokenizer (Python)	✓	1286.6	2637.1
	✗	1286.8	2580.9

Table 2.6: Throughput comparison between PIXEL's PangoCairo renderer and the fast and slow BERT tokenizers, implemented in Rust and Python respectively, from the HuggingFace tokenizers library. We estimate throughput, measured in examples per second, by how long it takes to process 1M lines of English (ENG) and Chinese (ZHO) Wikipedia text on the same desktop workstation (AMD Ryzen 9 3900X 12-core CPU). We distinguish between tokenizing all lines individually (Batched = ✗) and as one single batch (✓).

Efficiency analysis We briefly analyze the text processing (rendering versus tokenizing) efficiency in terms of a) length of the processed sequence, which has a direct effect on GPU memory consumption and the time it takes to compute forward and backward passes, and b) processing throughput.

For a), we follow Rust et al. (2021) and process the training and validation splits of all available UD v2.10 treebanks in various languages with the PIXEL renderer and the tokenizers of BERT and MBERT. We plot the resulting sentence length distributions in Figure 2.6, including a comparison with the reference segmentations from the UD annotators. For English text, the PIXEL renderer is slightly less efficient, i.e., it produces slightly longer sequences on average than the tokenizers. For other languages with Latin script, e.g. Finnish and Turkish, the renderer is more efficient than the BERT tokenizer, albeit slightly less efficient than the MBERT tokenizer. For non-Latin scripts such as Arabic and Japanese, we see that the renderer can be a lot more efficient than both tokenizers. The English BERT tokenizer is technically fairly space-efficient for non-Latin scripts but this is misleading because it largely produces [UNK]s (recall right side of Table 2.1) and each [UNK] is a single token; the functionality of the BERT model on a sequence of [UNK] is strongly compromised.

For b), we compare the processing throughput of HuggingFace's BERT tokenizers and our PIXEL renderer in Table 2.6. We find that the Rust-based BERT tokenizer with batch processing achieves the highest throughput by leveraging

parallelization. When not using batch processing, it is comparable in throughput with PIXEL's renderer, i.e. depending on the language or script, rendering can be slightly slower (ENG) or faster (ZHO) than tokenizing. Since the rendering backend (PangoCairo) is implemented in C, we expect to achieve similar gains in rendering throughput by also leveraging parallelization for batch processing (in contrast to the Python-based tokenizer which is limited by Python's global interpreter lock (GIL)). We plan to implement batch rendering functionality in the future.

2.7.5 Architecture & Pretraining Details

Parameter	Value
Image size	(16, 8464, 3)
Patch size P	16
Encoder hidden size D_{enc}	768
Encoder intermediate size	3072
Encoder num attention heads	12
Encoder num layers L	12
Decoder hidden size D_{dec}	512
Decoder intermediate size	2048
Decoder num attention heads	16
Decoder num layers K	8
Layer norm ε (Ba et al., 2016)	$1e{-}12$
Span masking ratio R	0.25
Span masking max length S	6
Span masking cumulative weights W	$\{0.2, 0.4, 0.6, 0.8, 0.9, 1\}$
Span masking spacing	Dynamic
Dropout probability	0.1
Hidden activation	GeLU (Hendrycks and Gimpel, 2016)
Optimizer	AdamW (Loshchilov and Hutter, 2019; Kingma and Ba, 2015)
Adam β	(0.9, 0.999)
Adam ε	$1e{-}8$
Weight decay	0.05
Peak learning rate	$1.5e{-}4$
Learning rate schedule	Cosine Decay (Loshchilov and Hutter, 2017)
Minimum learning rate	$1e{-}5$
Learning rate warmup ratio	0.05
Training steps	1M
Batch size	256

Table 2.7: PIXEL pretraining settings

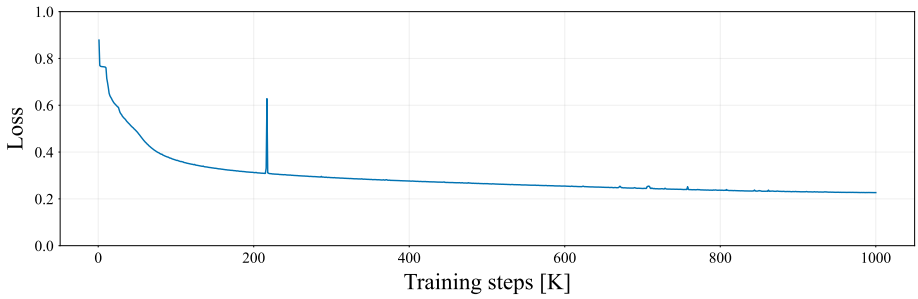


Figure 2.7: PIXEL pretraining loss curve

Patch Embeddings PIXEL reshapes each image $\mathbf{x}$ into a sequence of $N = W/P$ non-overlapping flattened 2D patches $\mathbf{x}_f \in \mathbb{R}^{N \times (P^2 C)}$, where $P = 16$ is the patch size, and linearly projects them via $\mathbf{E} \in \mathbb{R}^{(P^2 C) \times D_{\text{enc}}}$ to obtain patch embeddings $\mathbf{x}_p = (\mathbf{x}_f \mathbf{E}) \in \mathbb{R}^{N \times D_{\text{enc}}}$ with encoder hidden size $D_{\text{enc}} = P^2 C = 768$.[29] Afterwards, fixed sinusoidal position embeddings $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D_{\text{enc}}}$ are added, leaving out the position vector in position 0 for a classification (CLS) embedding later: $\tilde{\mathbf{x}}_p = \mathbf{x}_p + [\mathbf{E}^1_{\text{pos}}, \ldots, \mathbf{E}^{(N+1)}_{\text{pos}}]$.

Span Masking PIXEL then masks out $R = 25\%$ of the $N = 529$ embedded patches via span masking with max span length $S = 6$ and cumulative span weights $W = \{0.2, 0.4, 0.6, 0.8, 0.9, 1\}$, i.e. $\mathbb{E}(s) = 3.1$, as outlined in Algorithm 1. Applying the mask $\mathcal{M}$, we obtain the unmasked patches $\tilde{\mathbf{x}}_{\text{vis}} = \{\tilde{\mathbf{x}}^i_p : i \notin \mathcal{M}\}^N_{i=0}$.

Encoder Following ViT-MAE (He et al., 2022), the PIXEL encoder only operates on unmasked patches (i.e., ≈ 396 patches at 25% masking) and a special CLS embedding with its positional encoding $\mathbf{c} = \mathbf{x}_{[\text{cls}]} + \mathbf{E}^0_{\text{pos}} \in \mathbb{R}^{1 \times D_{\text{enc}}}$ is prepended to the sequence: $\mathbf{h}_0 = [\mathbf{c}, \tilde{\mathbf{x}}_{\text{vis}}] \in \mathbb{R}^{(1+\lfloor R \cdot N \rfloor) \times D_{\text{enc}}}$.[30] Let $\{\mathbf{h}_i\}^L_{i=1}$ be the encoder hidden states after each of the $L = 12$ encoder transformer layers, and $\mathbf{h}_0$ denotes the input sequence. The outputs of each transformer layer are computed as detailed in (Vaswani et al., 2017), [31] and the last layer's output $\mathbf{h}_L \in \mathbb{R}^{(1+\lfloor R \cdot N \rfloor) \times D_{\text{enc}}}$ is passed to the decoder.

[29] This is equivalent to projecting each rendered image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ via a 2D-convolutional layer with C input channels and D_{enc} output channels and kernel size and stride both equal to the patch size P, which we do in practice.

[30] In pretraining, no loss is computed for the CLS embedding but it can optionally be used when finetuning PIXEL for sequence-level downstream tasks.

[31] Note that encoder and decoder do not attend to the blank (padding) patches that appear after the EOS patch.

Decoder The PIXEL decoder first projects the encoder outputs via $\mathbf{E}_{\text{dec}} \in \mathbb{R}^{D_{\text{enc}} \times D_{\text{dec}}}$ to obtain decoder embeddings $\mathbf{x}_d = \mathbf{h}_L \mathbf{E}_{\text{dec}} \in \mathbb{R}^{(1+\lfloor R \cdot N \rfloor) \times D_{\text{dec}}}$, where $D_{\text{dec}} = 512$. Next, mask embeddings $\mathbf{x}_{[\text{mask}]} \in \mathbb{R}^{1 \times D_{\text{dec}}}$ are inserted at the masked-out positions and fixed sinusoidal position embeddings are added to obtain $\mathbf{d}_0 = [(\mathbf{x}_\mathbf{d} \cup \{\mathbf{x}_{[\text{mask}]} : i \in \mathcal{M}\}_{i=0}^{N}) + \mathbf{E}_{\text{pos}}] \in \mathbb{R}^{(N+1) \times D_{\text{dec}}}$. $\{\mathbf{d}_i\}_{i=1}^{K}$ are the decoder hidden states after each of the $K = 8$ decoder transformer layers, computed in the same way as the encoder hidden states, and $\mathbf{d}_0$ denotes the input sequence. There is no encoder-decoder cross-attention. The decoder output $\mathbf{d}_K \in \mathbb{R}^{(N+1) \times D_{\text{dec}}}$ is projected via $\mathbf{O} \in \mathbb{R}^{D_{\text{dec}} \times (P^2 C)}$ to obtain patch-wise logits $\mathbf{o} = (\mathbf{d}_K \mathbf{O}) \in \mathbb{R}^{(N+1) \times (P^2 C)}$. Finally, the CLS logits are removed and a normalized mean squared error (MSE) pixel reconstruction loss is computed: $\mathcal{L}_{\text{normpix}} = \frac{1}{|Q|} \sum_{i \in Q} |\text{normalize}(\mathbf{x}_f^i) - \mathbf{o}^i|^2$ with i denoting the indices in the set of *masked, non-blank (text)* patches $Q = \{i : i \in (\mathcal{M} \cap \mathcal{T})\}_{i=0}^{N}$ and normalize$(\cdot)$ dividing the difference between the target patch and its mean by its standard deviation.

2.7.6 Finetuning Details

Table 2.8 gives an overview of all languages used in our finetuning experiments, Table 2.9 links to our finetuning datasets, and Table 2.10 lists the UD treebanks we used.

We list our finetuning recipes in Table 2.11 for POS tagging, dependency parsing, NER, QA, and XNLI and in Table 2.12 for the GLUE tasks. Due to compute limitations we did not run comprehensive hyperparameter sweeps. Instead, we relied on sensible priors from finetuning BERT and made slight modifications as needed. In most cases, hyperparameters that work well for BERT also work well for PIXEL. For some of the semantic tasks, in particular NLI and SST-2, we found that some random initializations did not converge. In those cases, minor tweaks to the learning rate or increasing the batch size usually helped. For GLUE, we found that PIXEL performed slightly better on some tasks with the PangoCairo renderer, whereas for others, using the PyGame renderer (which PIXEL was pretrained with) was more stable. We plan to further optimize the training recipes and study PIXEL's convergence behaviour in the future.

For word-level tasks, we add padding in order to render each word at the start of a new image patch and so create a bijective mapping between words and patches. Doing so assumes that word boundaries are available. We note that subword-based and character-based models also make this assumption. In BERT, for instance, word-level tasks are formulated such that a word's label is assigned to

Language	ISO 639-3	Language Family	Script
Amharic	AMH	Afro-Asiatic	Ge'ez
Arabic	ARA	Afro-Asiatic	Arabic
Bengali	BEN	Indo-European	Bengali
Bulgarian	BUL	Indo-European	Cyrillic
Chinese	ZHO	Sino-Tibetan	Chinese
Coptic	COP	Afro-Asiatic	Coptic
English	ENG	Indo-European	Latin
Finnish	FIN	Uralic	Latin
French	FRA	Indo-European	Latin
German	DEU	Indo-European	Latin
Greek	ELL	Indo-European	Greek
Hausa	HAU	Afro-Asiatic	Latin
Hindi	HIN	Indo-European	Devanagari
Igbo	IBO	Niger-Congo	Latin
Indonesian	IND	Austronesian	Latin
Japanese	JPN	Japonic	Japanese
Kinyarwanda	KIN	Niger-Congo	Latin
Korean	KOR	Koreanic	Korean
Luganda	LUG	Niger-Congo	Latin
Luo	LUO	Nilo-Saharan	Latin
Naija Pidgin	PCM	English Creole	Latin
Russian	RUS	Indo-European	Cyrillic
Spanish	SPA	Indo-European	Latin
Swahili	SWA	Niger-Congo	Latin
Tamil	TAM	Dravidian	Tamil
Telugu	TEL	Dravidian	Telugu
Thai	THA	Kra-Dai	Thai
Turkish	TUR	Turkic	Latin
Urdu	URD	Indo-European	Perso-Arabic
Vietnamese	VIE	Austro-Asiatic	Latin
Wolof	WOL	Niger-Congo	Latin
Yorùbá	YOR	Niger-Congo	Latin

Table 2.8: Overview of languages used in our experiments.

its first subword token, requiring word boundaries. During training, continuation tokens are then masked out when computing the loss. Consequently, predictions for continuation tokens also need to be masked out at inference time, which again requires word boundaries or aggregation strategies that may introduce errors. The same applies to character-based models. For PIXEL, should this assumption be violated, it is still possible to render the text without adding spacing, although the mapping is then no longer bijective as multiple words can overlap on one image patch. In such cases, assigning the prediction for a patch to either word can cause loss of information. Although in practice this approach does not necessarily affect performance negatively, future work will investigate alternative approaches.

Dataset	Download Link & Reference
Universal Dependencies 2.10	LINDAT repository/xmlui/handle/11234/1-4758 Zeman et al. (2022); Nivre et al. (2020)
MasakhaNER	masakhane-io/masakhane-ner/tree/main/data Adelani et al. (2021)
GLUE	datasets/glue Wang et al. (2018)
TyDiQA-GoldP	datasets/tydiqa Clark et al. (2020)
SQuADv1.1	datasets/squad Rajpurkar et al. (2016)
KorQuAD 1.0	datasets/squad_kor_v1 Lim et al. (2019)
JaQuAD	datasets/SkelterLabsInc/JaQuAD So et al. (2022)
XNLI	datasets/xnli Conneau et al. (2018)

Table 2.9: Links and references to the datasets we used in our finetuning experiments.

Language	Treebank	#Sentences	Reference
ENG	English-EWT	16621	Silveira et al. (2014)
ARA	Arabic-PADT	7664	Hajič et al. (2009)
COP	Coptic-Scriptorium	2011	Zeldes and Abrams (2018)
HIN	Hindi-HDTB	16647	Palmer et al. (2009)
JPN	Japanese-GSD	8100	Asahara et al. (2018)
KOR	Korean-GSD	6339	Chun et al. (2018)
TAM	Tamil-TTB	600	Ramasamy and Žabokrtský (2012)
VIE	Vietnamese-VTB	3000	Nguyen et al. (2009)
ZHO	Chinese-GSD	4997	Shen et al. (2016)

Table 2.10: Overview of the Universal Dependencies v2.10 (Zeman et al., 2022; Nivre et al., 2020) treebanks used in our POS tagging and dependency parsing experiments with the number of sentences in their respective training splits. As mentioned in § 2.3.1, these treebanks were chosen with typological and script diversity in mind.

Parameter	POS	DP	NER	QA	XNLI
Rendering backend			PangoCairo		
CLS head pooling	—	—	—	—	CLS
Optimizer			AdamW		
Adam β			(0.9, 0.999)		
Adam ε			$1e{-}8$		
Weight decay			0		
Learning rate (LR)	$5e{-}5$	$\{5e{-}5, 8e{-}5\}$	$5e{-}5$	$\{3e{-}5, 5e{-}5, 7e{-}5\}$	$2e{-}5$
LR warmup steps	100	100	100	100	1000
LR schedule			Linear decay		
Max sequence length	256	256	196	400	196
Stride	—	—	—	160	—
Batch size	64	64	64	32	256
Max steps	15000	15000	15000	20000	50000
Early stopping			✓		
Eval steps	500	500	500	500	1000
Dropout probability			0.1		

Table 2.11: Finetuning settings for POS tagging, dependency parsing (DP), NER, QA, and XNLI. We did not run a comprehensive hyperparameter search due to compute limitations; these settings were manually selected based on a small number of preliminary runs. Maximum performance was often reached well before the specified number of max steps.

Parameter	MNLI	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	WNLI
Rendering backend	PangoCairo	PyGame	PangoCairo	PyGame	PyGame	PyGame	PyGame	PyGame	PyGame
CLS head pooling					Mean				
Optimizer					AdamW				
Adam β					(0.9, 0.999)				
Adam ε					$1e{-}8$				
Weight decay					0				
Learning rate (LR)	$3e{-}5$	$3e{-}5$	$3e{-}5$	$3e{-}5$	$2e{-}5$	$2e{-}5$	$3e{-}5$	$3e{-}5$	$1e{-}5$
LR warmup steps	100	100	100	100	200	100	100	200	100
LR schedule					Linear decay				
Max sequence length					256				
Batch size	64	256	64	256	256	64	64	64	256
Max steps	15000	15000	15000	15000	15000	15000	15000	15000	400
Early stopping					✓				
Eval interval	500 steps	500 steps	500 steps	500 steps	100 steps	100 steps	100 steps	250 steps	1 epoch
Dropout probability					0.1				

Table 2.12: Finetuning settings for GLUE tasks. We did not run a comprehensive hyperparameter search due to compute limitations; these settings were manually selected based on a small number of preliminary runs. Increasing the batch size to 256 and switching to the PyGame renderer helped achieve more consistent convergence behaviour for some tasks. For the smaller datasets (to the right of QQP), maximum performance was reached well before the specified number of max steps.

2.7.7 Examples of *Zeroé* orthographic attacks

Attack	Sentence
NONE	Penguins are designed to be streamlined
CONFUSABLE	Peπguɩns are desigπed to be streaMlıπed
SHUFFLE (INNER)	Pegnuins are dnesiged to be sieatrnmled
SHUFFLE (FULL)	ngePnius rae dsgednei to be etimaslernd
DISEMVOWEL	Pngns r dsgnd to be strmlnd
INTRUDE	Pe‘nguins a{re d)esigned t;o b*e stre<amlined
KEYBOARD TYPO	Penguinz xre dwsigned ro ne streamllned
NATURAL NOISE	Penguijs ard design4d ti bd streamlinfd
TRUNCATE	Penguin are designe to be streamline
SEGMENTATION	Penguinsaredesignedtobestreamlined
PHONETIC	Pengwains’s ar dhiseind te be storimlignd

Table 2.13: Examples of low-level orthographic attacks based on the *Zeroé* benchmark.

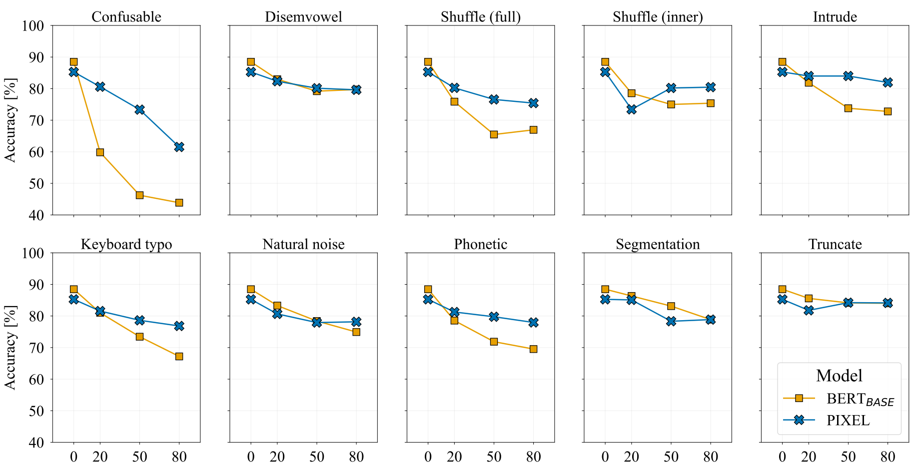


Figure 2.8: Test set accuracy for a single run of PIXEL and BERT across different levels of noise introduced through various orthographic attacks in SNLI. The results show that PIXEL is more robust than BERT to most of these attacks.

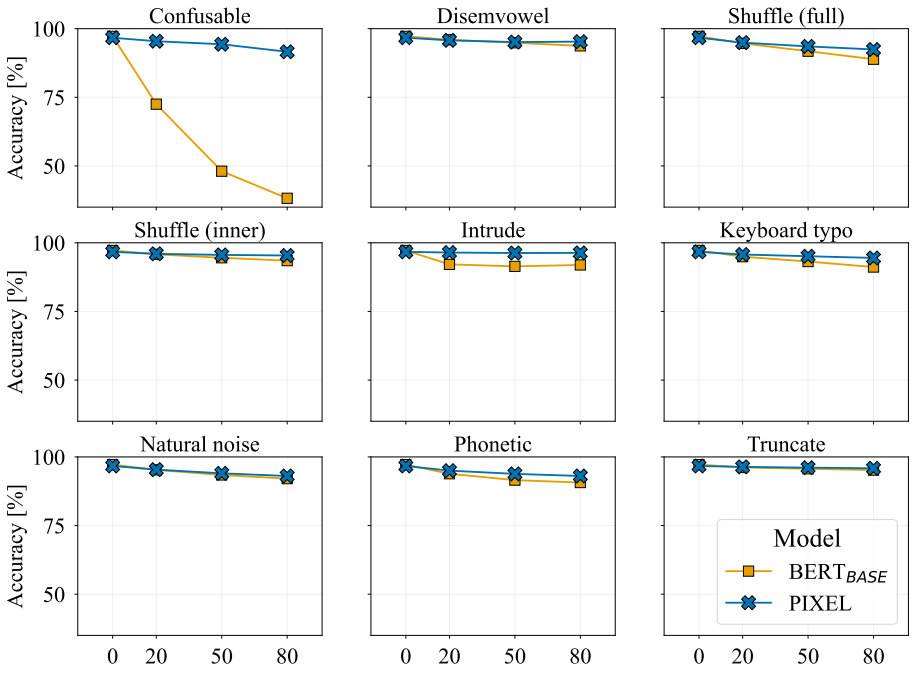


Figure 2.9: Test set accuracy for a single run of PIXEL and BERT across different levels of noise introduced through various orthographic attacks in POS tagging. The results show that PIXEL is more robust than BERT to most of these attacks, especially when dealing with visually-confusable character substitutions. SEGMENTATION is not applied to the task of POS tagging, since the joined words would not have a proper tag.

2.7.8 Font Transfer Analysis

In this section, we analyse the adaptation capabilities of PIXEL to new fonts at finetuning time. Specifically, we finetune PIXEL models for POS tagging and dependency parsing on the UD_English-EWT treebank and sentiment analysis on SST-2, once with a font similar to our `GoNotoCurrent / NotoSans-Regular` pretraining font, `NotoSerif-Regular`, and once with a font strikingly different from it, `JournalDingbats1`. We compare the three fonts in Table 2.14 below:

The font transfer results are shown in Table 2.15. We find that PIXEL exhibits fairly high font transfer ability *out-of-the-box*, i.e. without any font or image augmentation strategies employed during pretraining.[32] In line with our

[32]We believe such augmentation strategies would further improve robustness to font variations

Font	Rendered Example Sentence
GoNotoCurrent	My cat loves oatmeal and pancakes.
NotoSerif-Regular	My cat loves oatmeal and pancakes.
JournalDingbats1	

Table 2.14: An example sentence rendered in three different fonts.

	GoNotoCurrent	NotoSerif-Regular	JournalDingbats1
POS	96.7	95.9	93.9
DP	90.6	88.1	81.3
SST-2	89.6	84.2	72.9

Table 2.15: Results for fine-tuning PIXEL for POS tagging, dependency parsing (DP), and sentiment analysis on SST-2 with three different fonts: the font used in pretraining (GoNotoCurrent), a visually similar font (NotoSerif-Regular), and a highly dissimilar font (JournalDingbats1). We report test accuracy for POS, test LAS for DP, and validation accuracy for SST-2, each averaged over 5 runs.

expectations, transfer to a visually similar font (NotoSerif-Regular) is easier than to a dissimilar font (JournalDingbats1). Nevertheless, PIXEL is able to transfer surprisingly well to the JournalDingbats1 font, in which every letter is simply mapped to the icon of an object or animal.

2.7.9 Further analysis

To investigate where PIXEL currently lags behind BERT, we analyse the impact that dependency length has on both models in dependency parsing in ENG. We can see in Figure 2.10 that the LAS gap between BERT and PIXEL increases with longer dependencies, indicating that PIXEL struggles slightly more with long syntactic dependencies.

2.7.10 Limitations

This paper introduces a new approach to processing written language as images, which removes the need for a finite vocabulary, providing a solution to the *vocabulary bottleneck*. While our results show that PIXEL is a promising approach

and leave this experiment to future work. Considering that we have full control over the font when working with NLP text datasets, robustness to font variations was not a primary goal in this work.

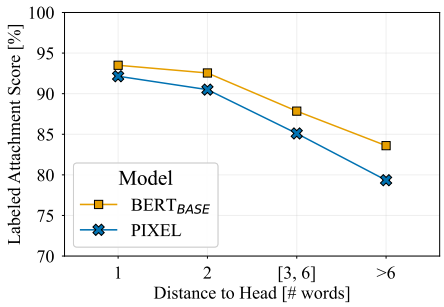


Figure 2.10: LAS scores (ENG) across different dependency lengths averaged over 5 random intitializations of BERT and PIXEL. In ENG, long syntactic dependencies are more challenging for PIXEL.

	#L	|θ|	ENG	ARA	BUL	DEU	ELL	FRA	HIN	RUS	SPA	SWA	THA	TUR	URD	VIE	ZHO
MBERT	104	179M	83.3	73.2	77.9	78.1	75.8	78.5	70.1	76.5	79.7	67.2	67.7	73.3	66.1	77.2	77.7
BERT	1	110M	**83.7**	**64.8**	**69.1**	**70.4**	**67.7**	**72.4**	**59.2**	**66.4**	**72.4**	**62.2**	35.7	**66.3**	**54.5**	**67.6**	46.2
PIXEL	1	86M	77.2	58.9	66.5	68.0	64.9	69.4	57.8	63.4	70.3	60.8	**50.2**	64.0	54.1	64.8	**52.0**

Table 2.16: Results for PIXEL and BERT finetuned on XNLI in the *translate-train-all* setting where we train on the joint training data in all 15 languages, originally translated from ENG by Conneau et al. (2018). We report test set accuracy averaged over 5 runs each. Despite the relatively large performance gap in favor of BERT in ENG (which is in line with the GLUE results in Table 2.3), the gap is much smaller for other languages, particularly those not using the Latin writing system. PIXEL is overall more consistent across scripts, outperforming BERT in THA and ZHO.

in this direction, this is only the first step. Here, we highlight current limitations and avenues for future work for pixel-based models:

- PIXEL is pretrained on predominantly English text written in the Latin script. The choice of English is driven by the scientific goal of comparing against a widely used model (English BERT) but English may not be the best source language for cross-lingual transfer (Turc et al., 2021; Blevins et al., 2022). We expect that PIXEL trained on typologically diverse languages in multiple scripts would considerably surpass the cross-script and cross-lingual transferability of English-only PIXEL but this remains to be verified, and training a model on large amounts of data will require large computational resources.

- PIXEL currently seems to be less sample-efficient than subword-based PLMs. PIXEL excels at syntactic tasks after being pretrained for the same number of steps/datapoints as BERT (a challenging setup within an academic budget), but still lags behind in semantic processing. As a consequence, it also requires more training steps than BERT to converge during finetuning. Closing this gap might involve longer pretraining with additional (long-dependency) objectives.
- There are challenges to be addressed when working with languages written right-to-left. PIXEL currently processes sentences in such languages from the end to the beginning, which may lead to learning inadequate features for sentence separation and position embeddings.
- PIXEL cannot be used for language generation tasks because it is not possible to produce discrete words from the pretrained decoder.
- Rendering text as images requires more disk space than reading text from a file. This can be alleviated by caching the dataset in a compressed format or rendering the images on-the-fly. Rendering images on-the-fly will create additional overhead when training for multiple epochs.

Chapter 3

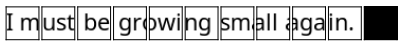
Text Rendering Strategies for Pixel Language Models

The work presented in this chapter is based on a paper that has been published as: Jonas Lotz, Elizabeth Salesky, **Phillip Rust**, and Desmond Elliott. 2023. Text rendering strategies for pixel language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10155–10172, Singapore. Association for Computational Linguistics.

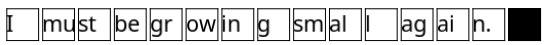
Abstract

Pixel-based language models process text rendered as images, which allows them to handle any script, making them a promising approach to open vocabulary language modelling. However, recent approaches use text renderers that produce a large set of almost-equivalent input patches, which may prove sub-optimal for downstream tasks, due to redundancy in the input representations. In this paper, we investigate four approaches to rendering text in the PIXEL model (Rust et al., 2023), and find that simple character bigram rendering brings improved performance on sentence-level tasks without compromising performance on token-level or multilingual tasks. This new rendering strategy also makes it possible to train a more compact model with only 22M parameters that performs on par with the original 86M parameter model. Our analyses show that character bigram rendering leads to a consistently better model but with an anisotropic patch embedding space, driven by a patch frequency bias, highlighting the connections between image patch- and tokenization-based language models.

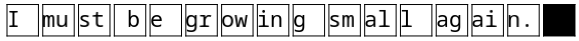
xplip/pixel/tree/TextRenderingStrategies 🤗 Team-PIXEL



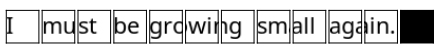
(a) Continuous rendering (CONTINUOUS)



(b) Structured rendering (BIGRAMS)



(c) Structured rendering (MONO)



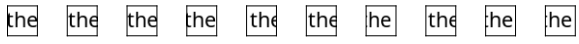
(d) Structured rendering (WORDS)

Figure 3.1: Examples of rendering strategies for the sentence "*I must be growing small again.*" from Carroll (1865a). Black patches mark the end of a sequence, following Rust et al. (2023).

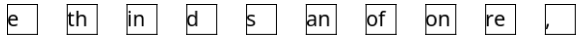
3.1 Introduction

There is a growing movement in NLP towards tokenization-free methods (Clark et al., 2022; Xue et al., 2022; Yu et al., 2023) including pixel-based representations of text (Salesky et al., 2021, 2023a; Rust et al., 2023; Tschannen et al., 2023). It has been shown that these tokenization-free methods can readily handle unseen languages and that they are more robust to noise attacks than tokenization-based models. In addition, pixel-based approaches can effectively exploit visual similarities between characters and scripts because they allow for complete parameter sharing across all inputs, making them a promising direction for multilingual NLP.

Previous work on pixel-based models segments the rendered text into either consecutive patches (Rust et al., 2023; Tschannen et al., 2023) or with a sliding window (Salesky et al., 2021, 2023a) as in speech processing. Although the proposed approaches have the appealing properties of yielding compact and transferable representations, they also result in a very large input space because there is no unique way to represent lexical units. As a consequence, pixel-based models could observe a new set of *image* representations with every new sentence, which adds redundancy in the input space and is sub-optimal for developing contextual *language* representations. We refer to these unstructured rendering strategies as CONTINUOUS and illustrate the point qualitatively in Figure 3.1 and Figure 3.2, and quantitatively in Figure 3.3. In this work, we ask whether



(a) Most frequent patches with CONTINUOUS rendering



(b) Most frequent patches with BIGRAMS rendering

Figure 3.2: A continuous rendering strategy results in many uniquely-valued image patches for similar inputs, while structured rendering (here, BIGRAMS) regularises and compresses the potential input space.

structuring the input, which leads to more frequent parameter updates through now-unique word representations, would enable pixel-based models to develop a deeper understanding of context and semantics. We then propose rendering strategies structured around providing the model with a compressed input space.

We demonstrate how enforcing a BIGRAMS-structured rendering strategy leads to both a more capable and data-efficient model: when evaluated on semantic sentence-level tasks, we find that a 22M parameters model performs competitively with the unstructured original at 86M parameters, and that scaling back up to 86M parameters narrows the performance gap to BERT (Devlin et al., 2019) trained on the same data. In subsequent analyses, we find that the added input structure provokes a clear visual token frequency bias in the learned embedding space. While also found in BERT, frequency biases have been shown to degrade the quality of embedding spaces when word representations are not only determined by semantic relations but also by the number of model updates (Gong et al., 2018; Gao et al., 2019b; Fuster Baggetto and Fresno, 2022). We show that frequent words have more context-specific representations than infrequent words, especially in the upper layers. Finally, we show that PIXEL models acquire a non-trivial semantic understanding during pretraining, but that their sentence representations are easily influenced by this frequency bias. We release all models and code for pretraining and finetuning.

3.2 Background: Modelling text as images

We build upon the general-purpose language encoder framework presented in Rust et al. (2023): PIXEL is a text autoencoder which builds on the Masked Autoencoding Vision Transformer (ViT-MAE; He et al., 2022) and is similarly pretrained with a masked reconstruction objective. However, instead of patches

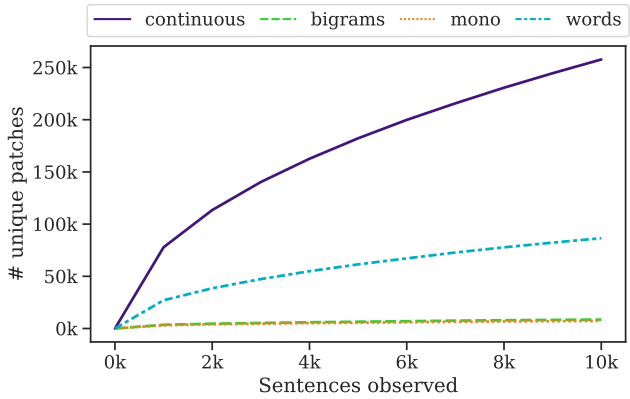


Figure 3.3: Number of unique image patches observed as a function of training data sequences. Structured rendering results in greater representational efficiency.

from natural images of objects (Deng et al., 2009), the patches now contain images of text. To go from text to images of text, PIXEL relies on a rendering library (PangoCairo)[1] to produce a sequence-level image which is sliced into image patches of size 16×16 pixels. The sequence-length maximum of 529 patches approximately equals the memory requirements of BERT, the closest benchmark for PIXEL. By using the Google Noto font family which supports the majority of Unicode codepoints,[2] the renderer supports all languages that can currently be typeset.

Before the first layer of the PIXEL model, image patches are linearly projected to obtain a sequence of patch 'embeddings'. During pretraining, 25% of embeddings are masked in spans of up to 6 patches and only the unmasked patches with a prepended CLS embedding are passed through the encoder. After replacing the masked embeddings amidst the encoder outputs, relying on fixed sinusoidal position embeddings for ordering information, the decoder predicts the pixel values of solely the masked patches. To later finetune the encoder on a classification task, the decoder can be replaced with a task-specific head and the masking ratio set to 0%.

[1] https://docs.gtk.org/PangoCairo
[2] https://fonts.google.com/noto

3.3 Structured rendering

Previously proposed approaches to rendering text as images render full sequences of text and segment into either consecutive patches (Rust et al., 2023; Tschannen et al., 2023) or with a sliding window (Salesky et al., 2021, 2023a). These CONTINUOUS strategies result in a significant number of uniquely-valued patches, many of which may be observed only once during training. We depict this redundancy in Figure 3.2 and quantify it in Figure 3.3, showing how similar text inputs result in unique visual representations.

We compare four rendering strategies: the original unstructured (CONTINUOUS), and three structured (WORDS, MONO, BIGRAMS), as depicted in Figure 3.1. To render WORDS we separate segments with additional whitespace[3] such that new segments begin at the beginning of the next image patch, regulating possible spatial variation. BIGRAMS, rendering two characters per image patch, is chosen to be widely applicable, without knowledge of word or morphemic segmentation (Mielke et al., 2021; Keren et al., 2022). More specifically—consider the word pairs ⟨"grow", "growing"⟩ and ⟨"growing", "walking"⟩—the BIGRAMS renderer will produce an overlap of image patches (underlined) for both pairs while the same extent is not guaranteed with WORDS-level rendering as it is regulated by character width. The choice of character $(n = 2)$-grams is motivated by what generally fits within a 16×16 pixels image patch in the setup from Rust et al. (2023). MONO instead applies monospaced fonts where each character is a fixed width; depending on font size, this may result in character bigram patches without breaks within characters, but this is not guaranteed. The main difference between BIGRAMS and MONO is that MONO simply slides across the sentence, two characters at the time, yielding two ways to represent a word whereas BIGRAMS renders the words and then pads with whitespace, ensuring unique inputs.[4]

As seen in Figure 3.3, the structured rendering strategies result in a greatly compressed input space as measured by the number of unique image patches processed by the model, but Figure 3.1 reveals that it comes at the cost of longer sequence lengths. While the rendering strategies we propose were not specifically designed for English, they may not equally generalise to other languages or scripts. We further discuss the representational efficiencies of these strategies in § 3.9.1

[3] We render whitespace at minimum 3 pixels wide, sometimes resulting in a blank patch between tokens in structured inputs.

[4] As an example, "be" in Figure 3.1 is split into 2 image patches with MONO rendering. Depending on the context, it could also be represented in a single image patch.

Model	Enc_L-Dec_L	Hid	MLP	Att	$\lvert\theta\rvert$
BASE	12-8	768	3072	12	86M
SMALL	12-4	384	1536	6	22M
TINY	12-2	192	768	3	5.5M

Table 3.1: Details of PIXEL model scale variants.

and limitations to generalisability under § 3.8.

3.4 Model scale variants

Recall from Figure 3.3 that CONTINUOUS rendering produces a significantly larger set of unique image patches compared to other approaches. A consequence of this is that models must learn to encode many almost-identical visual representations, which may be wasteful, both in terms of parameters and training efficiency. Therefore, we hypothesise that PIXEL models that operate over fewer unique image patches can be scaled down without sacrificing performance. While "Base" models and larger ones are widely used for their strong performance, proven scaling laws (Touvron et al., 2021; Zhai et al., 2022) enable greater experimentation and model development at smaller scale (Ivgi et al., 2022), which is both more environmentally friendly (Strubell et al., 2019; Bender et al., 2021; Hershcovich et al., 2022b) and facilitates contributions with limited computational resources.

With this in mind, we propose two smaller architectures which we will compare across downstream tasks in § 3.5. Our BASE model architecture is directly adopted from ViT (Dosovitskiy et al., 2021) and PIXEL, and we add two more compact SMALL and TINY model variants, as described in Table 3.1. The configurations of the smaller models are based on the ViT variants presented in Zhai et al. (2022). Following the scaling experiments in He et al. (2022), indicating that shallow decoders of as small as 2 layers can be sufficient for ViT-MAEs, we apply a scheme of halving the number of decoder layers at every scale reduction.

3.5 Experiments

We pretrain SMALL models with the proposed rendering strategies. The models are then evaluated on dependency parsing (UDP) with data from Universal Dependencies v2.10 treebanks (Zeman et al., 2022; Nivre et al., 2020) and GLUE (Wang et al., 2018), exploring the models' capabilities at syntactic processing on

Structure			Scale							
	UDP	GLUE			UDP		GLUE		TyDiQA-GoldP	
Renderer	*Avg.*	*Avg.*	*Variant*	$\lvert\theta\rvert$	*Avg.*	$\Delta\mu$	*Avg.*	$\Delta\mu$	*Avg.*	$\Delta\mu$
CONTINUOUS	76.2	71.0	TINY	5.5M	72.0	−0.3	66.5	+12.7	41.6	+4.9
BIGRAMS	76.1	75.4	SMALL	22M	76.1	−0.1	75.4	+4.4	50.8	+2.0
MONO	75.9	74.4	BASE	86M	75.5	−0.6	78.0	+3.9	52.8	+0.5
WORDS	76.6	74.7	BERT	110M	50.5	—	80.0	—	51.5	—

Table 3.2: **Structure** (left): averaged results for SMALL-models comparing downstream performance on UDP and GLUE following the different rendering strategies. **Scale** (right): averaged results across model scales using the BIGRAMS rendering structure. $\Delta\mu$ is the difference in average performance between BIGRAMS and CONTINUOUS rendering for a given model scale. BERT results are marked in gray to visually distinguish from pixel-based models.

the word level and semantic processing on the sentence level.

3.5.1 Pretraining

We pretrain all models on the English Wikipedia and Bookcorpus (Zhu et al., 2015) data used by Rust et al. (2023) for direct comparison with PIXEL and BERT, which results in ~16.8M training examples. We follow the suggested hyperparameters used for PIXEL with the exception of batch size. The smaller architectures of SMALL and TINY allow for larger batch sizes, which we double from 256 examples to 512 and 1024, respectively. We then halve the number of pretraining steps accordingly from 1M to 500k and 250k in order to train for the same number of epochs as PIXEL (~16 epochs, but varying slightly due to differing sequence lengths per rendering strategy).

Pretraining BASE takes 8 days on 8×40GB Nvidia A100 GPUs, while in comparison, pretraining SMALL takes less than 48 hours on 8×40GB Nvidia A100 GPUs, and TINY less than 24 hours. Loss trajectories for the different rendering strategies are in line with their representational efficiency (Figure 3.3), indicating that structured rendering may make the masked reconstruction task more data-efficient, achieving a low loss in fewer steps (see § 3.9.2: Figure 3.10).

3.5.2 Finetuning

To finetune our models for classification tasks we replace the decoder used for pretraining with a task-specific classification head. We do not search for more optimal hyperparameters than those used for PIXEL with the exception of the

learning rate; we find that the more compact architectures often benefit from a slightly higher learning rate.[5]

We follow the same protocol during finetuning as done for PIXEL: for word-level tasks we obtain the rendered image patch indices for every word and as a consequence, the CONTINUOUS strategy becomes identical to the WORDS structure when finetuning on UDP. § 3.6.1 further investigates the consequence of a mismatch between how the data is structured during pretraining and finetuning. When finetuning on GLUE the structure follows what was seen during pretraining for all rendering strategies. Reported performances for BERT and PIXEL are taken from Rust et al. (2023).

3.5.3 Rendering strategies

We present averaged results comparing the rendering strategies in the left part of Table 3.2. Detailed results for each downstream task are presented in Table 3.4 and Table 3.5 in the appendix. For UDP we find that the WORDS structure slightly outperforms BIGRAMS and MONO on this word-level task. When comparing the WORDS and CONTINUOUS strategies, we get a first hint as to the importance of including structure during pretraining as well, keeping in mind that the rendering structure is the same for both strategies when finetuning on UDP. For GLUE we see a large increase in performance when rendering with any structure and especially BIGRAMS. We attribute the difference in performance between BIGRAMS and MONO to the unique word representations with BIGRAMS, as discussed in § 3.3.

We find that BIGRAMS is the best performing structure on average, even slightly outperforming the 86M parameters PIXEL (average UDP: 76.1; average GLUE: 74.1) with only ¼ its model parameters. We provide an investigation into the mechanisms that enable this improved performance on GLUE in § 3.6.4. Next we pretrain TINY and BASE model variants with BIGRAMS rendering to evaluate performance at different model scales.

3.5.4 Model scaling

The right part of Table 3.2 compares the different model scales all following a BIGRAMS rendering strategy. Detailed results are likewise presented in Table 3.4, Table 3.5, and Table 3.6 in the appendix. We find that the TINY configuration performs competitively on the word-level tasks considering its only 5.5M parameters, but has a larger gap up to SMALL and BASE on the

[5] We search the space $\{1e{-}5, 3e{-}5, 5e{-}5, 7e{-}5, 9e{-}5\}$ and report the average over 3 seeds.

sentence-level GLUE tasks. SMALL proves to be a good trade-off between scale and performance where it is not far behind BASE on GLUE and even slightly outperforms on UDP.[6] BASE comes a step closer to closing the gap in performance up to BERT on GLUE. Comparing to the performance following a CONTINUOUS rendering strategy, summarised as the difference in average performance ($\Delta\mu$), it is clear that the more compact the model size, the greater the benefit from structured rendering.

To verify that BIGRAMS rendering does not degrade the performance on *multilingual* sentence-level tasks across different scripts and morphologies, we also include results on TyDiQA-GoldP (Clark et al., 2020).[7] Again, we find that SMALL performs competitively considering its size.

3.6 Ablations and supplementary analyses

In this section, we investigate how BIGRAMS rendering changes the model compared to CONTINUOUS. For clarity in what follows, we refer to the BASE model with BIGRAMS rendering from § 3.5.4 as BASE-BIGRAMS and keep referring to the original model from Rust et al. (2023) as PIXEL.

3.6.1 When does rendering structure matter?

Having established that a structured rendering strategy leads to improved downstream performance, we further investigate *when* it is needed: is it sufficient to finetune with structure, or does the model develop strategy-specific features during pretraining? We analyze this by comparing rendering strategies between pretraining and finetuning.

The results in Table 3.3 for GLUE show that a mismatch leads to lower downstream performance for both strategies, with BIGRAMS → CONTINUOUS being the most harmful, perhaps unsurprisingly. This result does not align with the finding for UDP in § 3.5.3 where CONTINUOUS overcomes the change to WORDS-structured rendering. It may indicate that the lower-level UDP tasks are easier for PIXEL-based models than the high-level GLUE tasks (Lauscher et al.,

[6] We expect that BASE could prevail and would benefit from a wider search for optimal hyperparameters during finetuning.

[7] With the CONTINUOUS rendering strategy, answer spans are extracted such that the answer may include leading or trailing characters when there is no exact mapping from a word to an image patch index. Therefore, we did not include TyDiQA-GoldP in the comparison in § 3.5.3. More details can be found in Rust et al. (2023). We discuss limitations to answer span extraction with BIGRAMS rendering in § 3.9.4.

Renderer		GLUE
Pretraining	Finetuning	*Avg*
BIGRAMS	BIGRAMS	75.4
CONTINUOUS	CONTINUOUS	71.0
CONTINUOUS	BIGRAMS	61.1
BIGRAMS	CONTINUOUS	53.0

Table 3.3: Rendering strategy combinations between pretraining and finetuning with SMALL models. For GLUE, matching pretraining structure is most effective.

2020). This is in line with the relatively good performance for TINY-BIGRAMS on UDP.

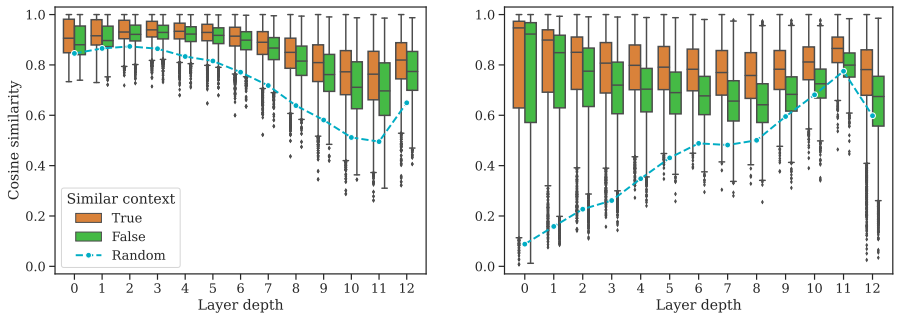
To emphasize the increase in performance on semantic tasks with BIGRAMS rendering, we demonstrate that BASE-BIGRAMS outperforms PIXEL by 3.6 points on average on MasakhaNER (Adelani et al., 2021), a named entity recognition benchmark for 10 African languages. This further illustrates the potential of PIXEL-based models for modelling low-resource languages. Detailed results are presented in Table 3.7 in the appendix. We next turn our attention to *how* BIGRAMS rendering enables better performance on semantic tasks.

3.6.2 Contextual representations

The extent to which language models capture semantic information is partly determined by their ability to contextualise text (Peters et al., 2018). We therefore analyse how capable BASE-BIGRAMS is at producing contextualised word representations. We use the Words in Context dataset (WiC; Pilehvar and Camacho-Collados, 2019) of sentences that contain target words (noun or verb) in either a similar (True) or different (False) context across sentence pairs.[8]

We compute the mean hidden state output over all tokens associated with the target word to obtain a representation. We infer that there is contextualisation if the model generates representations of a target word from different contexts with a low cosine similarity compared to target words in similar contexts. We report this indication of contextuality for each layer of the model, including the input layer, to better understand the properties of the different layers. Similarities between randomly chosen words from random examples (Random) are included

[8]Target words are not necessarily identical across sentence pairs and can vary e.g. in conjugation or number.



(a) BASE-BIGRAMS (b) BERT

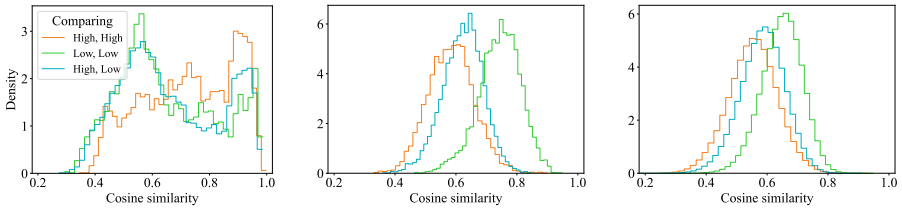
Figure 3.4: Distributions of cosine similarities for verbs and nouns from the WiC dataset across model layers 0-12, layer 0 being the input layer. Every example presents a target word in either a similar or different context across a sentence pair. The representation of the target word is computed as the mean hidden state output over the corresponding tokens. We generally see that BASE-BIGRAMS encodes target words in a similar context as more similar. The median cosine similarity between random words from random sentences are shown as a baseline.

as a baseline.[9]

Figure 3.4a plots the resulting distributions of similarities. We see that representations of target words from similar contexts have a higher cosine similarity than from different contexts, though with a considerable overlap, and higher for different contexts than for random. When comparing to BERT in Figure 3.4b, there is a clear difference in the similarity compared to random words. The difference in similarity between similar and random words gradually increases throughout the BASE-BIGRAMS model, until the final layers, whereas the difference steadily decreases throughout the model for BERT. Given the shared image patch embedding layer in PIXEL-based models, random words are more similar to each other at the input layer when modelled as images than entries in a vocabulary.

Taken together, these plots suggest that a PIXEL-based language model is capable of forming contextualised word representations and that these are more context-specific in upper layers, though not as fine-grained as seen for BERT.

[9]It is not possible to obtain an exact mapping from words to neat image patch indices following the CONTINUOUS rendering strategy, so we do not present this analysis for PIXEL.



(a) Words in isolation, PIXEL (b) Words in isolation, BASE-BIGRAMS (c) Words in context, BASE-BIGRAMS

Figure 3.5: Distributions of cosine similarities within samples of high-frequency words (High), low-frequency words (Low), or between the two samples. Rendering with BIGRAMS structure leads to less directionally aligned vector representations of frequent words that have seen more updates during pretraining compared to infrequent words.

3.6.3 Token frequency and similarity

The degree of cosine similarity between random words observed in Figure 3.4a encourages us to assess the isotropic nature of the model (Ethayarajh, 2019; Rajaee and Pilehvar, 2021). The high cosine similarities suggest that the word representations are not evenly distributed with respect to direction in the embedding space, but instead appear to be anisotropic. When learned vector representations populate a narrow cone in the embedding space, this geometric alignment leads to an overestimation of their similarity (Gao et al., 2019b), which is not an expected property of an expressive word embedding space (Arora et al., 2016; Mu and Viswanath, 2018).[10]

Recent work has shown that Transformer-based language models can develop a representation bias driven by token frequency, where low-frequency tokens are clustered together in the embedding space, leading to anisotropy in the model (Gao et al., 2019b; Fuster Baggetto and Fresno, 2022; Jiang et al., 2022). This bias leads to poor word contextualisation because the learned vector positions of low frequency words have not moved far from their random initialisation. Thus, their embeddings are not sufficiently distinct from unrelated words with similarly low token frequency (Gong et al., 2018; Cai et al., 2021). Tokens

[10]Following Cai et al. (2021) this *global* estimate of ansiotropy does not rule out the possibility of distinct and locally isotropic clusters in the embedding space. Ding et al. (2022) show that isotropy calibration methods (Gao et al., 2019b; Wang et al., 2020b; Li et al., 2020) do not lead to consistent improvements on downstream tasks when models already benefit from local isotropy. We leave this direction for PIXEL to future research.

with a higher frequency, and thus more parameter updates, can move further in the embedding space from their initialisation and become more *semantically meaningful*. Consequently, we hypothesise that compressing the input space in the form of structured rendering allows the model to build more contextualised word representations through more frequent parameter updates.

We investigate this by sampling inputs that were seen during pretraining with high and low frequency. Specifically, we take the 100 most frequently occurring words from the Wikipedia corpus that was seen during pretraining and 100 words that occur around 1000 times (rank $\approx$ 50k).[11] We first render each word from the two frequency samples in isolation. We then include a comparison to words in context across 100 unique sentences per word with BASE-BIGRAMS.[12]

We plot the distributions of cosine similarities between representations from the last encoder layer, where we expect embeddings from both models to be contextualised. Comparing the plots from the two rendering strategies, summarised in Figure 3.5, the effect of pretraining with a smaller set of unique tokens becomes clear: for PIXEL the distribution appears as mixtures with a larger distribution mass at higher values of cosine similarity from comparing high-frequency words to other high-frequency (excluding self-similarity for now) than when comparing low-frequency to other low-frequency. For BASE-BIGRAMS the frequent words both in isolation and in-context are less directionally aligned with each other compared to the infrequent, which is in line with the *representation degeneration problem* from Gao et al. (2019b) and more frequent updates leading to better contextualisation. Figure 3.6 visualises the in-context representations in 2 dimensions using t-SNE (van der Maaten and Hinton, 2008) and provides an additional indication of more frequent words having less locally compact representations.[13]

We expect that in-context representations from PIXEL also qualitatively resemble Figure 3.5a but cannot easily demonstrate this due to the aforementioned challenges in aligning patch embeddings with CONTINUOUS rendering.

3.6.4 Frequency bias and semantic modelling

While there is less evidence of representation degeneration with CONTINUOUS rendering, it is likely that the poorer performance on GLUE in § 3.5.4 is caused

[11] Excluding punctuation and numbers.

[12] Recall from § 3.6.2 that the CONTINUOUS rendering strategy by design makes an exact mapping from words in a sentence to neat image patch indices unattainable.

[13] Plotting the first 2 singular values from a singular value decomposition gives the same qualitative indications.

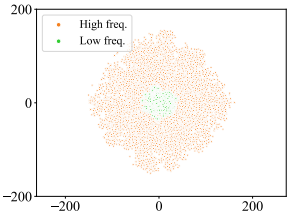


Figure 3.6: t-SNE plot of the output embeddings of high- and low-frequency words in context from BASE-BIGRAMS. Low-frequency words cluster tightly in this space.

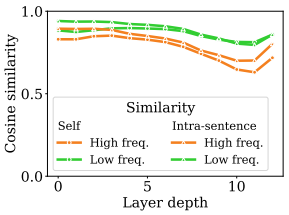


Figure 3.7: Self- and intra-sentence similarity from BASE-BIGRAMS. High-frequency words are the most context-specific; low-frequency words are influenced by their context.

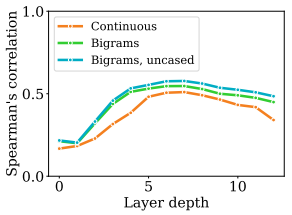


Figure 3.8: Evaluation performance on STS-B. Uncased sentences yield better performance than the original with BASE-BIGRAMS; the effect is less clear for PIXEL (not shown).

by PIXEL seeing too many different patches too few times. This is a direct consequence of the multitude of ways that similar inputs can be rendered by the CONTINUOUS approach. However, the drop in performance when mismatching the rendering strategies in § 3.6.1 for CONTINUOUS → BIGRAMS demonstrates that the model has developed a set of strategy-specific expectations and features that are not easily updated. In fact, the new rendering strategy for finetuning introduces a set of patches that likely never escape the low-frequency domain and therefore remain poorly contextualised.

Signs of a token frequency bias has also been found in BERT (Fuster Baggetto and Fresno, 2022).

We lastly assess the connection between visual token frequency and downstream semantic performance. With BERT, high-frequency words have the most context-specific representations (Ethayarajh, 2019), and upper-layer representations of low-frequency words are influenced more by their context than frequent words (Voita et al., 2019). Following Ethayarajh (2019), we see that this applies to BASE-BIGRAMS as well (illustrated in Figure 3.7 and discussed in greater detail in § 3.9.5). We expect that sentences that only vary in being cased or uncased would result in different representations when lowercase appears more frequently (for most words). This demonstrates the impact of observed token frequency on semantic modelling and is in line with observed biases in BERT's embedding space (Jiang et al., 2022).

We rely on the Semantic Textual Similarity Benchmark (STS-B; Cer et al.,

2017) also found in GLUE for this assessment. We measure the cosine similarity between sentence representations[14] and plot its correlation with the gold standard similarity scores as the measure of performance. Figure 3.8 proves that both CONTINUOUS and BIGRAMS rendering during pretraining lead to non-trivial semantic modelling capabilities. At peak performance, around the middle layers, the increase from simply ensuring that all words are uncased is roughly the same as the increase from PIXEL to BASE-BIGRAMS. This resembles how frequent and infrequent tokens have unequal influence *on* their context in BERT (Voita et al., 2019).

Seeing that BASE-BIGRAMS exhibits similar representational traits to that of BERT, future work could aim for more semantically capable PIXEL-based models by generalising advances found for tokenizer-based models (Gao et al., 2021).

3.7 Related work

Recent work on pixel-based language modelling has demonstrated how visual language understanding can be achieved through pixels only (Lee et al., 2023), observed that the visual similarity of languages plays an important role in cross-lingual transfer (Rahman et al., 2023), and shown how unifying the modalities for text and images allows a single encoder to perform multimodal tasks (Tschannen et al., 2023). By relying on bytes directly, the unification of modalities can be taken even further (Jaegle et al., 2021; Horton et al., 2023; Yu et al., 2023). The work most closely related to ours, after Rust et al. (2023), is the work on machine translation with pixel representations (Salesky et al., 2021, 2023a). A detailed discussion of previous pixel-based approaches can be found in Rust et al. (2023, § 5). Where PIXEL laid the foundation for general-purpose language encoding with pixel-based representations, this work takes the first step towards hypothesis-driven improvements without adding additional data (Yang et al., 2019) or scaling up the model (Conneau and Lample, 2019). Though it is possible that competitive performance could be achieved by a model with CONTINUOUS rendering by pretraining on more data for more steps (Liu et al., 2019).

Our addition of BIGRAMS structure resembles the addition of optional but hugely beneficial ($n = 4$)-grams in the character-based CANINE model (Clark et al., 2022). While character-level n-gram models (Wieting et al., 2016; Bojanowski et al., 2017) have been succeeded by Transformer-based language

[14] Mean hidden state output across all tokens in a sentence, excluding the CLS token and black end-of-sequence token.

models, character-level features remain valuable as they are less sparse and more robust to misspellings than word n-grams, and remain useful for especially morphologically rich languages (Garrette and Baldridge, 2013; Kulmizev et al., 2017). Previous works have hypothesised that character-level models would be more suitable than subword-based for modelling morphologically-rich languages (Tsarfaty et al., 2020; Keren et al., 2022), but a semantically capable design has proven non-obvious (Ma et al., 2020; Keren et al., 2022; Nzeyimana and Niyongabo Rubungo, 2022; Sun et al., 2023). We see potential for future work with pixel-based language models, exploring appropriate strategies for learning morphological patterns (Klein and Tsarfaty, 2020; Seker and Tsarfaty, 2020; Soulos et al., 2021).

3.8 Conclusion

We evaluate four text rendering strategies to address the problem of redundancy in the input space of PIXEL-based language models. Consequently, more frequent parameter updates lead to better contextualised language representations. We find that rendering two characters per image patch (BIGRAMS) is a good trade-off between efficiency and generalisability, resulting in substantial improvements on downstream semantic and sentence-level tasks; contributing to open-vocabulary NLP with limited computational resources.

Further analyses reveal how the added rendering structure provokes clear representational similarities to what has been found in BERT. We see potential in future work generalising improvements found for tokenization-based masked language models to PIXEL-based masked language models. Furthermore, considering that the Vision Transformer has also been applied to speech modelling (Huang et al., 2022), and that patch representation has been suggested to be a critical component for the success of ViTs (Trockman and Kolter, 2023), we see potential for image patches as the basis for unifying modalities.

Limitations

While the rendering strategies we propose here are well-suited to English, not all equally generalise to other languages or scripts. WORDS rendering relies on word boundaries which may not be readily available or well-defined for many languages which do not mark word or sentence boundaries with whitespace such as Thai or polysynthetic languages such as Inuktitut. MONO and BIGRAMS are more general

approaches, but may affect the rendering of positional characters such as diacritics or correct contextual forms based on where boundaries are created. For both approaches, it may be necessary to modulate font size across languages to ensure character pairs fit into a single patch, especially when rendering with diacritics. MONO provides further representational efficiency compared to BIGRAMS by fixing character width, but comes at the cost of more limited language coverage; many scripts cannot be made fixed-width and fewer than 10 have mono fonts available. CONTINUOUS rendering provides a more general approach which must be balanced with learning efficiency.

Acknowledgements

Jonas F. Lotz is funded by the ROCKWOOL Foundation (grant 1242). Elizabeth Salesky is supported by the Apple Scholars in AI/ML fellowship. Phillip Rust is funded by the Novo Nordisk Foundation (grant NNF 20SA0066568). This work was supported by a research grant (VIL53122) from VILLUM FONDEN.

3.9 Appendix

3.9.1 Representational efficiency

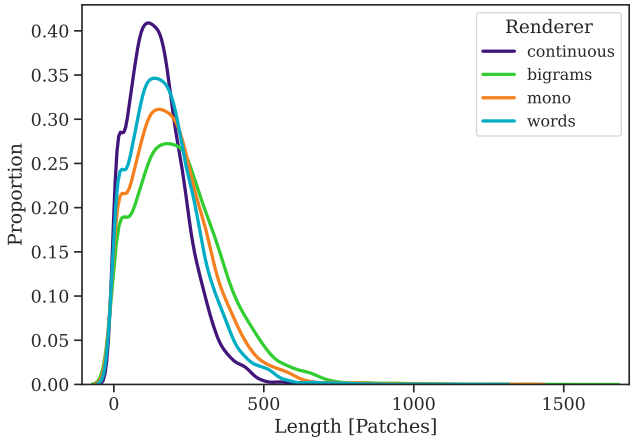


Figure 3.9: Distributions of sequence lengths (in patches) resulting from different rendering strategies.

As seen in Figure 3.1, structured rendering compresses the input space by reducing the positions characters may be observed in. This dramatically affects the number of unique inputs observed in a fixed number of sequences, as quantified in Figure 3.3. Concretely, the 10 most frequently observed image patches after processing 100,000 sequences from English Wikipedia are shown in Figure 3.2; with continuous rendering all are positional variants of the same subword, while with structured rendering each represents different words or morphemes. However, instituting word- or subword-level structure with whitespace padding increases sequence lengths compared to unstructured rendering as quantified in Figure 3.9.

3.9.2 Pretraining loss curves

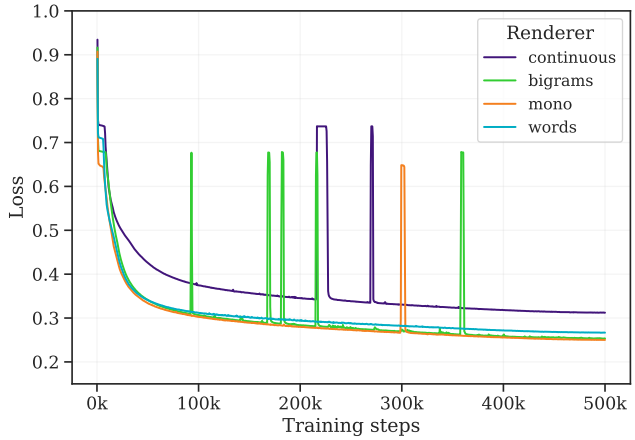


Figure 3.10: Pretraining loss for SMALL models with different rendering strategies, indicating that structured rendering may make the masked reconstruction task more data efficient, reaching a low loss in fewer steps.

3.9.3 Detailed experimental results

3.9.4 TyDiQa-GoldP

The CONTINUOUS rendering strategy used for PIXEL, in which words often overlap in an image patch, leads to extracted answer spans that potentially include leading or trailing characters that should not be part of the answer. BIGRAMS

	ENG	ARA	COP	HIN	JPN	KOR	TAM	VIE	ZHO	Avg
BERT	90.6	77.7	13.0	75.9	73.8	30.2	15.2	49.4	28.8	50.5
PIXEL	88.7	77.3	83.5	89.2	90.7	78.5	52.6	50.5	73.7	76.1
TINY-CONTINUOUS	78.9	74.6	80.0	87.9	89.9	75.1	48.3	46.2	69.5	72.3
Structure										
SMALL-CONTINUOUS	87.2	77.2	83.4	88.9	91.0	78.8	53.8	51.9	73.5	76.2
SMALL-BIGRAMS	87.9	75.4	84.1	88.9	90.8	79.4	53.9	50.9	73.9	76.1
SMALL-MONO	88.3	76.8	83.4	88.9	91.0	79.0	50.5	51.3	73.8	75.9
SMALL-WORDS	88.0	77.2	83.9	89.3	91.2	78.7	53.7	53.3	74.2	76.6
Scale										
TINY-BIGRAMS	82.9	70.6	79.1	86.2	90.0	76.2	44.9	47.6	69.8	72.0
SMALL-BIGRAMS	87.9	75.4	84.1	88.9	90.8	79.4	53.9	50.9	73.9	76.1
BASE-BIGRAMS	89.6	77.7	81.4	88.6	90.8	78.1	49.8	49.4	73.9	75.5

Table 3.4: Test set LAS results for dependency parsing on a selection of Universal Dependencies treebanks (UDP).

rendering adressess this issue by yielding clear word boundaries in the input representations.

However, the BIGRAMS rendering strategy poses new challenges to extracting answer spans for TyDiQA-GoldP. While the task is simplified compared to the primary task by removing language tracks that lack whitespace,[15] we find that a surprisingly high number of "words" are a string of comma-separated words or concatenations of characters and letters that should be delimited by whitespace. By design we consider and render these as one unit when we only split by whitespace. An example of a single "unit" from the training split highlights this issue more clearly: "oikeudet[1]Lääni[1]1**Vilna**523,0501387Vilnan"[16] where the expected answer is "**Vilna**" and highlighted in **bold**. In such an instance, a PIXEL BIGRAMS model will predict the whole unit, resulting in a lower performance. Furthermore, some of these "words" in the training data are more than a thousand characters long and therefore do not fit within the maximum sequence length of 529 patches.

[15] google-research-datasets/tydiqa/blob/master/gold_passage_baseline/README.md
[16] id = finnish-1438027099681899178-6

	MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	WNLI	Avg
BERT	84.0 / 84.2	87.6	91.0	92.6	60.3	88.8	90.2	69.5	51.8	80.0
PIXEL	78.1 / 78.9	84.5	87.8	89.6	38.4	81.1	88.2	60.5	53.8	74.1
TINY-CONTINUOUS	36.7 / 37.0	76.6	72.9	87.2	2.1	25.1	82.4	58.5	59.2	53.8
Structure										
SMALL-CONTINUOUS	72.2 / 73.6	84.8	86.2	88.3	19.1	81.7	84.6	61.4	57.7	71.0
SMALL-BIGRAMS	77.3 / 78.1	85.7	87.8	90.4	42.3	84.3	87.8	63.5	56.3	75.4
SMALL-MONO	77.4 / 77.6	84.7	86.8	89.4	42.3	82.4	86.9	57.5	58.9	74.4
SMALL-WORDS	76.7 / 77.3	84.5	86.6	89.9	44.6	80.5	87.4	62.8	56.3	74.7
Scale										
TINY-BIGRAMS	60.8 / 61.9	79.6	81.7	87.2	15.6	77.9	83.0	59.4	57.7	66.5
SMALL-BIGRAMS	77.3 / 78.1	85.7	87.8	90.4	42.3	84.3	87.8	63.5	56.3	75.4
BASE-BIGRAMS	81.1 / 81.4	87.6	89.7	90.4	53.3	86.6	90.2	63.5	56.3	78.0

Table 3.5: Validation set performance on GLUE. The reported metrics are F_1 score for QQP and MRPC, Matthew's correlation for CoLA, Spearman's ρ for STS-B, and accuracy for the rest.

3.9.5 Measuring self-similarity and intra-sentence similarity

We follow Ethayarajh (2019) and measure the degree of self-similarity and intra-sentence similarity for the words in the two frequency samples from § 3.6.3. Self-similarity is computed as the cosine similarity between the same word in different sentences and a high degree therefore indicates that representations vary little across contexts. For intra-sentence similarity we compute the cosine similarity between a word representation and the sentence representation (mean hidden state output across all tokens excluding the CLS token and black end-of-sequence token).[17] This captures how aligned the representation of a word is with the sentence as a whole. If a word has both a low degree of self-similarity and intra-sentence similarity, we infer that the word has a context-specific representation that is still distinct from the other words in that sentence. If self-similarity is low but intra-sentence similarity is high, this alludes to the word simply being contextualised by aligning its representation with the other words in that sentence. We summarise these two measures in Figure 3.7 and find that, just like in Figure 3.4a, the upper layers produce more context-specific representations as seen by the lower self-similarity, and that high-frequency words are the most context-specific. This is in line with Ethayarajh (2019) who finds that stopwords, being some of the most frequently observed words in the pretraining data, have

[17] Ethayarajh (2019) average over every word-sentence combination for a given sentence, not just a single word.

	ENG	ARA	BEN	FIN	IND	KOR	RUS	SWA	TEL	Avg
BERT	68.5	58.0	43.2	58.3	67.1	12.4	53.2	71.3	48.2	51.5
PIXEL	59.6	57.3	36.3	57.1	63.6	26.1	50.5	65.9	61.7	52.3
TINY-CONTINUOUS	42.6	45.0	12.4	45.3	48.1	13.2	36.7	46.8	45.7	36.6
SMALL-CONTINUOUS	57.1	53.3	20.3	57.5	62.9	22.3	51.1	65.3	58.1	48.8
Scale										
TINY-BIGRAMS	43.3	45.5	19.0	50.3	48.2	14.9	45.4	52.7	56.4	41.6
SMALL-BIGRAMS	50.8	53.2	37.1	59.1	57.5	20.1	52.8	62.4	64.2	50.8
BASE-BIGRAMS	53.8	53.1	46.5	59.6	60.3	18.8	54.1	64.1	65.7	52.8

Table 3.6: Validation set F_1 scores for TyDiQA-GoldP. Average (Avg) scores exclude ENG (Clark et al., 2020). With some rendering structures, answer span extraction adversely affects results (see discussion at § 3.9.4).

	AMH	HAU	IBO	KIN	LUG	LUO	PCM	SWA	WOL	YOR	Avg
BERT	0	86.6	83.5	72.0	78.4	73.2	87.0	83.3	62.2	73.8	62.7
PIXEL	47.7	82.4	79.9	64.2	76.5	66.6	78.7	79.8	59.7	70.7	70.6
BASE-BIGRAMS	50.1	85.6	82.2	68.4	78.4	72.5	82.8	82.4	64.4	74.8	74.2

Table 3.7: Test set F_1 scores on MasakhaNER (Adelani et al., 2021). We follow the implementation of Rust et al. (2023) and render each word at the start of a new image patch.

some of the most context-specific representations. The measure of intra-sentence similarity reveals that the contextualised representation of low-frequency words is more similar to that of its context, with high-frequency words having more nuance where words do not necessarily mean the same just because they appear in the same sentence.

Chapter 4

Pixel-Based Language Modeling of Historical Documents

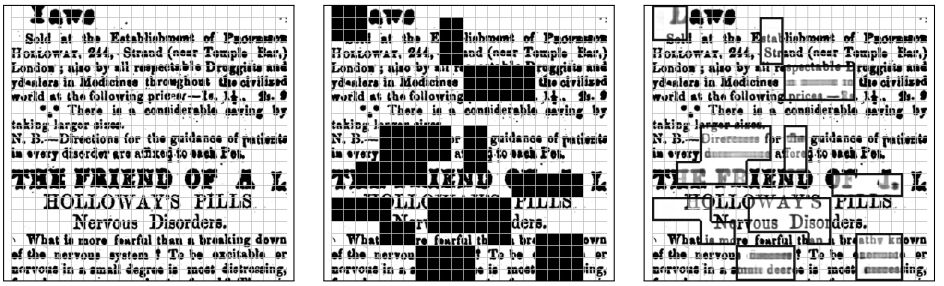
The work presented in this chapter is based on a paper that has been published as: Nadav Borenstein, **Phillip Rust**, Desmond Elliott, and Isabelle Augenstein. 2023b. PHD: Pixel-based language modeling of historical documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 87–107, Singapore. Association for Computational Linguistics.

Abstract

The digitisation of historical documents has provided historians with unprecedented research opportunities. Yet, the conventional approach to analysing historical documents involves converting them from images to text using OCR, a process that overlooks the potential benefits of treating them as images and introduces high levels of noise. To bridge this gap, we take advantage of recent advancements in pixel-based language models trained to reconstruct masked patches of pixels instead of predicting token distributions. Due to the scarcity of real historical scans, we propose a novel method for generating synthetic scans to resemble real historical documents. We then pre-train our model, PHD, on a combination of synthetic scans and real historical newspapers from the 1700-1900 period. Through our experiments, we demonstrate that PHD exhibits high proficiency in reconstructing masked image patches and provide evidence of our model's noteworthy language understanding capabilities. Notably, we successfully apply our model to a historical QA task, highlighting its utility in this domain.

nadavborenstein/pixel-bw

*Warning: This chapter shows dataset samples that are racist in nature



(a) Input example. (b) Masking the input. (c) Model predictions.

Figure 4.1: Our proposed model, PHD. The model is trained to reconstruct the original image (a) from the masked image (b), resulting in (c). The grid represents the 16 × 16 pixels patches that the inputs are broken into.

4.1 Introduction

Recent years have seen a boom in efforts to digitise historical documents in numerous languages and sources (Chadwyck, 1998; Groesen, 2015; Moss, 2009), leading to a transformation in the way historians work. Researchers are now able to expedite the analysis process of vast historical corpora using NLP tools, thereby enabling them to focus on interpretation instead of the arduous task of evidence collection (Laite, 2020; Gerritsen, 2012).

The primary step in most NLP tools tailored for historical analysis involves Optical Character Recognition (OCR). However, this approach poses several challenges and drawbacks. First, OCR strips away any valuable contextual meaning embedded within non-textual elements, such as page layout, fonts, and figures.[1] Moreover, historical documents present numerous challenges to OCR systems. This can range from deteriorated pages, archaic fonts and language, the presence of non-textual elements, and occasional deficiencies in scan quality (e.g., blurriness), all of which contribute to the introduction of additional noise. Consequently, the extracted text is often riddled with errors at the character level (Robertson and Goldwater, 2018; Bollmann, 2019), which most large language models (LLMs) are not tuned to process. Token-based LLMs are especially sensitive to this, as the discrete structure of their input space cannot handle well the abundance of out-of-vocabulary words that characterise OCRed historical documents (Rust et al., 2023). Therefore, while LLMs have proven

[1]Consider, for example, the visual data that is lost by processing the newspaper page in Figure 4.18 in § 4.7.3 as text.

remarkably successful in modern domains, their performance is considerably weaker when applied to historical texts (Manjavacas and Fonteyn, 2022; Baptiste et al., 2021, *inter alia*). Finally, for many languages, OCR systems either do not exist or perform particularly poorly. As training new OCR models is laborious and expensive (Li et al., 2023c), the application of NLP tools to historical documents in these languages is limited.

This work addresses these limitations by taking advantage of recent advancements in pixel-based language modelling, with the goal of constructing a general-purpose, image-based and OCR-free language encoder of historical documents. Specifically, we adapt PIXEL (Rust et al., 2023), a language model that renders text as images and is trained to reconstruct masked patches instead of predicting a distribution over tokens. PIXEL's training methodology is highly suitable for the historical domain, as (unlike other pixel-based language models) it does not rely on a pretraining dataset composed of instances where the image and text are aligned. Figure 4.1 visualises our proposed training approach.

Given the paucity of large, high-quality datasets comprising historical scans, we pretrain our model using a combination of 1) synthetic scans designed to resemble historical documents faithfully, produced using a novel method we propose for synthetic scan generation; and 2) real historical English newspapers published in the Caribbeans in the 18th and 19th centuries. The resulting pixel-based language encoder, **PHD** (**P**ixel-based model for **H**istorical **D**ocuments), is subsequently evaluated based on its comprehension of natural language and its effectiveness in performing Question Answering from historical documents.

We discover that PHD displays impressive reconstruction capabilities, being able to correctly predict both the form and content of masked patches of historical newspapers (§ 4.4.4). We also note the challenges concerning quantitatively evaluating these predictions. We provide evidence of our model's noteworthy language understanding capabilities while exhibiting an impressive resilience to noise. Finally, we demonstrate the usefulness of the model when applied to the historical QA task (§ 4.5.4).

To facilitate future research, we provide the dataset, models, and code at nadavborenstein/pixel-bw.

4.2 Background

4.2.1 NLP for Historical Texts

Considerable efforts have been invested in improving both OCR accuracy (Li et al., 2023c; Smith, 2023) and text normalisation techniques for historical documents (Drobac et al., 2017; Robertson and Goldwater, 2018; Bollmann et al., 2018; Bollmann, 2019; Lyu et al., 2021). This has been done with the aim of aligning historical texts with their modern counterparts. However, these methods are not without flaws (Robertson and Goldwater, 2018; Bollmann, 2019), and any errors introduced during these preprocessing stages can propagate to downstream tasks (Robertson and Goldwater, 2018; Hill and Hengchen, 2019). As a result, historical texts remain a persistently challenging domain for NLP research (Lai et al., 2021; De Toni et al., 2022; Borenstein et al., 2023c). Here, we propose a novel approach to overcome the challenges associated with OCR in historical material, by employing an image-based language model capable of directly processing historical document scans and effectively bypassing the OCR stage.

4.2.2 Pixel-based Models for NLU

Extensive research has been conducted on models for processing text embedded in images. Most existing approaches incorporate OCR systems as an integral part of their inference pipeline (Appalaraju et al., 2021; Li et al., 2021; Delteil et al., 2022). These approaches employ multimodal architectures where the input consists of both the image and the output generated by an OCR system.

Recent years have also witnessed the emergence of OCR-free approaches for pixel-based language understanding. Kim et al. (2022) introduce DONUT, an image-encoder-text-decoder model for document comprehension. DONUT is pretrained with the objective of extracting text from scans, a task they refer to as "pseudo-OCR". Subsequently, it is finetuned on various text generation tasks, reminiscent of T5 (Roberts et al., 2020). While architecturally similar to DONUT, Dessurt (Davis et al., 2023) and Pix2Struct (Lee et al., 2023) were pretrained by masking image regions and predicting the text in both masked and unmasked image regions. Unlike our method, all above-mentioned models predict in the text space rather than the pixel space. This presupposes access to a pretraining dataset comprised of instances where the image and text are aligned. However, this assumption cannot hold for historical NLP since OCR-independent ground

truth text for historical scans is, in many times, unprocurable and cannot be used for training purposes.

Text-free models that operate at the pixel level for language understanding are relatively uncommon. One notable exception is Li et al. (2022a), which utilises Masked Image Modeling for pretraining on document patches. Nevertheless, their focus lies primarily on tasks that do not necessitate robust language understanding, such as table detection, document classification, and layout analysis. PIXEL (Rust et al., 2023), conversely, is a text-free pixel-based language model that exhibits strong language understanding capabilities, making it the ideal choice for our research. The subsequent section will delve into a more detailed discussion of PIXEL and how we adapt it to our task.

4.3 Model

PIXEL We base PHD on PIXEL, a pretrained pixel-based encoder of language. PIXEL has three main components: A text renderer that draws texts as images, a pixel-based encoder, and a pixel-based decoder. The training of PIXEL is analogous to BERT (Devlin et al., 2019). During pretraining, input strings are rendered as images, and the encoder and the decoder are trained jointly to reconstruct randomly masked image regions from the unmasked context. During finetuning, the decoder is replaced with a suitable classification head, and no masking is performed. The encoder and decoder are based on the ViT-MAE architecture (He et al., 2022) and work at the patch level. That is, the encoder breaks the input image into patches of 16×16 pixels and outputs an embedding for each patch. The decoder then decodes these patch embeddings back into pixels. Therefore, random masking is performed at the patch level as well.

PHD We follow the same approach as PIXEL's pretraining and finetuning schemes. However, PIXEL's intended use is to process texts, not natural images. That is, the expected input to PIXEL is a string, not an image file. In contrast, we aim to use the model to encode real document scans. Therefore, we make several adaptations to PIXEL's training and data processing procedures to make it compatible with our use case (§ 4.4 and § 4.5).

Most crucially, we alter the dimensions of the model's input: The text renderer of PIXEL renders strings as a long and narrow image with a resolution of 16×8464 pixels (corresponding to 1×529 patches), such that the resulting image resembles a ribbon with text. Each input character is set to be not taller than 16

Source	#Issues	#Train Scans	#Test Scans
Caribbean Project	7 487	1 675 172	87 721
Danish Royal Library	5 661	300 780	15 159
Total	13 148	1 975 952	102 880

Table 4.1: Statistics of the newspapers dataset.

pixels and occupies roughly one patch. However, real document scans cannot be represented this way, as they have a natural two-dimensional structure and irregular fonts, as Figure 4.1a demonstrates (and compare to Figure 4.17a in § 4.7.3). Therefore, we set the input size of PHD to be 368 × 368 pixels (or 23 × 23 patches).

4.4 Training a Pixel-Based Historical LM

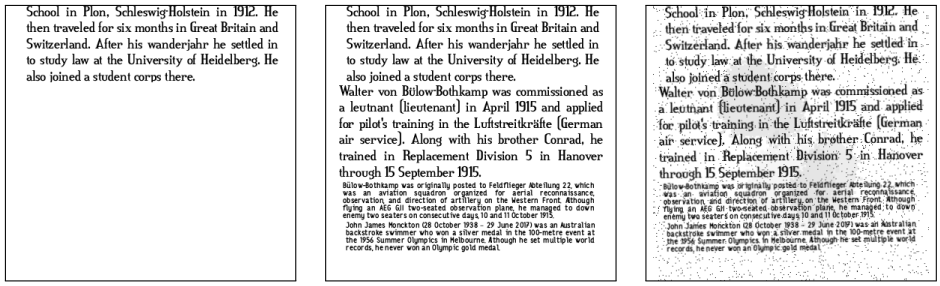
We design PHD to serve as a general-purpose, pixel-based language encoder of historical documents. Ideally, PHD should be pretrained on a large dataset of scanned documents from various historical periods and different locations. However, large, high-quality datasets of historical scans are not easily obtainable. Therefore, we propose a novel method for generating historical-looking artificial data from modern corpora (see § 4.4.1). We adapt our model to the historical domain by continuously pretraining it on a medium-sized corpus of real historical documents. Below, we describe the datasets and the pretraining process of the model.

4.4.1 Artificially Generated Pretraining Data

Our pretraining dataset consists of artificially generated scans of texts from the same sources that BERT used, namely the BookCorpus (Zhu et al., 2015) and the English Wikipedia.[2] We generate the scans as follows.

We generate dataset samples on-the-fly, adopting a similar approach as Davis et al. (2023). First, we split the text corpora into paragraphs, using the new-line character as a delimiter. From a paragraph chosen at random, we pick a random spot and keep the text spanning from that spot to the paragraph's end. We also sample a random font and font size from a pre-defined list of fonts (from

[2]We use the version "20220301.en" hosted on 🤗 datasets/wikipedia.



(a) Embedding one paragraph.

(b) Adding more paragraphs.

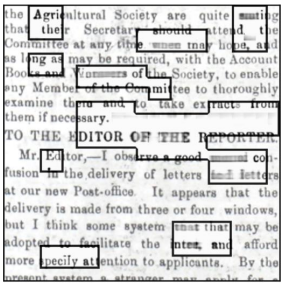
(c) Adding noise.

Figure 4.2: Process of generating a single artificial scan. Refer to § 4.4.1 for detailed explanations.

Davis et al. (2023)). The text span and the font are then embedded within an HTML template using the Python package Jinja,[3] set to generate a Web page with dimensions that match the input dimension of the model. Finally, we use the Python package WeasyPrint[4] to render the HTML file as a PNG image. Figure 4.2a visualises this process' outcome.

In some cases, if the text span is short or the selected font is small, the resulting image contains a large empty space (as in Figure 4.2a). When the empty space within an image exceeds 10%, a new image is generated to replace the vacant area. We create the new image by randomly choosing one of two options. In 80% of the cases, we retain the font of the original image and select the next paragraph. In 20% of the cases, a new paragraph and font are sampled. This pertains to the common case where a historical scan depicts a transition of context or font (e.g., Figure 4.1a). This process can repeat multiple times, resulting in images akin to Figure 4.2b.

Finally, to simulate the effects of scanning ageing historical documents, we degrade the image by adding various types of noise, such as blurring, rotations, salt-and-pepper noise and bleed-through effect (see Figure 4.2c and Figure 4.9 in § 4.7.3 for examples). § 4.7.1.2 enumerates the full list of the degradations and augmentations we use.



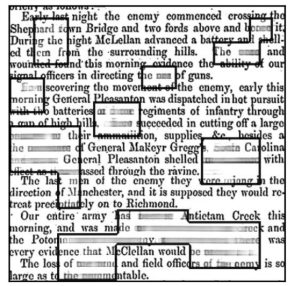


Figure 4.3: Examples of some image completions made by PHD. Masked regions marked by dark outlines.

4.4.2 Real Historical Scans

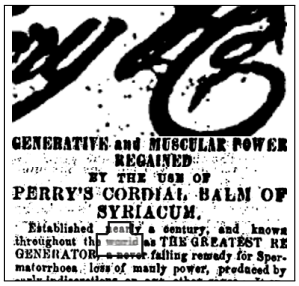
We adapt PHD to the historical domain by continuously pretraining it on a medium-sized corpus of scans of real historical newspapers. Specifically, we collect newspapers written in English from the "Caribbean Newspapers, 1718–1876" database,[5] the largest collection of Caribbean newspapers from the 18th–19th century available online. We extend this dataset with English-Danish newspapers published between 1770–1850 in the Danish Caribbean colony of Santa Cruz (now Saint Croix) downloaded from the Danish Royal Library's website.[6] See Table 4.1 for details of dataset sizes. While confined in its geographical and temporal context, this dataset offers a rich diversity in terms of content and format, rendering it an effective test bed for evaluating PHD.

Newspaper pages are converted into a 368×368 pixels crops using a sliding window approach over the page's columns. This process is described in more detail in § 4.7.1.2. We reserve 5% of newspaper issues for validation, using the rest for training. See Figure 4.10 in § 4.7.3 for dataset examples.

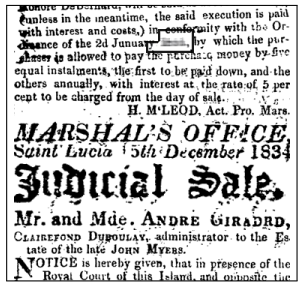
4.4.3 Pretraining Procedure

Like PIXEL, the pretraining objective of PHD is to reconstruct the pixels in masked image patches. We randomly occlude 28% of the input patches with 2D rectangular masks. We uniformly sample their width and height from $[2, 6]$ and

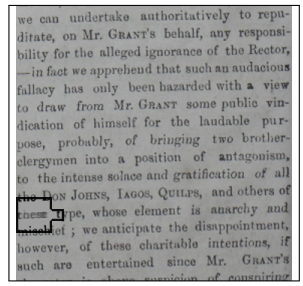
[3] https://jinja.palletsprojects.com/en/3.1.x/
[4] https://weasyprint.org/
[5] https://www.readex.com/products/caribbean-newspapers-series-1-1718-1876-american-antiquarian-society/
[6] https://www2.statsbiblioteket.dk/mediestream/



(a) world



(b) 1893



(c) every

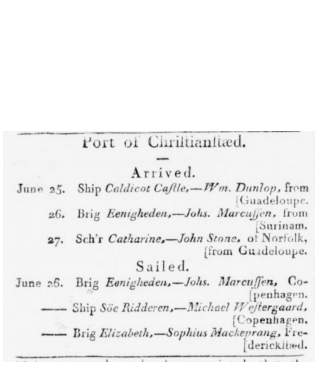
Figure 4.4: Single word completions made by our model. Figure captions depict the missing word. Fig (a) depicts a successful reconstruction, whereas Fig (b) and (c) represent fail-cases.

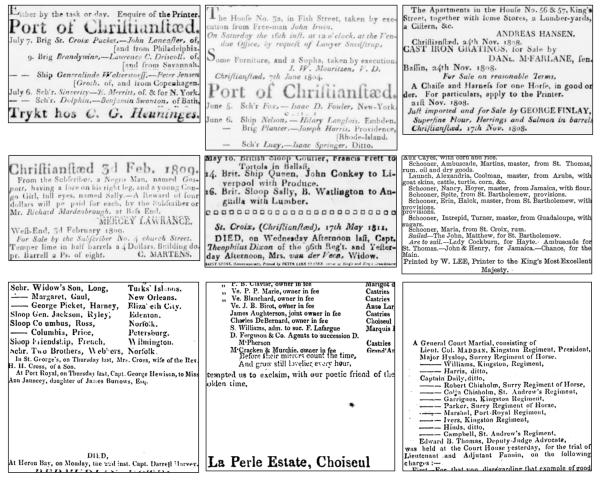
[2, 4] patches, respectively, and then place them in random image locations (See Figure 4.1b for an example). Training hyperparameters can be found in § 4.7.1.1.

4.4.4 Pretraining Results

Qualitative Evaluation We begin by conducting a qualitative examination of the predictions made by our model. Figure 4.3 presents a visual representation of the model's predictions on three randomly selected scans from the test set of the Caribbean newspapers dataset (for additional results on other datasets, refer to Figure 4.12, § 4.7.3). From a visual inspection, it becomes evident that the model accurately reconstructs the fonts and structure of the masked regions. However, the situation is less clear when it comes to predicting textual content. Similar to Rust et al. (2023), unsurprisingly, prediction quality is high and the results are sharp for smaller masks and when words are only partially obscured. However, as the completions become longer, the text quality deteriorates, resulting in blurry text. It is important to note that evaluating these blurry completions presents a significant challenge. Unlike token-based models, where the presence of multiple words with high, similar likelihood can easily be detected by examining the discrete distribution, this becomes impossible with pixel-based models. In pixel-based completions, high-likelihood words may overlay and produce a blurry completion. Clear completions are only observed when a single word has a significantly higher probability compared to others. This limitation is an area that we leave for future work.

We now move to analyse PHD's ability to fill in single masked words. We





(a) Semantic search target. (b) Retrieved scans.

Figure 4.5: Semantic search using our model. (a) is the target of the search, and (b) are scans retrieved from the newspaper corpus.

randomly sample test scans and OCRed them using Tesseract.[7] Next, we randomly select a single word from the OCRed text and use Tesseract's word-to-image location functionality to (heuristically) mask the word from the image. Results are presented in Figure 4.4. Similar to our earlier findings, the reconstruction quality of single-word completion varies. Some completions are sharp and precise, while others appear blurry. In some few cases, the model produces a sharp reconstruction of an incorrect word (Figure 4.4c). Unfortunately, due to the blurry nature of many of the results (regardless of their correctness), a quantitative analysis of these results (e.g., by OCRing the reconstructed patch and comparing it to the OCR output of the original patch) is unattainable.

Semantic Search A possible useful application of PHD is semantic search. That is, searching in a corpus for historical documents that are semantically similar to a concept of interest. We now analyse PHD's ability to assign similar historical scans with similar embeddings. We start by taking a random sample of 1000 images from our test set and embed them by averaging the patch embeddings of the final layer of the model. We then reduce the dimensionality of the embeddings with t-SNE (van der Maaten and Hinton, 2008). Upon visual inspection (Figure 4.13

[7] tesseract-ocr/tesseract

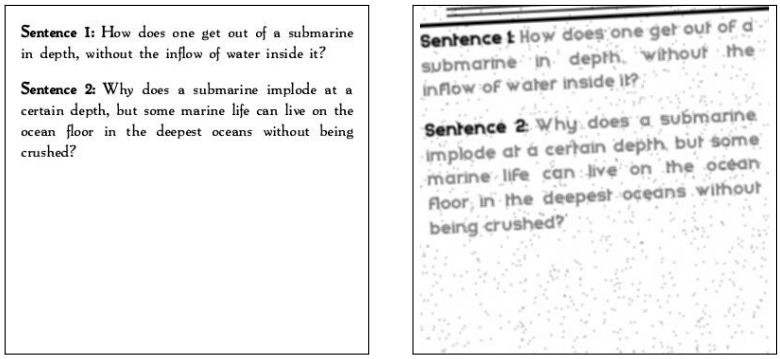


Figure 4.6: Samples from the clean and noisy visual GLUE datasets.

in § 4.7.3), we see that scans are clustered based on visual similarity and page structure.

Figure 4.13, however, does not provide insights regarding the semantic properties of the clusters. Therefore, we also directly use the model in semantic search settings. Specifically, we search our newspapers corpus for scans that are semantically similar to instances of the *Runaways Slaves in Britain* dataset, as well as scans containing shipping ads (See Figure 4.16 in § 4.7.3 for examples). To do so, we embed 1M random scans from the corpus. We then calculate the cosine similarity between these embeddings and the embedding of samples from the *Runaways Slaves in Britain* and embeddings of shipping ads. Finally, we manually examine the ten most similar scans to each sample.

Our results (Figure 4.5 and Figure 4.14 in § 4.7.3) are encouraging, indicating that the embeddings capture not only structural and visual information, but also the semantic content of the scans. However, the results are still far from perfect, and many retrieved scans are not semantically similar to the search's target. It is highly plausible that additional specialised finetuning (e.g., SentenceBERT's (Reimers and Gurevych, 2019) training scheme) is necessary to produce more semantically meaningful embeddings.

4.5 Training for Downstream NLU Tasks

After obtaining a pretrained pixel-based language model adapted to the historical domain (§ 4.4), we now move to evaluate its understanding of natural language and its usefulness in addressing historically-oriented NLP tasks. Below, we describe

Figure 4.7: Example from the *Runaways Slaves in Britain* dataset, rendered as visual question answering task. The gray overlay marks the patches containing the answer.

the datasets we use for this and the experimental settings.

4.5.1 Language Understanding

We adapt the commonly used GLUE benchmark (Wang et al., 2018) to gauge our model's understanding of language. We convert GLUE instances into images similar to the process described in § 4.4.1. Given a GLUE instance with sentences s_1, s_2 (s_2 can be empty), we embed s_1 and s_2 into an HTML template, introducing a line break between the sentences. We then render the HTML files as images.

We generate two versions of this visual GLUE dataset – clean and noisy. The former is rendered using a single pre-defined font without applying degradations or augmentations, whereas the latter is generated with random fonts and degradations. Figure 4.6 presents a sample of each of the two dataset versions. While the first version allows us to measure PHD's understanding of language in "sterile" settings, we can use the second version to estimate the robustness of the model to noise common to historical scans.

4.5.2 Historical Question Answering

QA applied to historical datasets can be immensely valuable and useful for historians (Borenstein et al., 2023a). Therefore, we assess PHD's potential for assisting historians with this important NLP task. We finetune the model on two novel datasets. The first is an adaptation of the classical SQuAD-v2 dataset (Rajpurkar et al., 2016), while the second is a genuine historical QA dataset.

Noise	Images	Model	MNLI 393k	QQP 364k	QNLI 105k	SST-2 67k	COLA 8.6k	STS-B 5.8k	MRPC 3.7k	RTE 2.5k	WNLI 635	AVG
✗	✗	BERT	**84.1**	**87.6**	**91.0**	**92.6**	**60.3**	**88.8**	**90.2**	**69.5**	51.8	**80.0**
		PIXEL	78.5	84.5	87.8	89.6	38.4	81.1	88.2	60.5	53.8	74.1
	✓	CLIP_{lin}	50.2	64.7	67.4	79.8	4.2	56.4	74.1	51.5	25.6	52.7
		DONUT	64.0	77.8	69.7	82.1	13.9	14.4	81.7	54.0	**57.7**	57.2
		Ours	70.1	82.7	82.3	82.5	15.9	80.2	83.4	59.9	54.1	67.9
✓	✓	OCR+BERT	**71.7**	77.5	**82.7**	**85.5**	**39.7**	68.4	**86.9**	58.8	51.3	**69.2**
		OCR+PIXEL	70.6	78.5	81.5	83.6	30.3	68.8	84.7	**59.7**	58.6	68.5
		CLIP_{lin}	45.3	67.4	64.4	79.2	3.5	57.9	78.8	47.3	32.7	52.9
		DONUT	61.6	74.1	75.1	75.5	10.2	20.6	81.9	56.7	**60.0**	57.3
		Ours	68.0	**80.4**	81.8	83.9	15.1	**80.4**	83.6	58.5	57.8	67.2

Table 4.2: Results for PHD finetuned on GLUE. The metrics are F_1 score for QQP and MRPC, Matthew's correlation for COLA, Spearman's ρ for STS-B, and accuracy for the remaining datasets. Bold values indicate the best model in category (noisy/clean), while underscored values indicate the best pixel-based model.

SQuAD Dataset We formulate SQuAD-v2 as a patch classification task, as illustrated in Figure 4.11 in § 4.7.3. Given a SQuAD instance with question q, context c and answer a that is a span in c, we render c as an image, I (Figure 4.11a). Then, each patch of I is labelled with 1 if it contains a part of a or 0 otherwise. This generates a binary label mask M for I, which our model tries to predict (Figure 4.11b). If any degradations or augmentations are later applied to I, we ensure that M is affected accordingly. Finally, similarly to Lee et al. (2023), we concatenate to I a rendering of q and crop the resulting image to the appropriate input size (Figure 4.11c).

Generating the binary mask M is not straightforward, as we do not know where a is located inside the generated image I. For this purpose, we first use Tesseract to OCR I and generate $\hat{c}$. Next, we use fuzzy string matching to search for a within $\hat{c}$. If a match $\hat{a} \in \hat{c}$ is found, we use Tesseract to find the pixel coordinates of $\hat{a}$ within I. We then map the pixel coordinates to patch coordinates and label all the patches containing $\hat{a}$ with 1. In about 15% of the cases, Tesseract fails to OCR I properly, and $\hat{a}$ cannot be found in $\hat{c}$, resulting in a higher proportion of SQuAD samples without an answer compared to the text-based version.

As with GLUE, we generate two versions of visual SQuAD, which we use to evaluate PHD's performance in both sterile and historical settings.

Historical QA Dataset Finally, we finetune PHD for a real historical QA task. For this, we use the English dataset scraped from the website of the *Runaways Slaves in Britain* project, a searchable database of over 800 newspaper adverts printed between 1700 and 1780 placed by enslavers who wanted to capture enslaved people who had self-liberated (Newman et al., 2019). Each ad was manually transcribed and annotated with more than 50 different attributes, such as the described gender and age, what clothes the enslaved person wore, and their physical description.

Following Borenstein et al. (2023a), we convert this dataset to match the SQuAD format: given an ad and an annotated attribute, we define the transcribed ad as the context c, the attribute as the answer a, and manually compose an appropriate question q. We process the resulting dataset similarly to how SQuAD is processed, with one key difference: instead of rendering the transcribed ad c as an image, we use the original ad scan. Therefore, we also do not introduce any noise to the images. See Figure 4.7 for an example instance. We reserve 20% of the dataset for testing.

4.5.3 Training Procedure

Similar to BERT, PHD is finetuned for downstream tasks by replacing the decoder with a suitable head. Table 4.4 in § 4.7.1.1 details the hyperparameters used to train PHD on the different GLUE tasks. We use the standard GLUE metrics to evaluate our model. Since GLUE is designed for models of modern English, we use this benchmark to evaluate a checkpoint of our model obtained after training on the artificial modern scans, but before training on the real historical scans. The same checkpoint is also used to evaluate PHD on SQuAD. Conversely, we use the final model checkpoint (after introducing the historical data) to finetune on the historical QA dataset: First, we train the model on the noisy SQuAD and subsequently finetune it on the *Runaways* dataset (see § 4.7.1.1 for training details).

To evaluate our model's performance on the QA datasets, we employ various metrics. The primary metrics include binary accuracy, which indicates whether the model agrees with the ground truth regarding the presence of an answer in the context. Additionally, we utilise patch-based accuracy, which measures the ratio of overlapping answer patches between the ground truth mask M and the predicted mask $\hat{M}$, averaged over all the dataset instances for which an answer exists. Finally, we measure the number of times a predicted answer and the

Task	Model	Noise / Image	Binary acc	Patch acc	One Overlap
S	BERT	✗/ ✗	72.3	47.3	53.9
	Ours	✗/ ✓	60.3	16.4	42.2
	Ours	✓/ ✓	61.7	14.4	41.2
R	BERT	- / ✗	78.3	52.0	55.8
	Ours	- / ✓	74.7	20.0	48.8

Table 4.3: Results for PHD finetuned on our visual SQuAD (S) and the *Runaways Slaves* (R) datasets.

ground truth overlap by at least a single patch. We balance the test sets to contain an equal number of examples with and without an answer.

4.5.4 Results

Baselines We compare PHD's performance on GLUE to a variety of strong baselines, covering both OCR-free and OCR-based methods. First, we use CLIP with a ViT-L/14 image encoder in the linear probe setting, which was shown to be effective in a range of settings that require a joint understanding of image and text—including rendered SST-2 (Radford et al., 2021b). While we only train a linear model on the extracted CLIP features, compared to full finetuning in PHD, CLIP is about 5× the size with ~427M parameters and has been trained longer on more data. Second, we finetune DONUT (§ 4.2.2), which has ~200M parameters and is the closest and strongest OCR-free alternative to PHD. Moreover, we finetune BERT and PIXEL on the OCR output of Tesseract. Both BERT and PIXEL are comparable in size and compute budget to PHD. Although BERT has been shown to be overall more effective on standard GLUE than PIXEL, PIXEL is more robust to orthographic noise (Rust et al., 2023). Finally, to obtain an empirical upper limit to our model, we finetune BERT and PIXEL on a standard, not-OCRed version of GLUE. Likewise, for the QA tasks, we compare PHD to BERT trained on a non-OCRed version of the datasets (the *Runaways* dataset was manually transcribed). We describe all baseline setups in § 4.7.2.

GLUE Table 4.2 summarises the performance of PHD on GLUE. Our model demonstrates noteworthy results, achieving scores of above 80 for five out of the nine GLUE tasks. These results serve as evidence of our model's language

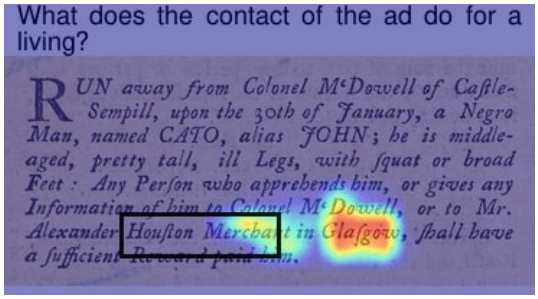
understanding capabilities. Although our model falls short when compared to text-based BERT by 13 absolute points on average, it achieves competitive results compared to the OCR-then-finetune baselines. Moreover, PHD outperforms other pixel-based models by more than 10 absolute points on average, highlighting the efficacy of our methodology.

Question Answering According to Table 4.3, our model achieves above guess-level accuracies on these highly challenging tasks, further strengthening the indications that PHD was able to obtain impressive language comprehension skills. Although the binary accuracy on SQuAD is low, hovering around 60% compared to the 72% of BERT, the relatively high "At least one overlap" score of above 40 indicates that PHD has gained the ability to locate the answer within the scan correctly. Furthermore, PHD displays impressive robustness to noise, with only a marginal decline in performance observed between the clean and noisy versions of the SQuAD dataset, indicating its potential in handling the highly noisy historical domain. The model's performance on the *Runaways Slaves* dataset is particularly noteworthy, reaching a binary accuracy score of nearly 75% compared to BERT's 78%, demonstrating the usefulness of the model in application to historically-oriented NLP tasks. We believe that the higher metrics reported for this dataset compared to the standard SQuAD might stem from the fact that *Runaways Slaves in Britain* contains repeated questions (with different contexts), which might render the task more trackable for our model.

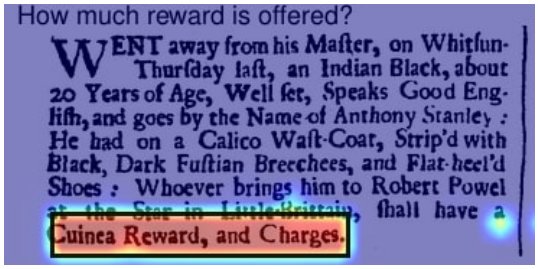
Saliency Maps Our patch-based QA approach can also produce visual saliency maps, allowing for a more fine-grained interpretation of model predictions and capabilities (Das et al., 2017). Figure 4.8 presents two such saliency maps produced by applying the model to test samples from the *Runaways Slaves in Britain* dataset, including a failure case (Figure 4.8a) and a successful prediction (Figure 4.8b). More examples can be found in Figure 4.15 in § 4.7.3.

4.6 Conclusion

In this study, we introduce PHD, an OCR-free language encoder specifically designed for analysing historical documents at the pixel level. We present a novel pretraining method involving a combination of synthetic scans that closely resemble historical documents, as well as real historical newspapers published in the Caribbeans during the 18th and 19th centuries. Through our experiments,



(a)



(b)

Figure 4.8: Saliency maps of PHD finetuned on the *Runaways Slaves in Britain* dataset. Ground truth label in a grey box. The figures were cropped in post-processing.

we observe that PHD exhibits high proficiency in reconstructing masked image patches, and provide evidence of our model's noteworthy language understanding capabilities. Notably, we successfully apply our model to a historical QA task, achieving a binary accuracy score of nearly 75%, highlighting its usefulness in this domain. Finally, we note that better evaluation methods are needed to further drive progress in this domain.

Acknowledgements

This research was partially funded by a DFF Sapere Aude research leader grant under grant agreement No 0171-00034B, the Danish-Israeli Study Foundation in Memory of Josef and Regine Nachemsohn, the Novo Nordisk Foundation (grant NNF 20SA0066568), as well as by a research grant (VIL53122) from VILLUM FONDEN. The research was also supported by the Pioneer Centre for AI, DNRF grant number P1.

Limitations

We see several limitations regarding our work. First, we focus on the English language only, a high-resource language with strong OCR systems developed for it. By doing so, we neglect low-resource languages for which our model can potentially be more impactful.

On the same note, we opted to pretrain our model on a single (albeit diverse) historical corpus of newspapers, and its robustness in handling other historical sources is yet to be proven. To address this limitation, we plan to extend our historical corpora in future research endeavours. Expanding the range of the historical training data would not only alleviate this concern but also tackle another limitation; while our model was designed for historical document analysis, most of its pretraining corpora consist of modern texts due to the insufficient availability of large historical datasets.

We also see limitations in the evaluation of PHD. As mentioned in § 4.4.4, it is unclear how to empirically quantify the quality of the model's reconstruction of masked image regions, thus necessitating reliance on qualitative evaluation. This qualitative approach may result in a suboptimal model for downstream tasks. Furthermore, the evaluation tasks used to assess our model's language understanding capabilities are limited in their scope. Considering our emphasis on historical language modelling, it is worth noting that the evaluation datasets predominantly cater to models trained on modern language. We rely on a single historical dataset to evaluate our model's performance.

Lastly, due to limited computational resources, we were constrained to training a relatively small-scale model for a limited amount of steps, potentially impeding its ability to develop the capabilities needed to address this challenging task. Insufficient computational capacity also hindered us from conducting comprehensive hyperparameter searches for the downstream tasks, restricting our ability to optimize the model's performance to its full potential. This, perhaps, could enhance our performance metrics and allow PHD to achieve more competitive results on GLUE and higher absolute numbers on SQuAD.

Parameter	MNLI	QQP	QNLI	SST-2	COLA	STS-B	MRPC	RTE	WNLI
Classification-head-pooling	Mean								
Optimizer	AdamW								
Adam β	(0.9, 0.999)								
Adam ϵ	$1e{-}8$								
Weight decay	$1e{-}5$								
Learning rate	$5e{-}2$								
Learning rate warmup steps	100								
Learning rate schedule	Cosine annealing								
Batch size	172	172	128	128	128	128	172	172	172
Max steps	10 000								
Early stopping	✓								
Eval interval (steps/epoch)	500	500	500	500	100	100	100	250	100
Dropout probability	0.0								

Table 4.4: The hyperparameters used to train PHD on GLUE tasks.

4.7 Appendix

4.7.1 Reproducibility

4.7.1.1 Training

Pretraining We pretrain PHD for 1M steps on with the artificial dataset using a batch size of 176 (the maximal batch size that fits our system) using AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with a linear warm-up over the first 50k steps to a peak learning rate of $1.5e{-}4$ and a cosine decay to a minimum learning rate of $1e{-}5$. We then train PHD for additional 100k steps with the real historical scans using the same hyperparameters but without warm-up. Pretraining took 10 days on $2 \times$ 80GB Nvidia A100 GPUs.

GLUE Table 4.4 contains the hyperparameters used to finetune PHD on the GLUE benchmark. We did not run a comprehensive hyperparameter search due to compute limitations; these settings were manually selected based on a small number of preliminary runs.

SQuAD To finetune PHD on SQuAD, we used a learning rate of $6.75e{-}6$, batch size of 128, dropout probability of 0.0 and weight decay of $1e{-}5$. We train the model for 50 000 steps.

Runaways Slaves in Britain To finetune PHD on the *Runaways Slaves in Britain* dataset, first trained the model on SQuAD using the hyperparameters mentioned above. Then, we finetuned the resulting model for an additional 1000 steps on the *Runaways Slaves in Britain*. The only hyperparameter we changed between the two runs is the dropout probability, which we increased to 0.2.

4.7.1.2 Dataset Generation

List of dataset augmentations To generate the synthetic dataset described in § 4.4.1, we applied the following transformations to the rendered images: text bleed-through effect; addition of random horizontal and lines; salt and pepper noise; Gaussian blurring; water stains effect; "holes-in-image" effect; colour jitters on image background; and random rotations.

Converting the Caribbean Newspapers dataset into 368 × 368 scans We convert full newspaper pages into a collection of 368 × 368 pixels using the following process. First, we extract the layout of the page using the Python package Eynollah.[8] This package provides the location of every paragraph on the page, as well as their reading order. As newspapers tend to be multi-columned, we "linearise" the page into a single-column document. We crop each paragraph and resize it such that its width equals 368 pixels. We then concatenate all the resized paragraphs with respect to their reading order to generate a long, single-column document with a width of 368 pixels. Finally, we use a sliding window approach to split the linear page into 368 × 368 crops, applying a stride of 128 pixels. We reserve 5% of newspaper issues for validation, using the rest for training. See Figure 4.10 in § 4.7.3 for dataset examples.

4.7.2 Historical GLUE Baselines

For all baselines below, we compute and average scores over 5 random initializations.

OCR + BERT/PIXEL For each GLUE task, we first generate 5 epochs of noisy training data and run Tesseract on it to obtain noisy text datasets. Similarly, however without oversampling, we obtain noisy versions of our fixed validation sets. We then finetune BERT-base and PIXEL-base in the same way as Rust et al. (2023), with one main difference: the noisy OCR output prevents us from

[8] qurator-spk/eynollah

separating the first and second sentence in sentence-level tasks. Therefore we treat each sentence pair as a single sequence and leave it for the models to identify sentence boundaries itself, similar to how PHD has to identify sentence boundaries in the images. We use the codebase and training setup from Rust et al. (2023).[9]

CLIP We run linear probing on CLIP using an adaptation of OpenAI's official codebase.[10] We first extract image features from the ViT-L/14 CLIP model and then train a logistic regression model with L-BFGS solver for all classification tasks and an ordinary least squares linear regression model for the regression tasks (only STS-B).

DONUT We finetune DONUT-base using an adaptation of ClovaAI's official codebase.[11] We frame each of the GLUE tasks as image-to-text tasks: the model receives the (noisy) input image and is trained to produce an output text sequence such as <s_glue><s_class><positive/></s_class></s>. In this example, taken from SST-2, the < X > tags are new vocabulary items added to DONUT and the label is an added vocabulary item for the positive sentiment class. All classification tasks in GLUE can be represented in this way. For STS-B, where the label is a floating point value denoting the similarity score between two sentences, we follow Raffel et al. (2020) to round and convert the floats into strings.[12] We finetune with batch size 32 and learning rate between $1e{-}5$ and $3e{-}5$ for a maximum of 30 epochs or 15 000 steps on images resized to a resolution of 320 × 320 pixels.

OCR-free BERT/PIXEL For GLUE, we take results reported in (Rust et al., 2021). For SQuAD, we take a BERT model finetuned on SQuAD-v2,[13] and evaluate it on the validation set of SQuAD-v2, after being balanced for the existence of an answer. For the *Runaways Slaves in Britain* dataset, we finetune a BERT-base-cased model[14] on a manually transcribed version of the dataset. We use the default SQuAD-v2 hyperparameters reported in the official Huggingface

[9] xplip/pixel
[10] openai/CLIP#linear-probe-evaluation
[11] clovaai/donut
[12] Code example in: google-research/text-to-text-transfer-transformer/blob/main/t5/data/preprocessors.py#L816-L855
[13] deepset/bert-base-cased-squad2
[14] bert-base-cased

repository for training on SQuAD-v2.[15] We then evaluate the model on a balanced test set, containing 20% of the ads.

4.7.3 Additional Material

Figure 4.9 additional examples from our artificially generated dataset.
Figure 4.10 Sample scans from the real historical dataset, as described in § 4.4.2.
Figure 4.11 The process of generating the *Visual SQuAD* dataset. We first render the context as an image (a), generate a patch-level label mask highlighting the answer (b), add noise and concatenate the question (c).
Figure 4.12 Additional examples of PHD's completions over test set samples.
Figure 4.13 Dimensionality reduction of embedding calculated by our model on historical scans. We see that scans are clustered based on visual similarity and page structure. However, further investigation is required to determine whether scans are also clustered based on semantic similarity.
Figure 4.14 Using PHD for semantic search. Figure 4.14a and is the target of the search (the concept we are looking for), while Figure 4.14b and are the retrieved scans.
Figure 4.15 Additional examples of PHD's saliency maps for samples from the test set of the *Runaways Slaves in Britain* dataset.
Figure 4.16 Examples of shipping ads Newspapers. Newspapers in the Caribbean region routinely reported on passenger and cargo ships porting and departing the islands. These ads are usually well-structured and contain information such as relevant dates, the ship's captain, route, and cargo.
Figure 4.17 Input samples for PIXEL. The images are rolled, i.e., the actual input resolution is 16×8464 pixels. The grid represents the 16×16 patches that the inputs are broken into.
Figure 4.18 An example of a full newspaper page downloaded from the "Caribbean project."

[15] https://colab.research.google.com/github/huggingface/notebooks/blob/master/examples/question_answering.ipynb

Figure 4.9: Samples of our artificially generated dataset, and compare to Figure 4.10.

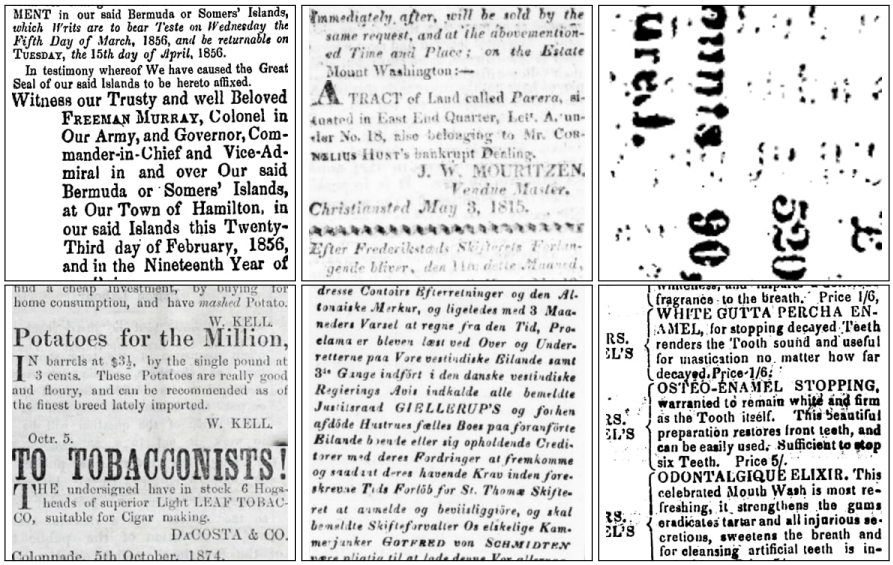
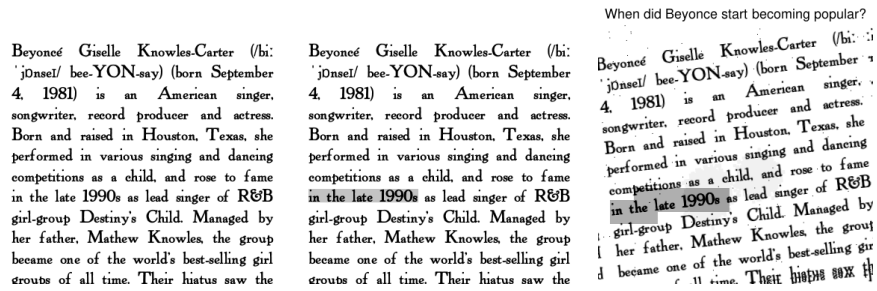


Figure 4.10: Sample scans from the real historical dataset.



(a) Rendering context c as an image I.

(b) Generating a label mask M.

(c) Adding q and degradations.

Figure 4.11: Process of generating the *Visual SQuAD* dataset. We first render the context as an image (a), generate a patch-level label mask highlighting the answer (b), add noise and concatenate the question (c).

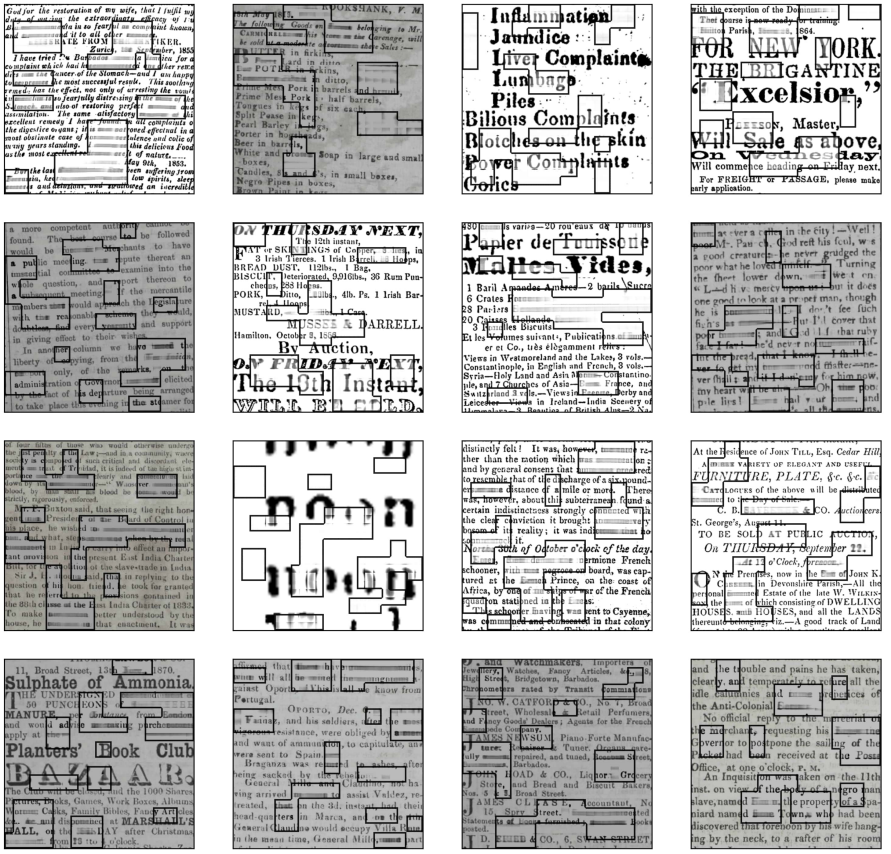


Figure 4.12: Additional examples of PHD's completions.

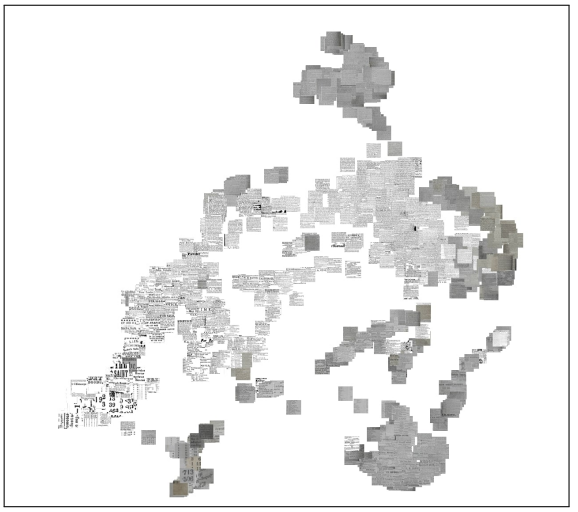


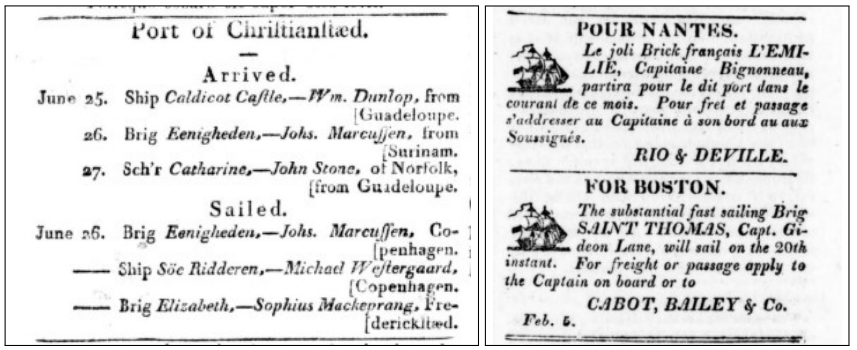
Figure 4.13: Dimensionality reduction of embedding calculated by our model on historical scans.

(a) Semantic search target.

(b) Retrieved scans.

Figure 4.14: Semantic search using our model. (a) is the target of the search, and (b) are scans retrieved from the newspaper corpus.

Figure 4.15: Additional examples of PHD's saliency maps for samples from the test set of the *Runaways Slaves in Britain* dataset.



Port of Christianstæd.

Arrived.

June 25. Ship *Caldicot Castle*,—*Wm. Dunlop*, from [Guadeloupe.
26. Brig *Eenigheden*,—*Johs. Marcussen*, from [Surinam.
27. Sch'r *Catharine*,—*John Stone*, of Norfolk, [from Guadeloupe.

Sailed.

June 26. Brig *Eenigheden*,—*Johs. Marcussen*, Co- [penhagen.
—— Ship *Söe Ridderen*,—*Michael Westergaard*, [Copenhagen.
—— Brig *Elizabeth*,—*Sophius Mackeprang*, Fre- [dericksted.

POUR NANTES.

Le joli Brick français L'EMILIE, Capitaine Bignonneau, partira pour le dit port dans le courant de ce mois. Pour fret et passage s'addresser au Capitaine à son bord au aux Soussignés.

RIO & DEVILLE.

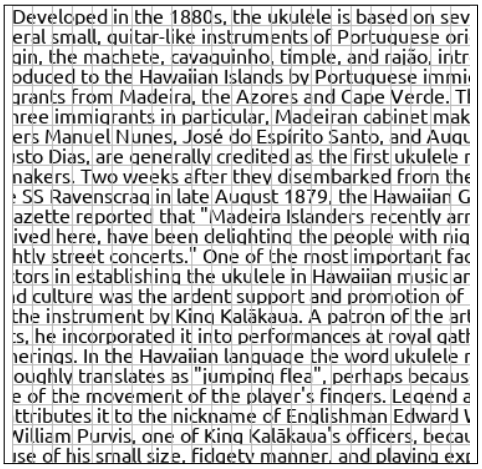
FOR BOSTON.

The substantial fast sailing Brig SAINT THOMAS, Capt. Gideon Lane, will sail on the 20th instant. For freight or passage apply to the Captain on board or to

CABOT, BAILEY & Co.

Feb. 5.

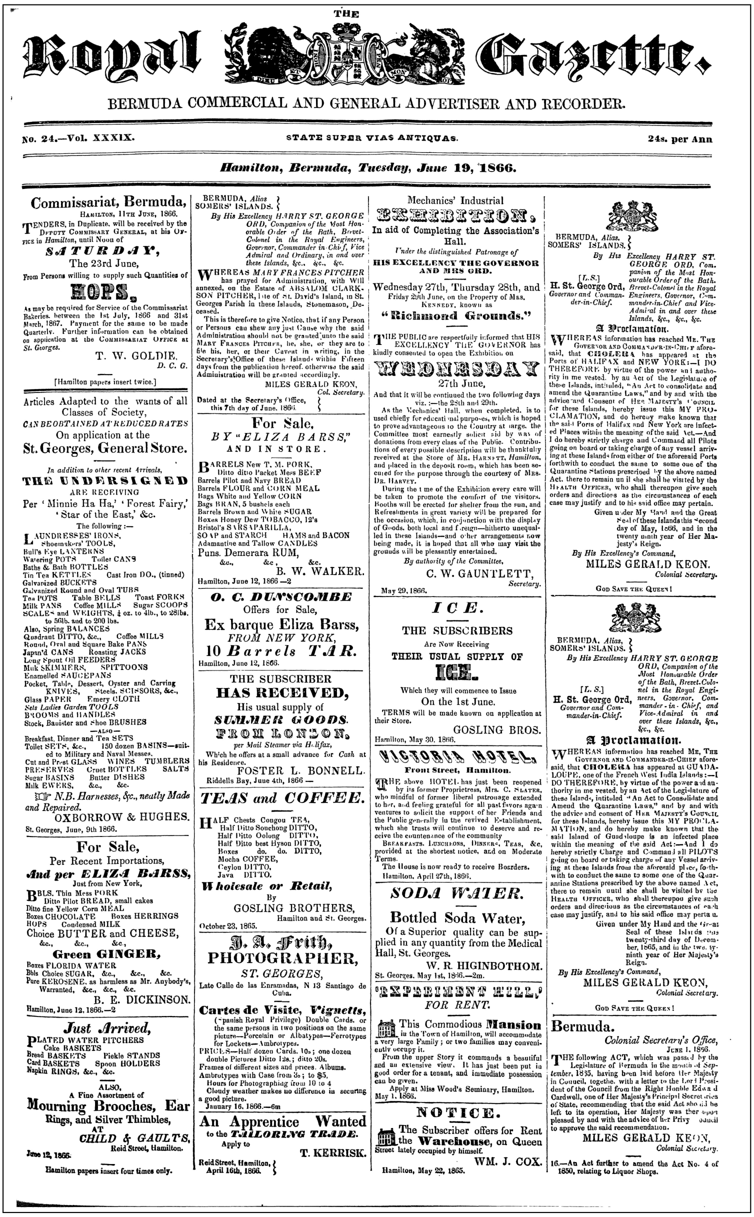
Figure 4.16: Shipping ads samples. Newspapers in the Caribbean region routinely reported on passenger and cargo ships porting and departing the islands. These ads are usually well-structured and contain information such as relevant dates, the ship's captain, route, and cargo.



(a) PIXEL's input.

(b) PIXEL's masking.

Figure 4.17: Input samples for PIXEL. The images are rolled, i.e., the actual input resolution is 16×8464 pixels. The grid represents the 16×16 patches that the inputs are broken into.



THE

Royal Gazette.

BERMUDA COMMERCIAL AND GENERAL ADVERTISER AND RECORDER.

No. 24.—Vol. XXXIX. STATE SUPER VIAS ANTIQUAS. 24s. per Ann

Hamilton, Bermuda, Tuesday, June 19, 1866.

Commissariat, Bermuda,

HAMILTON, 11TH JUNE, 1866.

TENDERS, in Duplicate, will be received by the DEPUTY COMMISSARY GENERAL, at his OFFICE in *Hamilton*, until Noon of

SATURDAY,

The 23rd June,

From Persons willing to supply such Quantities of

HOPS,

As may be required for Service of the Commissariat Bakeries, between the 1st July, 1866 and 31st March, 1867. Payment for the same to be made Quarterly. Further information can be obtained on application at the COMMISSARIAT OFFICE at *St. Georges.*

T. W. GOLDIE,
D. C. G.

[Hamilton papers insert twice.]

Articles Adapted to the wants of all Classes of Society,

CAN BE OBTAINED AT REDUCED RATES

On application at the

St. Georges, General Store.

In addition to other recent Arrivals,

THE UNDERSIGNED

ARE RECEIVING

Per 'Minnie Ha Ha,' 'Forest Fairy,' 'Star of the East,' &c.

The following :—

LAUNDRESSES' IRONS,
Shoemakers' TOOLS,
Bull's Eye LANTERNS
Watering POTS Toilet CANS
Baths & Bath BOTTLES
Tin Tea KETTLES Cast Iron DO., (tinned)
Galvanized BUCKETS
Galvanized Round and Oval TUBS
Tea POTS Table BELLS Toast FORKS
Milk PANS Coffee MILLS Sugar SCOOPS
SCALES and WEIGHTS, ¼ oz. to 4lb., to 28lbs. to 56lb. and to 200 lbs.
Also, Spring BALANCES
Quadrant DITTO, &c., Coffee MILLS
Round, Oval and Square Bake PANS
Japan'd CANS Roasting JACKS
Long Spout Oil FEEDERS
Milk SKIMMERS, SPITTOONS
Enamelled SAUCEPANS
Pocket, Table, Dessert, Oyster and Carving KNIVES, Steels, SCISSORS, &c.,
Glass PAPER Emery CLOTH
Sets Ladies Garden TOOLS
BROOMS and HANDLES
Stock, Banister and Shoe BRUSHES

—ALSO—

Breakfast, Dinner and Tea SETS
Toilet SETS, &c., 150 dozen BASINS—suited to Military and Naval Messes.
Cut and Prest GLASS WINES TUMBLERS
PRESERVES Cruet BOTTLES SALTS
Sugar BASINS Butter DISHES
Milk EWERS, &c., &c.

☞ *N.B. Harnesses, &c., neatly Made and Repaired.*

OXBORROW & HUGHES.

St. Georges, June, 9th 1866.

For Sale,

Per Recent Importations,

And per ELIZA BARSS,

Just from New York,

BBLS. Thin Mess PORK
Ditto Pilot BREAD, small cakes
Ditto fine Yellow Corn MEAL
Boxes CHOCOLATE Boxes HERRINGS
HOPS Condensed MILK
Choice BUTTER and CHEESE,
&c., &c., &c.,

Green GINGER,

Boxes FLORIDA WATER
Bbls Choice SUGAR, &c., &c., &c.
Pure KEROSENE, as harmless as Mr. Anybody's, Warranted, &c., &c., &c.

B. E. DICKINSON.

Hamilton, June 12, 1866.—2

Just Arrived,

PLATED WATER PITCHERS
Cake BASKETS
Bread BASKETS Pickle STANDS
Card BASKETS Spoon HOLDERS
Napkin RINGS, &c., &c.

ALSO,

A Fine Assortment of

Mourning Brooches, Ear Rings, and Silver Thimbles,

AT

CHILD & GAULTS,
Reid Street, Hamilton.

June 12, 1866.

Hamilton papers insert four times only.

BERMUDA, *Alias* SOMERS' ISLANDS.

By His Excellency HARRY ST. GEORGE ORD, Companion of the Most Honorable Order of the Bath, Brevet-Colonel in the Royal Engineers, Governor, Commander in Chief, Vice Admiral and Ordinary, in and over these Islands, &c., &c., &c.

WHEREAS *MARY FRANCES PITCHER* has prayed for Administration, with Will annexed, on the Estate of ABSALOM CLARKSON PITCHER, late of St. David's Island, in St. Georges Parish in these Islands, Stonemason, Deceased.

This is therefore to give Notice, that if any Person or Persons can shew any just Cause why the said Administration should not be granted unto the said MARY FRANCES PITCHER, he, she, or they are to file his, her, or their Caveat in writing, in the Secretary's Office of these Islands within Fifteen days from the publication hereof, otherwise the said Administration will be granted accordingly.

MILES GERALD KEON,
Col. Secretary.

Dated at the Secretary's Office, this 7th day of June, 1866.

For Sale,

BY "ELIZA BARSS,"

AND IN STORE.

BARRELS New T. M. PORK,
Ditto ditto Packet Mess BEEF
Barrels Pilot and Navy BREAD
Barrels FLOUR and CORN MEAL
Bags White and Yellow CORN
Bags BRAN, 5 bushels each
Barrels Brown and White SUGAR
Boxes Honey Dew TOBACCO, 12's
Bristol's SARSAPARILLA,
SOAP and STARCH HAMS and BACON
Adamantine and Tallow CANDLES
Puns. Demerara RUM,
&c., &c., &c.

B. W. WALKER.

Hamilton, June 12, 1866 —2

O. C. DUNSCOMBE

Offers for Sale,

Ex barque Eliza Barss,

FROM NEW YORK,

10 *Barrels TAR.*

Hamilton, June 12, 1866.

THE SUBSCRIBER

HAS RECEIVED,

His usual supply of

SUMMER GOODS.

FROM LONDON,

per Mail Steamer via Halifax,

Which he offers at a small advance for *Cash* at his Residence.

FOSTER L. BONNELL.

Riddells Bay, June 4th, 1866 —

TEAS and COFFEE.

HALF Chests Congou TEA,
Half Ditto Souchong DITTO,
Half Ditto Oolong DITTO,
Half Ditto best Hyson DITTO,
Boxes do. do. DITTO,
Mocha COFFEE,
Ceylon DITTO,
Java DITTO.

Wholesale or Retail,

By

GOSLING BROTHERS,
Hamilton and St. Georges.

October 23, 1865.

J. A. Frith,

PHOTOGRAPHER,

ST. GEORGES,

Late Calle de las Enramadas, N 13 Santiago de Cuba.

Cartes de Visite, *Vignetts,*

(Spanish Royal Privilege) Double Cards, or the same persons in two positions on the same picture—Porcelain or Albatypes—Ferrotypes for Lockets—Ambrotypes.

PRICES—Half dozen Cards, 10s; one dozen double Pictures Ditto 12s.; ditto 20s.

Frames of different sizes and prices. Albums.

Ambrotypes with Case from 3s; to $5,

Hours for Photographing from 10 to 4

Cloudy weather makes no difference in securing a good picture.

January 16, 1866.—6m

An Apprentice Wanted

to the *TAILORING TRADE.*

Apply to

T. KERRISK.

Reid Street, Hamilton, April 16th, 1866.

Mechanics' Industrial

EXHIBITION,

In aid of Completing the Association's Hall.

Under the distinguished Patronage of

HIS EXCELLENCY THE GOVERNOR AND MRS ORD.

Wednesday 27th, Thursday 28th, and Friday 29th June, on the Property of MRS. KENNEDY, known as

"Richmond Grounds."

THE PUBLIC are respectfully informed that HIS EXCELLENCY THE GOVERNOR has kindly consented to open the Exhibition on

WEDNESDAY

27th June,

And that it will be continued the two following days viz. :—the 28th and 29th.

As the Mechanics' Hall, when completed, is to used chiefly for educational purposes, which is hoped to prove advantageous to the Country at large, the Committee most earnestly solicit aid by way of donations from every class of the Public. Contributions of every possible description will be thankfully received at the Store of MR. HARNETT, *Hamilton*, and placed in the deposit room, which has been secured for the purpose through the courtesy of MRS. DR. HARVEY.

During the time of the Exhibition every care will be taken to promote the comfort of the visitors. Booths will be erected for shelter from the sun, and Refreshments in great variety will be prepared for the occasion, which, in conjunction with the display of Goods, both local and foreign—hitherto unequalled in these Islands—and other arrangements now being made, it is hoped that all who may visit the grounds will be pleasantly entertained.

By authority of the Committee,

C. W. GAUNTLETT,
Secretary.

May 29, 1866.

ICE.

THE SUBSCRIBERS

Are Now Receiving

THEIR USUAL SUPPLY OF

ICE.

Which they will commence to Issue

On the 1st June.

TERMS will be made known on application at their Store.

GOSLING BROS.

Hamilton, May 30, 1866.

VICTORIA HOTEL,

Front Street, Hamilton.

THE above HOTEL has just been reopened by its former Proprietress, Mrs. C. SLATER, who mindful of former liberal patronage extended to her, and feeling grateful for all past favors again ventures to solicit the support of her Friends and the Public generally in the revived Establishment, which she trusts will continue to deserve and receive the countenance of the community

BREAKFASTS, LUNCHEONS, DINNERS, TEAS, &c, provided at the shortest notice, and on Moderate Terms.

The House is now ready to receive Boarders.

Hamilton, April 27th, 1866.

SODA WATER.

Bottled Soda Water,

Of a Superior quality can be supplied in any quantity from the Medical Hall, St. Georges.

W. R. HIGINBOTHOM.

St. Georges, May 1st, 1866.—2m.

"EXPERIMENT HILL,"

FOR RENT.

This Commodious Mansion in the Town of Hamilton, will accommodate a very large Family; or two families may conveniently occupy it.

From the upper Story it commands a beautiful and an extensive view. It has just been put in good order for a tenant, and immediate possession can be given.

Apply at Miss Wood's Seminary, Hamilton.

May 1, 1866.

NOTICE.

The Subscriber offers for Rent the Warehouse, on Queen Street lately occupied by himself.

WM. J. COX.

Hamilton, May 22, 1865.

BERMUDA, *Alias.* SOMERS' ISLANDS.

By His Excellency HARRY ST. GEORGE ORD, Companion of the Most Honourable Order of the Bath, Brevet-Colonel in the Royal Engineers, Governor, Commander-in-Chief and Vice-Admiral in and over these Islands, &c., &c., &c.

[*L.S.*]
H. St. George Ord,
Governor and Commander-in-Chief.

A Proclamation.

WHEREAS information has reached ME, THE GOVERNOR AND COMMANDER-IN-CHIEF aforesaid, that CHOLERA has appeared at the Ports of HALIFAX and NEW YORK:—I DO THEREFORE by virtue of the power and authority in me vested, by an Act of the Legislature of these Islands, intituled, "An Act to consolidate and amend the Quarantine Laws," and by and with the advice and Consent of HER MAJESTY'S COUNCIL for these Islands, hereby issue this MY PROCLAMATION, and do hereby make known that the said Ports of Halifax and New York are infected Places within the meaning of the said Act.—And I do hereby strictly charge and Command all Pilots going on board or taking charge of any vessel arriving at these Islands from either of the aforesaid Ports forthwith to conduct the same to some one of the Quarantine Stations prescribed by the above named Act, there to remain until she shall be visited by the HEALTH OFFICER, who shall thereupon give such orders and directions as the circumstances of each case may justify and to his said office may pertain.

Given under My Hand and the Great Seal of these Islands this Second day of May, 1866, and in the twenty ninth year of Her Majesty's Reign.

By His Excellency's Command,

MILES GERALD KEON.
Colonial Secretary.

GOD SAVE THE QUEEN!

BERMUDA, *Alias,* SOMERS' ISLANDS.

By His Excellency HARRY ST. GEORGE ORD, Companion of the Most Honourable Order of the Bath, Brevet-Colonel in the Royal Engineers, Governor, Commander-in-Chief, and Vice-Admiral in and over these Islands, &c., &c., &c.

[*L. S.*]
H. St. George Ord,
Governor and Commander-in-Chief.

A Proclamation.

WHEREAS information has reached ME, THE GOVERNOR AND COMMANDER-IN-CHIEF aforesaid, that CHOLERA has appeared at GUADALOUPE, one of the French West India Islands:—I DO THEREFORE, by virtue of the power and authority in me vested, by an Act of the Legislature of these Islands, intituled "An Act to Consolidate and Amend the Quarantine Laws," and by and with the advice and consent of HER MAJESTY'S COUNCIL for these Islands, hereby issue this MY PROCLAMATION, and do hereby make known that the said Island of Guadaloupe is an infected place within the meaning of the said Act:—And I do hereby strictly Charge and Command all PILOTS going on board or taking charge of any Vessel arriving at these Islands from the aforesaid place, forthwith to conduct the same to some one of the Quarantine Stations prescribed by the above named Act, there to remain until she shall be visited by the HEALTH OFFICER, who shall thereupon give such orders and directions as the circumstances of each case may justify, and to his said office may pertain.

Given under My Hand and the Great Seal of these Islands this twenty-third day of December, 1865, and in the twenty-ninth year of Her Majesty's Reign.

By His Excellency's Command,

MILES GERALD KEON,
Colonial Secretary.

GOD SAVE THE QUEEN!

Bermuda.

Colonial Secretary's Office,
JUNE 1, 1866.

THE following ACT, which was passed by the Legislature of Bermuda in the month of *September*, 1865, having been laid before Her Majesty in Council, together with a letter to the Lord President of the Council from the Right Honble Edward Cardwell, one of Her Majesty's Principal Secretaries of State, recommending that the said Act should be left to its operation, Her Majesty was thereupon pleased by and with the advice of her Privy Council to approve the said recommendation.

MILES GERALD KEON,
Colonial Secretary.

16.—An Act further to amend the Act No. 4 of 1850, relating to Liquor Shops.

Figure 4.18: An example of a full newspaper page downloaded from the "Caribbean project". § 4.4.2 details the way of processing full newspaper pages so that they can be inputted to our model.

Chapter 5

Differential Privacy, Linguistic Fairness, and Training Data Influence in Multilingual Language Models

The work presented in this chapter is based on a paper that has been published as: **Phillip Rust** and Anders Søgaard. 2023. Differential privacy, linguistic fairness, and training data influence: Impossibility and possibility theorems for multilingual language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29354–29387. PMLR.

Abstract

Language models such as mBERT, XLM-R, and BLOOM aim to achieve multilingual generalization or compression to facilitate transfer to a large number of (potentially unseen) languages. However, these models should ideally also be private, linguistically fair, and transparent by relating their predictions to training data. Can these requirements be simultaneously satisfied? We show that multilingual compression and linguistic fairness are compatible with differential privacy, but that differential privacy is at odds with training data influence sparsity, an objective for transparency. We further present a series of experiments on two common NLP tasks and evaluate multilingual compression and training data influence sparsity under different privacy guarantees, exploring these trade-offs in more detail. Our results suggest that we need to develop ways to jointly optimize for these objectives in order to find practical trade-offs.

xplip/multilingual-lm-objectives

5.1 Introduction

One of the open challenges in AI is bridging the widening digital language divide by providing technologies that work well for all languages. Multilingual language models such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020a), and BLOOM (Scao et al., 2022), facilitate transfer between closely related languages, enabling roll-out of technologies for low-resource languages, and are used for a wide range of real-world applications in many languages—e.g., from named entity recognition (Khalifa et al., 2021) to legal document classification (Wang and Banko, 2021). Generalization across languages is challenged by typological divides, language families, or scripts (Singh et al., 2019; Dufter and Schütze, 2020) and finding architectures that best facilitate such transfer, achieving optimal **multilingual compression** (Ravishankar and Søgaard, 2021) through parameter sharing (rather than compartmentalization), remains an open research problem.

With the widespread adaptation of multilingual language models also comes responsibility and requirements that models are trustworthy (Pruksachatkun et al., 2021). What does trustworthiness amount to for multilingual language models? A crucial requirement is that multilingual NLP models perform equally well across languages, not favoring any languages over others. Choudhury and Deshpande (2021) refer to this property as **linguistic fairness**. Linguistic fairness is defined as zero variance across language-specific losses, typically estimated on held-out data.[1]

Another crucial requirement is *transparency*, i.e., the ability to say *why* models make particular predictions. Methods to achieve transparency come in two flavors; Some methods—commonly referred to as feature attribution methods—present rationales behind predictions in terms of input token attributions, but such rationales are limited in that they cannot explain predictions motivated by the absence of input tokens or the presence of particular token combinations. Feature attribution methods have also been shown to be unreliable (Kindermans et al., 2019; Arun et al., 2020). Other methods highlight training data influence, i.e., provide influential data points as rationales for decisions. Often referred to as instance-based interpretability methods, they are argued to be more useful across different NLP tasks (Han et al., 2020; Han and Tsvetkov, 2021; Zhou et al., 2021b).

[1]This definition of linguistic fairness is an instantiation of *equal risk fairness* or overall performance parity, i.e., equal model performance across groups (Berk et al., 2018; Verma and Rubin, 2018; Williamson and Menon, 2019), which balances precision-based and recall-based metrics and is considered more relevant than calibration-based metrics for standard NLP applications. Since the three are mutually exclusive (Miconi, 2017), we ignore calibration and balance precision and recall.

We refer to the objective of achieving sparse training data influence, i.e., strong instance-interpretability, as **training data influence sparsity**. Finally, for many NLP applications, we further need our models to be private, for which **differential privacy** (DP; Dwork, 2006) provides a theoretically rigorous framework.

The trustworthiness objectives as defined above have primarily been considered in a monolingual context, and are often (falsely) assumed to be independent (Ruder et al., 2022).[2] Our paper investigates *the extent to which these objectives align or are at odds*. We do so in a multilingual setting and show how multilinguality presents options and challenges.[3] Our theoretical contributions show that while privacy and linguistic fairness are compatible through multilingual compression, privacy and training data influence sparsity are not, and our empirical results indicate that these objectives interact in non-linear ways.

Contributions We begin (in § 5.2) with a theoretical exploration of differential privacy, training data influence, and linguistic fairness in the context of multilingual language models. We show that differential privacy and training data influence sparsity are fundamentally at odds, a result which is not limited to the multilingual setting. While differential privacy and fairness are often said to be at odds, we also show that differential privacy and linguistic fairness over languages are compatible in the multilingual setting, as a result of compression.

Subsequently (in § 5.3–§ 5.5), we present empirical results on the impact of differentially private fine-tuning on multilingual compression and training data influence: We analyze the effect of such fine-tuning on the multilingual compression of large LMs and find that it is possible to achieve (i) high compression with strong privacy at the cost of performance; (ii) high compression with high performance at the cost of privacy; or (iii) privacy and accuracy at the cost of compression. Since we show in § 5.2 that performance, privacy and compression *are theoretically* compatible, this leaves us with an open problem: How do we practically optimize for both performance, privacy and compression?

Furthermore, we compare four (proxy) metrics for quantifying multilingual compression—sentence retrieval, centered kernel alignment (CKA; Kornblith

[2]One exception is a growing body of work showing fairness and differential privacy are at odds (Bagdasaryan et al., 2019; Cummings et al., 2019; Chang and Shokri, 2021; Hansen et al., 2024). While Naidu et al. (2021) show that differential privacy and GradCAM (Selvaraju et al., 2019), a feature attribution method, are compatible, the interaction between differential privacy and training data influence remains unexplored.

[3]We are, to the best of our knowledge, first to consider differential privacy in a multilingual setting specifically, with the exception of work on differentially private neural machine translation (Kim et al., 2021).

et al., 2019), IsoScore (Rudman et al., 2022), representational similarity analysis (RSA; Kriegeskorte et al., 2008; Edelman, 1998)—and discuss their usefulness for balancing these trade-offs.

Finally, we show that LMs exhibiting high multilingual compression are less instance-interpretable in that they make highlighting training data influence more difficult.

In sum, our work shows that *linguistically fair and private high-performance multilingual models are possible, even if learning them is challenging. However, training data influence methods will fail for such models*.

5.2 Theoretical Exploration

We consider language model learning and fine-tuning in a multilingual setting, in which our training data $D = D_1 \cup \ldots \cup D_{|L|}$ is the union of disjoint training data from $|L|$ different languages. We consider the interaction of differential privacy, training data influence and linguistic fairness with performance and compression in this setting.

Preliminaries We briefly introduce our formal definitions here: A randomized algorithm, here model, $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{Y}$ is ε_p-*differentially private* (Dwork, 2006) iff for all adjacent datasets $D, D' \in \mathcal{D}$ and all $Y \subset \mathcal{Y}$,

$$\mathbb{P}(\mathcal{M}(D) \in Y) \leq \exp(\varepsilon_p) \cdot \mathbb{P}(\mathcal{M}(D') \in Y).^{4}$$

Adjacent means that the datasets differ by exactly one example x_{diff}.

A model $\mathcal{M}$ is said to be ε_i-*instance-interpretable*, i.e., having sparse training data influence, iff for any $D, D', D'' \in \mathcal{D}$ with $D' = D \setminus \{x_{diff}\}$, $D'' = D \setminus \{x'\}$, and $x_{diff} \neq x'$, where x_{diff} is the most influential training data point under leave-one-out influence,[5] it holds that

$$\mathbb{P}(\mathcal{M}(D) \in Y) - \mathbb{P}(\mathcal{M}(D') \in Y) > \exp(\varepsilon_i) \cdot (\mathbb{P}(\mathcal{M}(D) \in Y) - \mathbb{P}(\mathcal{M}(D'') \in Y)).$$

In other words, x_{diff} had more influence on $\mathcal{M}$ than any other data point x' by some margin $\exp(\varepsilon_i)$ (Koh and Liang, 2017).

[4] Note how standard empirical risk minimization is not private, since it is a linear combination of training samples near the decision boundary, and if D and D' differ in one of those, the classifier changes significantly.

[5] Leave-one-out here means $D' = D \setminus \{x_{diff}\}$ and is the gold standard for instance-based methods, which explains the close connection to DP where we also deal with adjacent datasets.

A model $\mathcal{M}$ is said to be fair (Williamson and Menon, 2019) if for a group partitioning $g(D) \rightarrow D_{g_1}, \ldots, D_{g_n}$ into smaller samples and for some loss function ℓ, e.g., 0-1 loss,

$$\ell(\mathcal{M}(D_{g_i})) = \ell(\mathcal{M}(D_{g_j})).$$

A model that is fair for a group partitioning by languages is said to be linguistically fair (Choudhury and Deshpande, 2021).

Finally, a model $\mathcal{M}$ exhibits perfect multilingual compression when it outputs identical representations for semantically equivalent inputs irrespective of the input language. Formally, for a pair of translation equivalent sentences, (i_j, i_q), the representations of i_j and i_q are identical at any layer l of the model, i.e.

$$\mathcal{M}^l(i_j) = \mathcal{M}^l(i_q).$$

In the following paragraphs, we discuss under what conditions DP, training data influence, linguistic fairness, and multilingual compression are at odds or are compatible, and how these conditions align with common scenarios in multilingual NLP.[6]

Differential Privacy and Training Data Influence Sparsity We first show that differential privacy and training data influence sparsity are fundamentally at odds:

Theorem 1. *A model $\mathcal{M}$ becomes less ε_i-instance-interpretable as it becomes more ε_p-differentially private, and vice-versa.*

Proof. Let $\mathbb{P}(\mathcal{M}(D) \in Y)$ be abbreviated as p, $\mathbb{P}(\mathcal{M}(D') \in Y) = \mathbb{P}(\mathcal{M}(D \setminus \{x_{diff}\} \in Y)$ be abbreviated as p_d, and let $\mathbb{P}(\mathcal{M}(D'') \in Y) = \mathbb{P}(\mathcal{M}(D \setminus \{x'\} \in Y)$ be abbreviated as p_2. Assume that $\mathcal{M}$ is ε_i-instance-interpretable and ε_p-differentially private.

If $\mathcal{M}$ is ε_p-differentially private, it holds that

$$\begin{aligned} & p && \leq \exp(\varepsilon_p) \cdot p_d \\ \Rightarrow \quad & \exp(\varepsilon_p) && \geq \frac{p}{p_d} \end{aligned} \tag{5.1}$$

[6] Differential privacy meaningfully protects any individual training example. However, sensitive information may be repeated across many training examples, so ε-DP does not necessarily prevent leakage of such information at the granularity of individual people, real-world events, etc. For example, in our multilingual setting, an attacker may still gain access to a social security number learned by the model, but they will be unable to identify whether the number was leaked in a particular language.

If $\mathcal{M}$ is also ε_i-instance-interpretable, it also holds that

$$\begin{aligned}
(i) \qquad & p - p_d > && \exp(\varepsilon_i)(p - p_2) \\
(ii) \Rightarrow \qquad & p > && \exp(\varepsilon_i)(p - p_2) + p_d \\
(iii) \Rightarrow \qquad & \frac{p}{p_d} > && \frac{\exp(\varepsilon_i)(p - p_2) + p_d}{p_d} \\
(iv) \Rightarrow \qquad & \exp(\varepsilon_p) > && \frac{\exp(\varepsilon_i)(p - p_2)}{p_d} + 1
\end{aligned} \tag{5.2}$$

Step (iv) follows from Equation 5.1. We can now see from Equation 5.2 step (iv) that ε_p increases with increasing ε_i, i.e. the model becomes less differentially private as it becomes more instance-interpretable, and vice-versa. □

This result is not limited to the multilingual setting.

Differential Privacy and Linguistic Fairness Fairness and differential privacy are occasionally at odds, as shown by Bagdasaryan et al. (2019); Cummings et al. (2019); Chang and Shokri (2021); Hansen et al. (2024),[7] but in the multilingual setting, fairness and privacy can be compatible (for the common definitions above). We first note that there is a trivial solution to obtaining differential privacy and linguistic fairness (a joint optimum), namely randomness. This simply shows that the two objectives can be simultaneously satisfied. Next, imagine a perfectly compressed multilingual language model trained on a multi-parallel dataset.

Theorem 2. *If a model $\mathcal{M}_D$ trained on parallel data from $|L| \geq 2$ languages, $D = \{\ldots, i_1, \ldots, i_{|L|}, \ldots\}$, with i_j and i_q being translation equivalents, is perfectly multilingually compressed, then it is ε_p-differentially private.*

Proof. Since $\mathcal{M}_D$ is perfectly compressed, the representation of i_j is identical to i_q at any layer l, i.e., $\mathcal{M}_D^l(i_j) = \mathcal{M}_D^l(i_q)$. This gives us strong k-anonymity (Li et al., 2012) in the representation space of $\mathcal{M}_D$, with $k = |L|$ and all dimensions as quasi-identifiers. Since k-anonymity is not obtained through a deterministic (reversible) procedure, but a randomly initialized learning procedure with random sampling, and since our attributes are randomly initialized, k-anonymization entails differential privacy in our setting.[8] $\mathcal{M}_D$, given perfect compression and convergence, is 0-differentially private, i.e., the probability distribution of $\mathcal{M}_D$ is unaffected by the removal of any single row. □

[7] Several authors have considered practical trade-offs between privacy and fairness, including Jagielski et al. (2019), Lyu et al. (2020), Pannekoek and Spigler (2021), and Liu et al. (2021b).

[8] The procedure also is not dependent on any individual input, because all individual data properties are either random (from initialization) or k-anonymous, by construction.

It follows directly from perfect compression that $\mathcal{M}_D$ is also linguistically fair because identical representations imply identical performance across languages. It is therefore an immediate corollary of the above result that a linguistically fair model can be differentially private.

While the assumptions of a perfectly compressed model and clean multi-parallel dataset rarely hold up in practice and there is no obvious way to satisfy them while maintaining utility, the practical significance of this result is a reminder that multilingual training converges toward k-anonymization, and that safe k-anonymization of the representation space, if obtained, would provide us differential privacy. In the absence of strong guarantees, increasing the number of training languages (larger k) would strengthen privacy (Li et al., 2012). Our empirical results below (§ 5.4) suggest that we can often obtain strong privacy and strong compression, but at the cost of performance.

5.3 Experimental Setup

In our experiments, we investigate the relation between the performance and multilingual compression of finetuned multilingual language models, and their privacy and training data influence. We rely on a commonly used multilingual pretrained language model, which we finetune with different levels of (ε, δ)-differential privacy on two common NLP tasks and evaluate using metrics of compression and training data influence.[9] This section presents the pretrained language model, the tasks, the training protocol, the metrics of compression and training data influence, and the evaluation procedure.

Model We use a pretrained XLM-R Base (Conneau et al., 2020a), which is a 12-layer encoder-only transformer with ~277M parameters and 250k vocabulary size trained on CC-100 (100 languages) via masked language modeling.

Tasks and Data We finetune in a zero-shot cross-lingual transfer setting for part-of-speech (POS) tagging and natural language inference (NLI). Why these tasks? First, while POS tagging is driven by lower-level syntactic features, NLI requires a higher-level understanding (Lauscher et al., 2020). Second, we can leverage *multi-parallel* corpora for multilingual fine-tuning and zero-shot cross-lingual

[9] For completeness, we explain the difference between ε-DP and (ε, δ)-DP in Appendix 5.8.2.

transfer in both tasks, which helps eliminate confounders.[10]

For POS tagging, we use the Parallel Universal Dependencies (PUD) treebank from Universal Dependencies (UD) v2.8 (Nivre et al., 2020; Zeman et al., 2021), which contains 1000 sentences parallel across 15 languages. We train in 7 of these languages (FR, IT, JA, PT, TH, TR, ZH),[11] exclude English,[12] and use the remaining 7 languages (AR, DE, ES, HI, ID, KO, RU) for validation. This split ensures that (1) we both train and evaluate on typologically diverse language samples, (2) there exist additional UD v2.8 treebanks in our validation set languages that we can harness for testing, and (3) there exist parallel sentences in our training set languages that we can harness to evaluate multilingual compression. We use the test splits of the following treebanks for testing: Arabic-PADT, German-GSD, Spanish-GSD, Hindi-HDTB, Indonesian-GSD, Korean-Kaist, and Russian-SynTagRus. Appendix Table 5.3 lists the treebanks' sizes.[13]

For NLI, we rely on the XNLI dataset (Conneau et al., 2018), which contains (premise, hypothesis, label)-triplets multi-parallel across 15 languages. We, again, train in 7 of these languages (BG, ES, FR, HI, TR, VI, ZH), exclude the original English data, and validate in the remaining 7 languages (AR, DE, EL, RU, SW, TH, UR). We train and validate our models on the original XNLI validation data (7500 examples per language), and we test the models on the original test data (15000 examples per language) in the validation set languages.

The idea to train and validate on the same sentences (in different languages) while testing on sentences from different treebanks (as we do for POS) or a different dataset split (as for XNLI) is to induce a slight distributional shift between validation and test data for the same language sample. This shift lets us evaluate the regularization strength of the gradient noise added by the DP-optimizer.

Training We employ the standard fine-tuning procedures for token classification (POS) and sequence classification (XNLI) proposed by Devlin et al. (2019). Similar to Li et al. (2022b), we use DP-AdamW (i.e., the DP-SGD algorithm (Abadi et al., 2016) applied to the AdamW optimizer with default hyperparameters (Loshchilov and Hutter, 2019; Kingma and Ba, 2015)) to train with (ε, δ)-DP.

[10] One limitation of this selection is that we only consider classification but no generative tasks, which could be worth exploring in the future.

[11] See Table 5.1 for language details.

[12] We exclude English to keep the number of languages balanced and because the combined corpus is already biased towards Indo-European with Latin scripts (see Table 5.1).

[13] Regardless of test split size, each language contributes equally to the mean accuracy reported in Figure 5.1.

We evaluate 6 different privacy budgets with $\varepsilon \in \{1, 3, 8, 15, 30, \infty\}$.[14] We set $\delta = \frac{1e-4}{|D_{train}|}$ for POS, where $|D_{train}| = 7000$ is the length of the training dataset, and $\delta = 1e{-}6$ for XNLI.[15] The noise multiplier σ corresponding to a particular (ε, δ)-budget is determined numerically before training through binary search. Our implementation builds upon the optimized Opacus (Yousefpour et al., 2021) privacy engine by Li et al. (2022b).[16,17] We use the Rényi differential privacy (RDP; Mironov, 2017; Mironov et al., 2019) accountant with conversion to (ε, δ)-DP (Canonne et al., 2020). Hyper-parameter tuning on private data—which the POS and XNLI data in our study simulate—has been shown to incur additional privacy leakage (Liu and Talwar, 2019; Papernot and Steinke, 2022). Therefore, we try to keep hyper-parameter tuning to a minimum and rely on sensible priors to select a suitable range of hyper-parameters. For POS, we find that the range of good hyper-parameters for non-private settings transfers well to private settings if we just use slightly higher learning rates. For XNLI, we select hyper-parameters such that the sampling rate matches that used by Li et al. (2022b) for NLI tasks in the GLUE benchmark (Wang et al., 2018).[18] Accordingly, we train with a maximum sequence length of 128 for 10 epochs with a total batch size of 96 for POS and 30 epochs with batch size 512 for XNLI.[19] At each privacy budget, we train models (3 random initializations each) with 6 learning rates for POS ($1e{-}4$, $3e{-}4$, $5e{-}4$, $7e{-}4$, $1e{-}5$, $5e{-}5$, $7e{-}5$, $1e{-}6$) and 3 learning rates for XNLI ($3e{-}4$, $4e{-}4$, $5e{-}4$ for private models and $9e{-}5$, $1e{-}4$, $2e{-}4$ for non-private models). Based on the validation accuracy, we then select the 5 best settings for each privacy level and task, listed in Appendix 5.8.3. The learning rate is linearly decayed after 50 warm-up steps for POS and without warm-up for XNLI. We perform gradient clipping (per-sample in private settings) with a threshold of 0.1. Weight decay is set to 0.01.

[14] $\varepsilon = \infty$ refers to the standard, non-private setting.

[15] We deliberately use a larger δ for XNLI because it turned out to be much harder to achieve convergence than for POS. Even with the looser DP bounds from $\delta = 1e - 6$, we were unable to find a hyper-parameter setting for $\varepsilon = 1$ where the finetuned model was substantially better than random guessing.

[16] `lxuechen/private-transformers`

[17] We do not use ghost clipping, their proposed technique to fit larger batches on the GPU at the cost of training time, as we can still fit sufficiently large batches on our GPUs without.

[18] The sampling rate $q = \frac{B_{\text{train}}}{|D_{\text{train}}|}$, B denoting the batch size.

[19] Note that using fixed-size batches technically breaks the privacy guarantees of RDP based on the Sampled Gaussian Mechanism (Mironov et al., 2019). We follow the convention of using fixed-size batches, avoiding potential out-of-memory GPU issues, as a proxy for the true privacy spending and performance (see (Li et al., 2022b) and Appendix D.4 in (Tramèr and Boneh, 2021)).

Quantifying Multilingual Compression We present four metrics of multilingual compression: A common proxy task to measure the quality of cross-lingual representations is sentence retrieval (Artetxe and Schwenk, 2019; Dufter and Schütze, 2020; Libovický et al., 2020; Ravishankar and Søgaard, 2021; Liu et al., 2021c; Maronikolakis et al., 2021). Dufter and Schütze (2020) quantify the degree of multilingual compression using bidirectional sentence retrieval precision as follows:[20]

$$P = \frac{1}{2m} \sum_{i=1}^{m} \mathbb{1}_{\arg\max_k R_{ik}=i} + \mathbb{1}_{\arg\max_k R_{ki}=i}. \tag{5.3}$$

Here, $R \in \mathbb{R}^{m \times m}$ denotes the matrix of cosine similarities $R_{ij} = \cos(e_i^q, e_j^r)$ between the m sub-word representations e_i^q and e_j^r from a LM at indices i and j for a set of parallel sentences in the languages q and r.[21]

Kornblith et al. (2019) propose to use linear centered kernel alignment (CKA) as a similarity index for neural network representations. It is defined as

$$\text{CKA}(X, Y) = \frac{\|Y^\mathrm{T} X\|_F^2}{\|X^\mathrm{T} X\|_F \|Y^\mathrm{T} Y\|_F}. \tag{5.4}$$

For LMs, the matrices X and Y are obtained by mean-pooling n sub-word representations at model layer l (Conneau et al., 2020b; Glavaš and Vulić, 2021). Typically, X and Y correspond to the representations from two different models for identical examples (Kornblith et al., 2019; Phang et al., 2021). We instead use the representations from a single model for a parallel sentence pair (s_q, s_r) in languages q and r as X and Y, respectively, to study the similarity of representations across languages, similar to Muller et al. (2021) and Conneau et al. (2020b). Yang et al. (2022b) also use CKA as a metric of compression.

IsoScore (Rudman et al., 2022) is an isotropy metric, computed as outlined in Appendix 5.8.4, that quantifies the degree to which a point cloud uniformly utilizes the vector space. In our context, this point cloud corresponds to the n sub-word representations of all examples in a corpus at layer l. Prior work has shown that anisotropic representation spaces, such as the embedding spaces of large LMs (Ethayarajh, 2019), suffer from so-called *representation degeneration* (Gao et al., 2019a), and that the isotropy of a model's representation space correlates with its task performance (Zhou et al., 2019; Wang et al., 2020c; Zhou

[20] Note that Dufter and Schütze (2020) also consider word alignment in their multilinguality score. We omit this task as it is not trivial to obtain ground truth alignments in our setup.

[21] The sub-word representations are taken from the LM's layer l and mean-pooled over the sequence length (excluding special tokens).

et al., 2021a; Rajaee and Pilehvar, 2021, *inter alia*). High isotropy also means languages are not compartmentalized and should therefore correlate with high compression.

Representational similarity analysis (RSA; Kriegeskorte et al., 2008; Edelman, 1998) was originally introduced in the field of cognitive neuroscience to analyze the similarity of fMRI activity patterns, but it is also applicable to neural network representations (Bouchacourt and Baroni, 2018; Chrupała, 2019; Chrupała and Alishahi, 2019; Lepori and McCoy, 2020; He et al., 2021c, *inter alia*), e.g., to analyze their similarity across languages. RSA measures the similarity between the representational geometries (i.e., the arrangement in the vector space) of two sets of representations. The representational geometry is determined through pairwise (dis)similarity metrics, and similarity is typically measured using a rank-based correlation metric such as Spearman's ρ (Diedrichsen and Kriegeskorte, 2017).

Quantifying Training Data Influence Training data influence metrics can help us gain an understanding of the inner workings of a model (Koh and Liang, 2017; Yeh et al., 2018; Charpiat et al., 2019; Koh et al., 2019; Pruthi et al., 2020; Basu et al., 2020; K and Søgaard, 2021; Zhang et al., 2021; Kong and Chaudhuri, 2021, *inter alia*). Such metrics are approximations of leave-one-out-influence. Pruthi et al. (2020) proposed a both effective and practical method, called TracInCP,[22] to compute the influence of a training example z on the model's prediction for another example z', which could be a test example or z itself (called the self-influence). The influence is computed as follows:

$$\text{TracInCP}(z, z') = \sum_{i=1}^{k} \eta_i \nabla \ell(\theta_i, z) \cdot \nabla \ell(\theta_i, z'), \tag{5.5}$$

where η_i is the learning rate and $\nabla \ell(\theta_i, z)$ is the gradient of the loss w.r.t. the model parameters θ_i and inputs z for the i-th model checkpoint. We will use TracInCP as an approximation of training data influence in our experiments.

Evaluation We evaluate our models both during and after fine-tuning. For POS, we evaluate every 100 steps, and for XNLI, every 200 steps. We measure zero-shot cross-lingual transfer performance on the validation and test data by accuracy (token-level for POS and sequence-level for XNLI). To account for randomness,

[22] "CP" stands for checkpoint; the method approximates TracInIdeal, which is impractical to compute, through model checkpoints taken during training (Pruthi et al., 2020).

we take the mean of the best 5 seeds for each privacy budget.

The measures of multilingual compression (sentence retrieval precision, CKA, IsoScore, RSA) are computed using distinct evaluation corpora comprising parallel sentences for all language pairs in the respective training set language sample. For models trained on XNLI, we use 3000 sentence pairs per language pair from the TED 2020 corpus (Reimers and Gurevych, 2020) and 3500 pairs from the WikiMatrix dataset (Schwenk et al., 2021). For models trained for POS, we use 3500 pairs from TED 2020, 3500 pairs from WikiMatrix, and 900 pairs from Tatoeba,[23,24,25] numbers chosen based on availability and memory usage.

Following Dufter and Schütze (2020), we evaluate the models at layers 0 and 8, which complement each other well with regard to the properties they capture, e.g., multilinguality and task-specificity (Choenni and Shutova, 2020; de Vries et al., 2020; Muller et al., 2021). We compute the sentence retrieval precision between language pairs and take the mean.[26] The IsoScore is computed for the contextualized representations of all examples in the respective corpus at once. In contrast, CKA and RSA scores are also computed per language pair, and then averaged across those.[27] For RSA, we use $D = 1 -$ Spearman's ρ and $S =$ Spearman's ρ as the dissimilarity and similarity metrics, respectively.[28] Finally, we average results for all four metrics across TED 2020, WikiMatrix, and Tatoeba, the two layers, and the 5 best seeds for each privacy budget. For comparison, we also compute all metrics for the original pretrained and a randomly initialized XLM-R model.

5.4 Results

Privacy, Compression, Performance We now empirically investigate the relationship between differential privacy, multilingual compression, and cross-lingual transfer performance. We present aggregated results in Figure 5.1 and non-aggregated results in Appendix 5.8.7. We observe that the zero-shot accuracy

[23] https://tatoeba.org

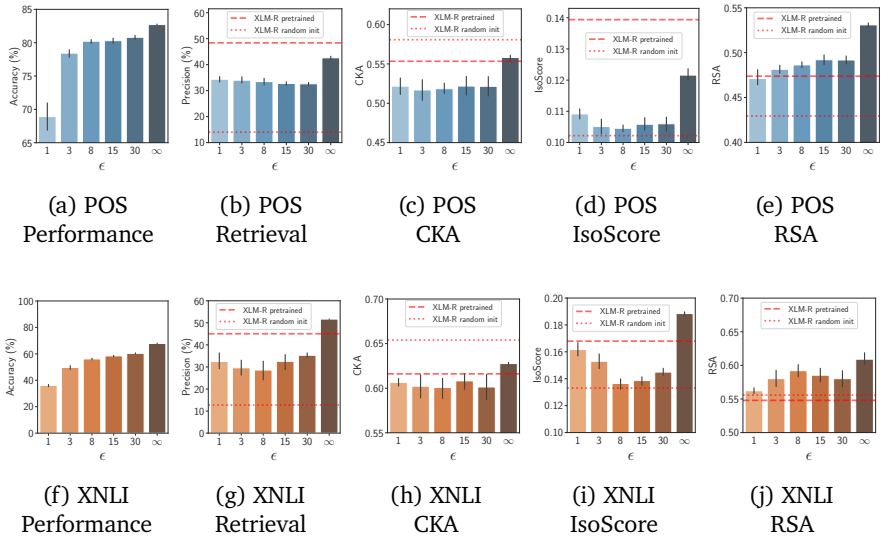
[24] We extract sentence pairs from Tatoeba using the tatoebatools library LBeaudoux/tatoebatools.

[25] We exclude TH from the WikiMatrix and Tatoeba evaluation sets for POS as there are insufficiently many sentence pairs available between TH and the remaining languages.

[26] Sentence retrieval is bidirectional (see Equation 5.3). Given $|L|$ languages, we therefore average over the full $\mathbb{R}^{|L|\times|L|}$ language pair matrix, only excluding the main diagonal.

[27] CKA and RSA are symmetrical. Given $|L|$ languages, we thus only use the upper triangle of the $\mathbb{R}^{|L|\times|L|}$ language pair matrix, still excluding the main diagonal.

[28] This is consistent with the results of Zhelezniak et al. (2019) and Lepori and McCoy (2020) showing that Spearman's ρ is more suitable for RSA with embeddings than conventional similarity metrics such as cosine similarity.



(a) POS Performance (b) POS Retrieval (c) POS CKA (d) POS IsoScore (e) POS RSA

(f) XNLI Performance (g) XNLI Retrieval (h) XNLI CKA (i) XNLI IsoScore (j) XNLI RSA

Figure 5.1: Task performance, sentence retrieval, CKA, IsoScore, and RSA results when fine-tuning with different privacy guarantees (∞=non-private). We add the original pretrained XLM-R and XLM-R with randomly initialized weights for comparison. The results show how non-private fine-tuning balances multilingual compression and task performance. Strongly private fine-tuning ($\varepsilon = 1$) is compatible with high compression (retrieval, CKA, IsoScore), but not with task performance. For medium levels of privacy (e.g., $\varepsilon = 8$), we see the result of balancing privacy and task performance at the expense of multilingual compression.

decreases as we finetune with stronger privacy guarantees (Figure 5.1a and 5.1f), which is expected due to the *privacy–utility tradeoff* (Geng et al., 2020). In particular, the relatively small sizes of our training datasets make private LM fine-tuning more challenging (Kerrigan et al., 2020; Habernal, 2021; Senge et al., 2022; Yu et al., 2022a) because, for a fixed number of update steps, the gradient noise added per update step grows as the size of the training dataset decreases (Tramèr and Boneh, 2021; McMahan et al., 2018). Note that although the private models tend to underperform the non-private models by a large margin on the validation set (>30% for XNLI, as shown in Appendix Table 5.6), the performance gap on the test set is noticeably smaller, showing that training with differential privacy, like other noise injection schemes (Bishop, 1995), is also a form of regularization.

Figure 5.1b and 5.1g display sentence retrieval precision when fine-tuning with different privacy budgets. The highest compression is achieved by the non-private models. The second-highest compression is achieved for $\varepsilon = 1$, our most private models. Both suggest non-linear privacy–compression interactions, with POS showing lowest compression for $\varepsilon = 30$ (or higher) and XNLI showing lowest compression for $\varepsilon = 8$. The results are very similar for IsoScore (Figure 5.1d, 5.1i) and also similar, albeit less pronounced for CKA (Figure 5.1c, 5.1h).[29] RSA, in contrast, exhibits very low scores for highly private models; see Appendix 5.8.5.

These results show that we can achieve *strong compression and strong performance at the cost of privacy* ($\varepsilon = \infty$), *strong compression and strong privacy at the cost of performance* ($\varepsilon = 1$), or *trade-off performance and privacy at the cost of compression* (e.g., $\varepsilon = 8$). It may seem counter-intuitive that multilingual compression and cross-lingual transfer performance are not strictly correlated. However, in the fine-tuning setting, we can sacrifice task-specific knowledge in favor of multilingual compression, which leads to poor performance. Vice-versa, a model may exploit spurious correlations in the data to make correct predictions without actually relying on cross-lingual signal. An example for the former case is the pretrained (but not finetuned) XLM-R, which scores highly in multilingual compression (as displayed in Figure 5.1) but has poor cross-lingual transfer performance in the downstream tasks.

We also find that in some fine-tuning settings, e.g., $\varepsilon = \infty$, the multilingual compression surpasses that of the pretrained XLM-R. While Liu et al. (2021c) have previously shown that sentence retrieval performance typically drops (i.e., compression worsens) over the course of fine-tuning (which we confirm in Appendix Figure 5.5), this finding clearly shows that there are exceptions. Future work may investigate this further.

Lastly, retrieval and CKA scores are always highest between typologically similar languages and languages over-represented in pretraining (see Table 5.1 for a comparison across languages) *across all levels of privacy*, as shown by the non-aggregated results in the Appendix Figure 5.6–5.13. This finding thus extends conclusions from prior work (Pires et al., 2019; Wu and Dredze, 2019; K et al., 2020; Lauscher et al., 2020) to private models.

[29] The randomly initialized XLM-R model shows high CKA scores. This is explained by the high dimensionality ($d = 768$) of the contextualized representations, considering that CKA saturates with increasing network width (Kornblith et al., 2019), and the high centroid similarity of random activations.

5.5 More multilingual, less interpretable?

Metric To answer this question, we introduce InfU (**Inf**luence **U**niformity), a measure of uniformity based on TracInCP influence scores for each training example in the multiparallel dataset $D = \{\ldots, i_1, \ldots, i_{|L|}, \ldots\}$, with i_j and i_k translation equivalents. We compute InfU for $\mathcal{M}$ and the translation equivalents $i = \{i_1, \ldots, i_{|L|}\}$ as follows:

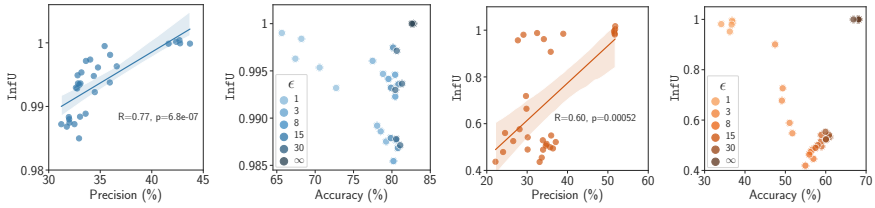
$$\text{InfU}(i) = \frac{1}{|L|} \sum_{k=1}^{|L|} \text{H}(\sigma(\text{TracInCP}(i_k, i))), \tag{5.6}$$

where H is the entropy with $\log_{|L|}$ and σ is a softmax used to obtain a probability distribution over influence scores. InfU is maximized (InfU = 1) for uniform influence scores, fulfilling $\text{TracInCP}(i_j, i_k) = \text{TracInCP}(i_q, i_r)$, $\forall j, k, q, r \in L$. This means a perfectly multilingual model that yields equivalent representations for translation equivalent examples obtains InfU = 1. In this scenario of maximum uniformity our model is also the least instance-interpretable because training data influence is minimally sparse, so we cannot easily identify influential examples for a prediction. We use InfU to study to what extent influence sparsity aligns with the metrics privacy and cross-lingual performance.

Setup We use 1000 training examples and compute TracInCP scores from the last 3 model checkpoints, taken every 100 steps, with their corresponding learning rates.[30]

Results and Analysis We plot the mean InfU against the mean sentence retrieval precision for our finetuned models and compute Pearson's R in Figure 5.2a and 5.2c. For both tasks, there is a significant ($p < 0.05$) strong positive correlation between the InfU score and multilingual compression as determined through sentence retrieval. This supports the idea that *multilingual compression is at odds with training data influence*. See also how highly private and low-performing models score highly in InfU (Figure 5.2b, 5.2d); and non-private and high-performing models do the same. For medium levels of privacy we, however, see a trade-off characterized by lower InfU, i.e., better instance-interpretability, and medium performance. Strong *privacy* guarantees, sparse training data influence estimates, and performance are incompatible, because the high-performing models

[30] Since the learning rate changes every training step, we use the learning rate from the end of each checkpointing interval.



(a) POS InfU – Retrieval (b) POS InfU – Perf. (c) XNLI InfU – Retrieval (d) XNLI InfU – Perf.

Figure 5.2: Linear fit and Pearson correlation between the influence uniformity InfU and sentence retrieval precision (5.2a, 5.2c) and InfU versus downstream performance for different levels of privacy (5.2b, 5.2d). We see significant positive correlations between retrieval precision and InfU, suggesting a negative correlation between multilingual compression and training data influence sparsity. For task performance, we see the trade-off between training data influence sparsity (InfU) and privacy, which aligns with our theoretical expectations (§ 5.2).

are strictly low in privacy and training data influence sparsity, and the models high in privacy are strictly low in performance and training data influence sparsity.

5.6 Related Work

While privacy, fairness, and interpretability *individually* have enjoyed ample attention from the research community in recent years (Liu et al., 2021a; Mehrabi et al., 2021; Søgaard, 2021), the interactions between these objectives have not been explored much (Ruder et al., 2022). Some prior work has focused on the interactions between group fairness and differential privacy, suggesting that the two objectives are at odds, although this relationship also depends on the selected notion of fairness (Bagdasaryan et al., 2019; Cummings et al., 2019; Chang and Shokri, 2021; Hansen et al., 2024). Somewhat in contrast to this work, we show that linguistic fairness (group fairness over linguistic communities) and differential privacy may align for multilingual language models. Furthermore, Naidu et al. (2021) and Shokri et al. (2021) have studied the interaction between privacy and feature attribution methods for model explainability. While the former show that privacy and feature attribution methods can align, the latter find that model explanations are at risk of membership inference attacks. Closest to our work is contemporaneous work by Strobel and Shokri (2022) who discuss the interactions of data privacy with fairness, explainability, and robustness. Our

work differs from theirs in that we are particularly concerned with multilingual language models and we consider instance-based interpretability methods while they consider feature attribution methods. Strobel and Shokri (2022) also call for more research at the intersection of different objectives rather than working on one at a time.

5.7 Conclusion

We presented a preliminary investigation of how multilingual compression, differential privacy, training data influence, and linguistic fairness interact in multilingual models. We found that privacy and influence are incompatible, while privacy and linguistic fairness, often said to be at odds, are theoretically compatible through multilingual compression. We also explored these interactions empirically. Our results support the idea that high multilingual compression can be achieved either while optimizing for performance or while optimizing for privacy, but that by trading off privacy and performance, we compromise compression. Finding practical trade-offs between *all* these dimensions remains an open challenge. Finally, we introduced a new diagnostic metric, influence uniformity, which we used to validate that privacy and training data influence sparsity are incompatible, and that the interactions between privacy, training data influence sparsity, and multilingual compression are, therefore, also non-linear.

Ethical Aspects and Broader Impact

It is crucial that NLP goes beyond performance and studies the interaction of objectives such as privacy, interpretability, and fairness, also in multilingual NLP (Ruder et al., 2022). Our work aims to provide a starting point for further research in this area. Our empirical investigation, including the models we train, fully relies on publicly available models and data. Moreover, we do not create any new datasets. Therefore, we foresee no misuse of the results of our work.

Acknowledgements

We thank the anonymous reviewers and members of the CoAStaL group for their helpful feedback and suggestions. Phillip Rust is funded by the Novo Nordisk Foundation (grant NNF 20SA0066568).

Language	ISO	Family	Script	Tokens (M)	Size (GiB)
Arabic	AR	Afro-Asiatic	Arabic	2 869	28.0
Bulgarian	BG	Indo-European	Cyrillic	5 487	57.5
Chinese	ZH	Sino-Tibetan	Chinese	435	63.5
French	FR	Indo-European	Latin	9 780	56.8
German	DE	Indo-European	Latin	10 297	66.6
Greek	EL	Indo-European	Greek	4 285	46.9
Hindi	HI	Indo-European	Devanagari*	1 803	20.7
Indonesian	ID	Austronesian	Latin	22 704	148.3
Italian	IT	Indo-European	Latin	4 983	30.2
Japanese	JA	Japonic	Japanese	530	69.3
Kiswahili	SW	Niger-Congo	Latin	275	1.6
Korean	KO	Koreanic	Korean	5 644	54.2
Portuguese	PT	Indo-European	Latin	8 405	49.1
Russian	RU	Indo-European	Cyrillic	23 408	278.0
Thai	TH	Kra-Dai	Thai	1 834	71.7
Turkish	TR	Turkic	Latin	2 736	20.9
Urdu	UR	Indo-European	Arabic*	815	6.2
Vietnamese	VI	Austro-Asiatic	Latin	24 757	137.3

Table 5.1: Overview of languages used in our experiments. Tokens (in millions) and size (in Gibibytes) refer to the respective monolingual corpora in XLM-R's pretraining corpus. Numbers taken from Conneau et al. (2020a). *: includes romanized variants also used in pretraining.

5.8 Appendix

5.8.1 Reproducibility

We make our code available at `xplip/multilingual-lm-objectives`.

Implementation Our implementation is written in PyTorch v1.10.0 (Paszke et al., 2019) for Python 3.9.5 and builds on code from the following repositories:

- `huggingface/transformers` v4.9.2 (Wolf et al., 2020) for model training and evaluation
- `lxuechen/private-transformers` v0.1.0 (Li et al., 2022b) for DP-training
- `pdufter/minimult` (Dufter and Schütze, 2020) for computing sentence retrieval precision
- `jayroxis/CKA-similarity` for computing CKA scores

Dataset	Download Link & Reference
UD v2.8 (POS)	repository/xmlui/handle/11234/1-3683 Nivre et al. (2020); Zeman et al. (2021)
XNLI	datasets/xnli Conneau et al. (2018); Lhoest et al. (2021)
TED 2020	UKPLab/sentence-transformers/blob/master/docs/datasets/TED2020.md Reimers and Gurevych (2020)
WikiMatrix	facebookresearch/LASER/tree/main/tasks/WikiMatrix Schwenk et al. (2021)
Tatoeba	LBeaudoux/tatoebatools

Table 5.2: Links and references to the datasets we used in our experiments. License information is also available via these links. We ensure that we comply with respective license conditions and only use the data within their intended use policy where applicable.

- mlepori1/Picking_BERTs_Brain (Lepori and McCoy, 2020) for computing RSA scores
- bcbi-edu/p_eickhoff_isoscore (Rudman et al., 2022) for computing IsoScores
- FengNiMa/VAE-TracIn-pytorch (Kong and Chaudhuri, 2021) for computing TracInCP scores.

Models We primarily use the pretrained XLM-RoBERTa (XLM-R; Conneau et al., 2020a) base model and tokenizer from xlm-roberta-base. XLM-R (base) is a 12-layer encoder-only transformer with a vocabulary size of 250k and ~277M total parameters pretrained via masked language modeling on the 100-language CC-100 dataset.

In Appendix 5.8.6, we further conduct experiments with multilingual BERT (mBERT; Devlin et al., 2019), using the base model and tokenizer from bert-base-multilingual-cased. mBERT is a 12-layer encoder-only transformer with a vocabulary size of 120k and ~177M total parameters pretrained via masked language modeling on Wikipedia data in 104 languages.

Data We provide download links and references for the various datasets we used in Table 5.2.

Language	Treebank	# Sentences
AR	Arabic-PADT	680
DE	German-GSD	977
ES	Spanish-GSD	426
HI	Hindi-HDTB	1 684
ID	Indonesian-GSD	557
KO	Korean-Kaist	2 287
RU	Russian-SynTagRus	6 491

Table 5.3: Overview of the UD v2.8 (Nivre et al., 2020; Zeman et al., 2021) treebanks (test splits only) that we use as test sets in our POS tagging experiments (§ 5.3,5.4) including their respective sizes (number of sentences).

Hardware We train on single Nvidia Titan RTX, A100 (both with CUDA version 11.0), and RTX 3090 (with CUDA version 11.5) GPUs. All machines have at least 64GB of RAM, which is required to compute the IsoScore for our larger evaluation sets (e.g., TED 2020 for POS).

Runtime Fine-tuning with evaluation during training on the Titan RTX, which is the slowest of the GPUs used, takes 2–3 hours for POS and 5–6 hours for XNLI. Computing TracInCP influence scores for one finetuned model takes about 30–45 minutes.

Carbon Footprint Our fine-tuning runs accumulated ~36 compute days on the hardware mentioned above (most experiments were conducted on the less powerful Titan RTX GPUs) according to Weights & Biases[31], where we logged our experiments. Although we do not have precise numbers, a highly conservative estimate of the total compute spent including prototyping, hyper-parameter search, and all our evaluations is ~75 compute days.

5.8.2 (ε, δ)-Differential Privacy

In § 5.2, we provide the definition of ε-differential privacy (DP), also called pure DP, as the basis for our theoretical exploration. In our experiments, we rely on (ε, δ)-DP (Dwork and Roth, 2014), also called approximate-DP, which is typically used in practice and relaxes the privacy guarantees by a (small) δ as follows:

[31] https://wandb.ai/

A randomized algorithm $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{Y}$ is *(ε, δ)-differentially private* (Dwork, 2006) iff for all adjacent datasets $D, D' \in \mathcal{D}$ and all $Y \subset \mathcal{Y}$,

$$\mathbb{P}(\mathcal{M}(D) \in Y) \leq \exp(\varepsilon) \cdot \mathbb{P}(\mathcal{M}(D') \in Y) + \delta.$$

5.8.3 Best Fine-Tuning Settings

As mentioned in § 5.3, we pre-selected a set of suitable learning rates (LRs) for each task and ran 3 random initializations each. Based on the validation performance, we then selected the following 5 best settings for each privacy budget and task:

ε	POS LR (# Seeds)	XNLI LR (# Seeds)
1	5e−4 (2); 7e−4 (3)	3e−4 (1); 4e−4 (2); 5e−4 (2)
3	5e−4 (2); 7e−4 (3)	3e−4 (1); 4e−4 (2); 5e−4 (2)
8	5e−4 (3); 7e−4 (2)	4e−4 (2); 5e−4 (3)
15	3e−4 (1); 5e−4 (2); 7e−4 (2)	3e−4 (1); 4e−4 (2); 5e−4 (2)
30	3e−4 (1); 5e−4 (2); 7e−4 (2)	3e−4 (1); 4e−4 (2); 5e−4 (2)
∞	5e−5 (2); 7e−5 (2); 1e−4 (1)	9e−5 (2); 1e−4 (3)

Table 5.4: Best 5 settings for each task and privacy budget. Includes LR and the corresponding number of random initializations (# seeds).

5.8.4 IsoScore Algorithm

Algorithm 2 describes the IsoScore algorithm (Rudman et al., 2022).

Algorithm 2 IsoScore (Rudman et al., 2022)

begin Let $X \subset \mathbb{R}^n$ be a finite collection of points.
  Let X^{PCA} denote the points in X transformed by the first n principal components.
  Define $\Sigma_D \in \mathbb{R}^n$ as the diagonal of the covariance matrix of X^{PCA}.
  Normalize diagonal to $\hat{\Sigma}_D := \sqrt{n} \cdot \Sigma_D / \|\Sigma_D\|$, where $\|\cdot\|$ is the standard Euclidean norm.
  The isotropy defect is $\delta(X) := \|\hat{\Sigma}_D - \mathbf{1}\| / \sqrt{2(n - \sqrt{n})}$, where $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^n$
  X uniformly occupies $\phi(X) := (n - \delta(X)^2(n - \sqrt{n}))^2 / n^2$ percent of ambient dimensions.
  Transform $\phi(X)$ so it can take values in $[0, 1]$, via $\iota(X) := (n \cdot \phi(X) - 1)/(n - 1)$.
  return: $\iota(X)$
end

5.8.5 Further Analysis of RSA Results

As we see in § 5.4, RSA aligns with sentence retrieval precision, CKA, and IsoScore in producing higher scores for non-private models. However, there is a mismatch between RSA and the other metrics in highly private regimes, where our most private models ($\varepsilon = 1$) do not exhibit high RSA scores. Instead, the aggregated RSA scores peak at medium levels of privacy ($\varepsilon \in \{8, 15\}$) and for the non-private ($\varepsilon = \infty$) models. Unlike for the other metrics, there is also no clear trend among our two tasks in terms of whether the pretrained or a randomly initialized XLM-R model scores higher in RSA.

A closer look at the non-aggregated results (Appendix Figure 5.10, 5.11, and 5.14) shows how the similarity patterns obtained from RSA are often unexpected. For instance, the similarities between the typologically distant languages FR and ZH are consistently high for the TED 2020 corpus whereas scores for typologically closer languages are lower (Figure 5.10). Based on prior work by, for example, Pires et al. (2019), Wu and Dredze (2019), and Lauscher et al. (2020), we would expect the model to first compress similar languages before achieving compression for distant ones. Sometimes, we also observe extreme jumps in similarity between layers 0 and 8, for instance, between IT and TR in the Tatoeba corpus (Figure 5.11). We do not find these jumps in CKA and sentence retrieval.

One reason why RSA scores may be more sensitive to stricter privacy guarantees (e.g., $\varepsilon = 1$) is that the correlation between sentence vector distances is very sensitive to outliers. Differential privacy reduces the number of such outliers, effectively regularizing the correlation coefficients.

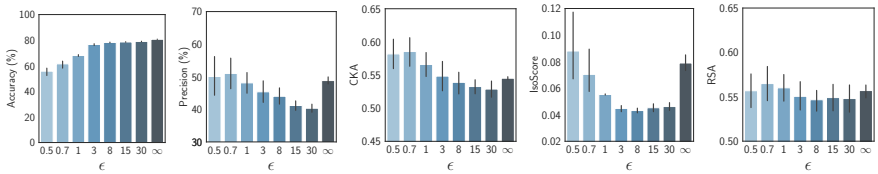
5.8.6 Multilingual BERT Results

In Figure 5.3 and 5.4, we present results from re-running the experiments from § 5.4 and § 5.5 with mBERT. We make two changes to the experimental setup outlined above: We use representations extracted at layer 8, which showed to be more meaningful than layer 0 in the XLM-R experiments, to compute the multilinguality metrics. We also include two additional privacy settings, $\varepsilon = 0.5$ and $\varepsilon = 0.7$, as we found mBERT to be easier to finetune with strong privacy guarantees than XLM-R.

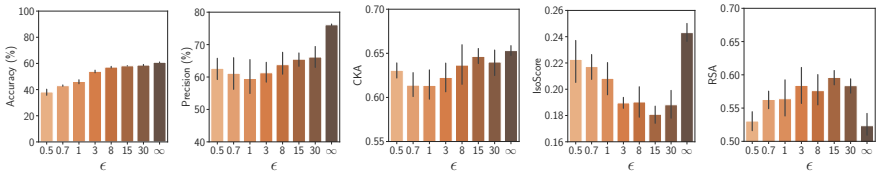
We see the same trends as for XLM-R: performance strictly increases with decreasing privacy while the multilinguality metrics tend to follow a U-shape,[32] i.e., they are high for strong privacy settings (small ε) and low privacy settings

[32]We again refer to Appendix 5.8.5 for a discussion of the RSA results.

(large ϵ) and decrease towards medium privacy. Likewise, we find a positive correlation between InfU and cross-lingual sentence retrieval precision. The correlation is strong for part-of-speech tagging (POS) but it is mild for XNLI. We believe this may be due to mBERT being less sensitive to the privacy parameter (Figure 5.3g is not symmetrical; considering even stronger privacy settings would likely even out the U-shape). Overall, these results further support our finding that there is a negative correlation between multilingual compression and training data influence sparsity.

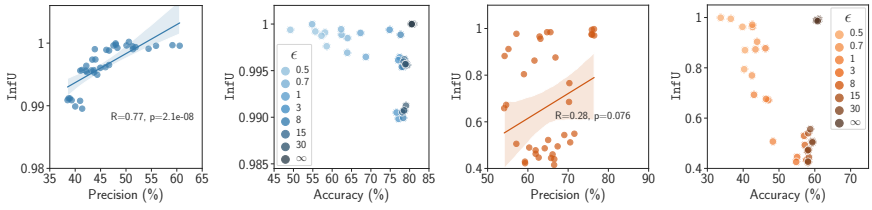


(a) POS Performance (b) POS Retrieval (c) POS CKA (d) POS IsoScore (e) POS RSA



(f) XNLI Performance (g) XNLI Retrieval (h) XNLI CKA (i) XNLI IsoScore (j) XNLI RSA

Figure 5.3: Aggregated **mBERT** results, analogous to Figure 5.1.



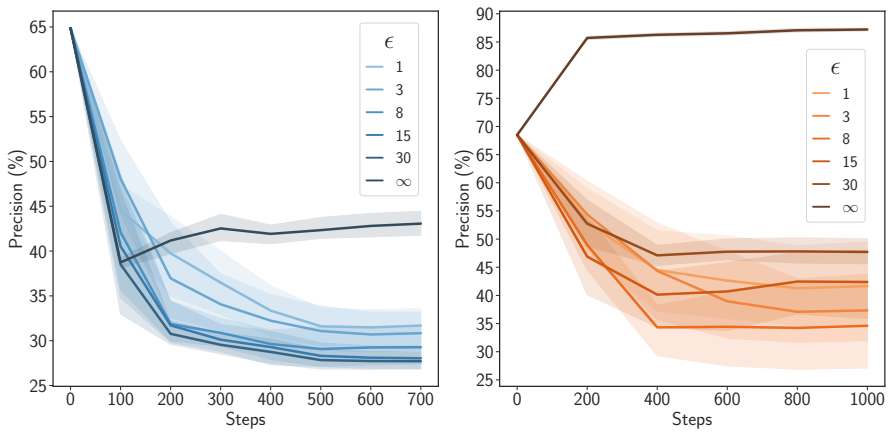
(a) POS InfU – Retrieval (b) POS InfU – Perf. (c) XNLI InfU – Retrieval (d) XNLI InfU – Perf.

Figure 5.4: Aggregated **mBERT** results, analogous to Figure 5.2.

5.8.7 Detailed Results for Experiments in § 5.4

Figure 5.5 shows the development of the mean sentence retrieval precision at layer 8 for POS and XNLI over the course of fine-tuning with different privacy budgets.
We further present non-aggregated results for

- POS performance in Table 5.5
- XNLI performance in Table 5.6
- Sentence retrieval for POS in Figure 5.6 and 5.7
- Sentence retrieval for XNLI in Figure 5.12
- CKA for POS in Figure 5.8 and 5.9
- CKA for XNLI in Figure 5.13
- IsoScore for POS in Table 5.7
- IsoScore for XNLI in Table 5.8
- RSA for POS in Figure 5.10 and 5.11
- RSA for XNLI in Figure 5.14.



(a) POS (b) XNLI

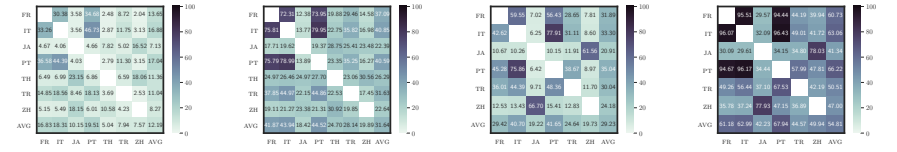
Figure 5.5: Mean sentence retrieval precision for our TED 2020 splits (different languages/data for POS and XNLI) at layer 8 over the course of fine-tuning with different privacy budgets (ε). $\varepsilon = \infty$ denotes non-private models. Error bands show variation around the mean over 5 random seeds. At Steps = 0, all models are equivalent to the pretrained XLM-R Base. We see that the non-private models can retain (and for XNLI even improve) their multilingual compression much better than the private models and have less variation.

ε	AR	DE	ES	HI	ID	KO	RU	AVG
1	68.3 / 64.6	75.5 / 75.1	79.8 / 79.0	65.0 / 63.3	73.8 / 71.9	66.1 / 54.2	74.8 / 74.0	71.9 / 68.9
3	79.1 / 76.6	86.6 / 86.8	90.3 / 89.3	74.4 / 70.9	82.6 / 79.4	71.1 / 59.4	86.1 / 86.3	81.4 / 78.4
8	81.0 / 77.6	88.4 / 88.3	91.6 / 90.2	78.2 / 75.6	84.2 / 81.2	70.8 / 60.9	87.1 / 87.4	83.0 / 80.2
15	81.3 / 78.4	88.8 / 89.0	92.4 / 90.9	77.0 / 73.2	83.9 / 80.7	71.9 / 61.8	87.7 / 87.8	83.3 / 80.3
30	81.8 / 78.7	89.4 / 89.6	92.9 / 91.5	77.6 / 74.0	84.3 / 81.1	72.3 / 62.2	88.2 / 88.4	83.8 / 80.8
∞	**83.8 / 79.7**	**91.5 / 91.2**	**95.0 / 93.2**	**82.8 / 80.2**	**86.2 / 81.3**	**74.2 / 62.9**	**89.9 / 90.2**	**86.2 / 82.7**

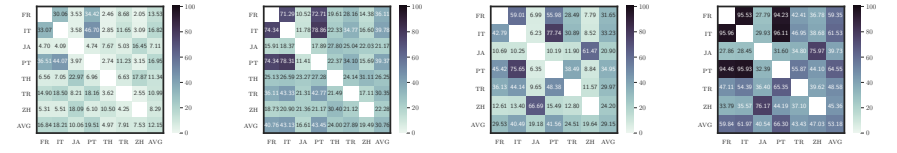
Table 5.5: **POS** Performance (validation / test accuracy) when fine-tuning XLM-R Base with different privacy budgets (ε). We show results averaged over 5 random seeds each. $\varepsilon = \infty$ denotes non-private models. AVG is the average over the 7 languages. See § 5.3 for our experimental setup. We see that performance increases with decreased privacy across all languages.

ε	AR	DE	EL	RU	SW	TH	UR	AVG
1	37.3 / 37.4	36.8 / 37.0	36.6 / 36.5	36.3 / 36.2	34.3 / 34.5	35.6 / 35.7	35.6 / 35.6	36.1 / 36.1
3	49.6 / 50.3	49.3 / 51.0	50.8 / 51.5	49.7 / 50.2	45.9 / 47.2	48.8 / 49.5	47.6 / 48.2	48.8 / 49.7
8	55.9 / 56.4	56.8 / 58.5	58.2 / 58.1	56.3 / 57.1	52.0 / 53.2	55.6 / 55.7	53.3 / 53.7	55.5 / 56.1
15	59.1 / 58.3	60.4 / 60.8	61.5 / 60.9	59.7 / 59.5	54.4 / 54.8	58.9 / 58.2	56.4 / 56.1	58.6 / 58.4
30	61.6 / 60.8	63.6 / 63.1	64.8 / 62.0	62.0 / 61.1	56.5 / 57.3	61.2 / 60.2	58.6 / 57.8	61.2 / 60.3
∞	**90.9 / 67.8**	**96.2 / 70.5**	**95.5 / 70.1**	**93.4 / 69.7**	**79.0 / 62.5**	**91.6 / 68.5**	**86.8 / 65.4**	**90.5 / 67.8**

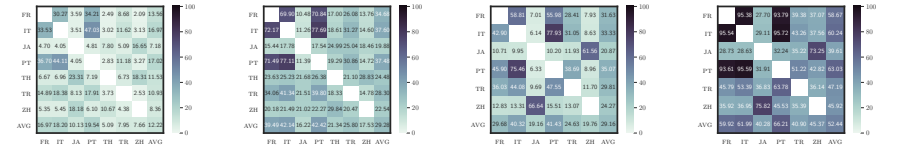
Table 5.6: **XNLI** Performance (validation / test accuracy) when fine-tuning XLM-R Base with different privacy budgets (ε). We show results averaged over 5 random seeds each. $\varepsilon = \infty$ denotes non-private models. AVG is the average over the 7 languages. See § 5.3 for our experimental setup. We see that performance increases with decreased privacy across all languages. Here, we also particularly observe that the gap between validation and test performance is substantially lower for private models, which shows the strong regularization effect of training with differential privacy.



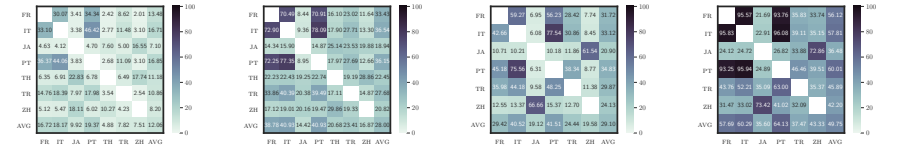
(a) TED, $\varepsilon = 1$, $l = 0$ (b) TED, $\varepsilon = 1$, $l = 8$ (c) WM, $\varepsilon = 1$, $l = 0$ (d) WM, $\varepsilon = 1$, $l = 8$



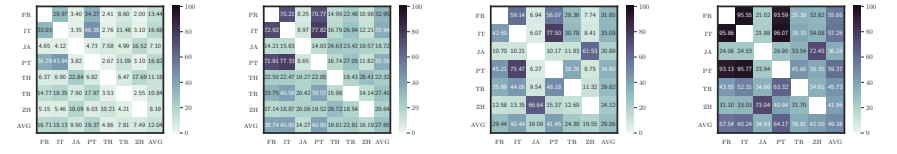
(e) TED, $\varepsilon = 3$, $l = 0$ (f) TED, $\varepsilon = 3$, $l = 8$ (g) WM, $\varepsilon = 3$, $l = 0$ (h) WM, $\varepsilon = 3$, $l = 8$



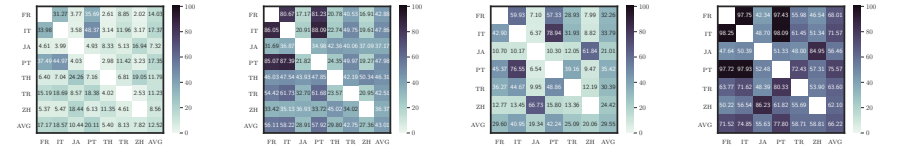
(i) TED, $\varepsilon = 8$, $l = 0$ (j) TED, $\varepsilon = 8$, $l = 8$ (k) WM, $\varepsilon = 8$, $l = 0$ (l) WM, $\varepsilon = 8$, $l = 8$



(m) TED, $\varepsilon = 15$, $l = 0$ (n) TED, $\varepsilon = 15$, $l = 8$ (o) WM, $\varepsilon = 15$, $l = 0$ (p) WM, $\varepsilon = 15$, $l = 8$

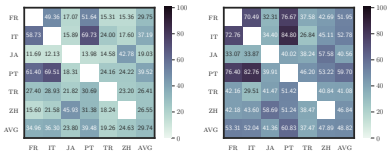


(q) TED, $\varepsilon = 30$, $l = 0$ (r) TED, $\varepsilon = 30$, $l = 8$ (s) WM, $\varepsilon = 30$, $l = 0$ (t) WM, $\varepsilon = 30$, $l = 8$

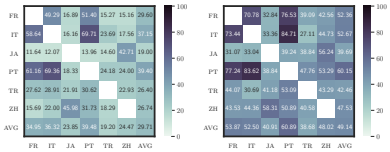


(u) TED, $\varepsilon = \infty$, $l = 0$ (v) TED, $\varepsilon = \infty$, $l = 8$ (w) WM, $\varepsilon = \infty$, $l = 0$ (x) WM, $\varepsilon = \infty$, $l = 8$

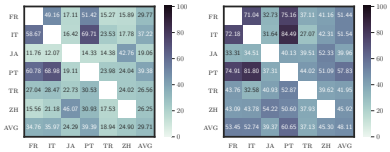
Figure 5.6: **POS** Sentence retrieval results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.



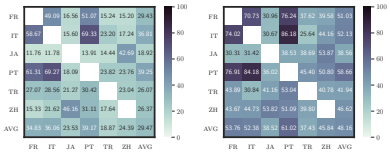
(a) $\varepsilon = 1$, $l = 0$ (b) $\varepsilon = 1$, $l = 8$



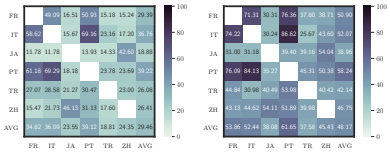
(c) $\varepsilon = 3$, $l = 0$ (d) $\varepsilon = 3$, $l = 8$



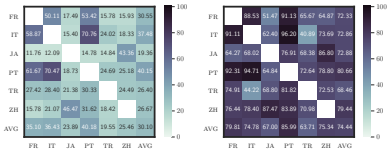
(e) $\varepsilon = 8$, $l = 0$ (f) $\varepsilon = 8$, $l = 8$



(g) $\varepsilon = 15$, $l = 0$ (h) $\varepsilon = 15$, $l = 8$

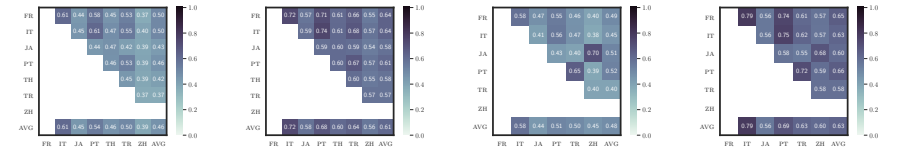


(i) $\varepsilon = 30$, $l = 0$ (j) $\varepsilon = 30$, $l = 8$

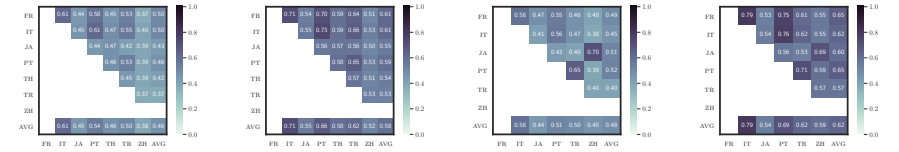


(k) $\varepsilon = \infty$, $l = 0$ (l) $\varepsilon = \infty$, $l = 8$

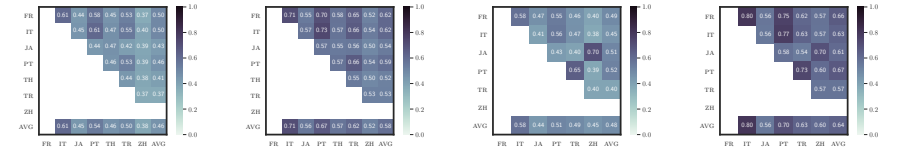
Figure 5.7: **POS** sentence retrieval results for the Tatoeba dataset and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.



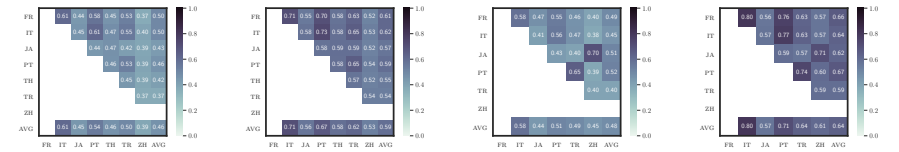
(a) TED, $\varepsilon = 1$, $l = 0$ (b) TED, $\varepsilon = 1$, $l = 8$ (c) WM, $\varepsilon = 1$, $l = 0$ (d) WM, $\varepsilon = 1$, $l = 8$



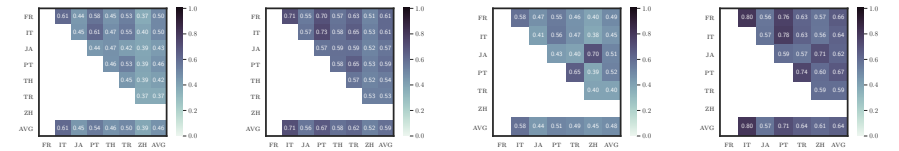
(e) TED, $\varepsilon = 3$, $l = 0$ (f) TED, $\varepsilon = 3$, $l = 8$ (g) WM, $\varepsilon = 3$, $l = 0$ (h) WM, $\varepsilon = 3$, $l = 8$



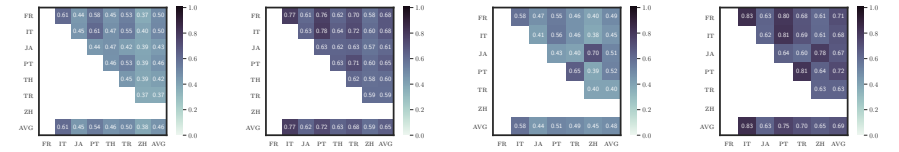
(i) TED, $\varepsilon = 8$, $l = 0$ (j) TED, $\varepsilon = 8$, $l = 8$ (k) WM, $\varepsilon = 8$, $l = 0$ (l) WM, $\varepsilon = 8$, $l = 8$



(m) TED, $\varepsilon = 15$, $l = 0$ (n) TED, $\varepsilon = 15$, $l = 8$ (o) WM, $\varepsilon = 15$, $l = 0$ (p) WM, $\varepsilon = 15$, $l = 8$

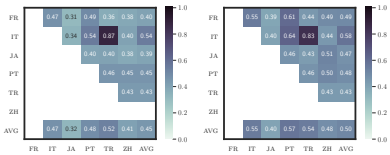


(q) TED, $\varepsilon = 30$, $l = 0$ (r) TED, $\varepsilon = 30$, $l = 8$ (s) WM, $\varepsilon = 30$, $l = 0$ (t) WM, $\varepsilon = 30$, $l = 8$



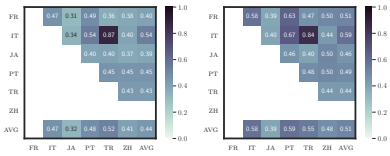
(u) TED, $\varepsilon = \infty$, $l = 0$ (v) TED, $\varepsilon = \infty$, $l = 8$ (w) WM, $\varepsilon = \infty$, $l = 0$ (x) WM, $\varepsilon = \infty$, $l = 8$

Figure 5.8: **POS** CKA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.

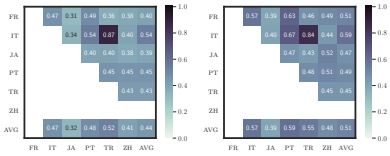


(a) $\varepsilon = 1, l = 0$ (b) $\varepsilon = 1, l = 8$

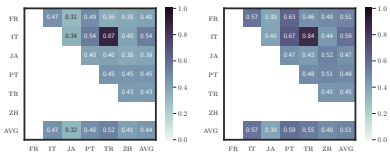
(c) $\varepsilon = 3, l = 0$ (d) $\varepsilon = 3, l = 8$



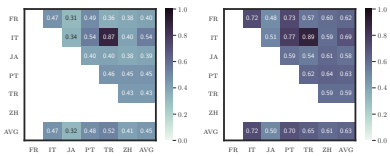
(e) $\varepsilon = 8, l = 0$ (f) $\varepsilon = 8, l = 8$



(g) $\varepsilon = 15, l = 0$ (h) $\varepsilon = 15, l = 8$

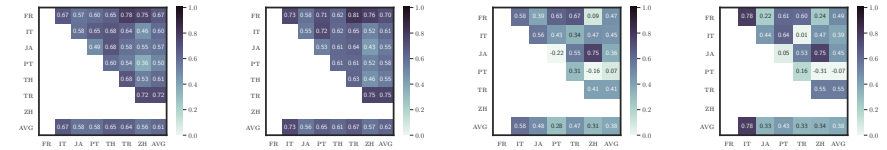


(i) $\varepsilon = 30, l = 0$ (j) $\varepsilon = 30, l = 8$

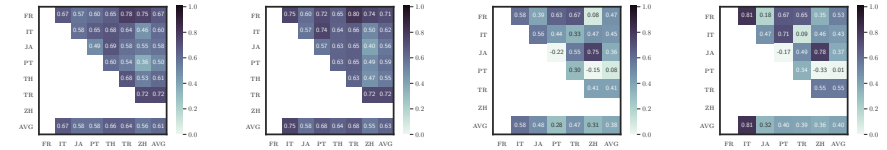


(k) $\varepsilon = \infty, l = 0$ (l) $\varepsilon = \infty, l = 8$

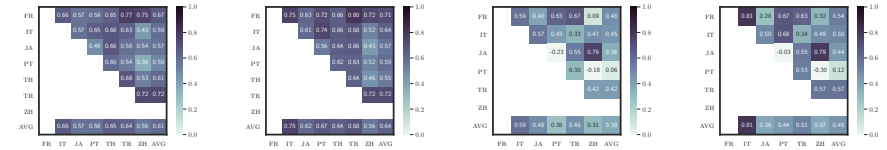
Figure 5.9: **POS** CKA results for the Tatoeba dataset and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.



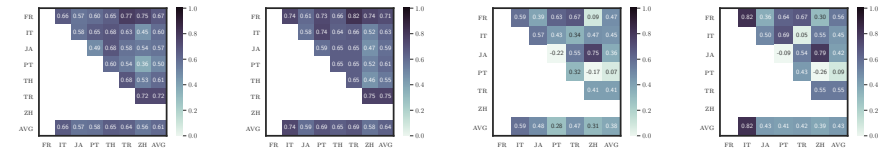
(a) TED, $\varepsilon = 1$, $l = 0$ (b) TED, $\varepsilon = 1$, $l = 8$ (c) WM, $\varepsilon = 1$, $l = 0$ (d) WM, $\varepsilon = 1$, $l = 8$



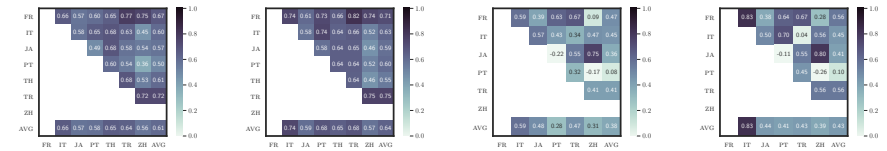
(e) TED, $\varepsilon = 3$, $l = 0$ (f) TED, $\varepsilon = 3$, $l = 8$ (g) WM, $\varepsilon = 3$, $l = 0$ (h) WM, $\varepsilon = 3$, $l = 8$



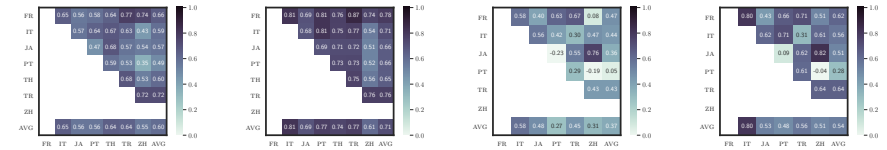
(i) TED, $\varepsilon = 8$, $l = 0$ (j) TED, $\varepsilon = 8$, $l = 8$ (k) WM, $\varepsilon = 8$, $l = 0$ (l) WM, $\varepsilon = 8$, $l = 8$



(m) TED, $\varepsilon = 15$, $l = 0$ (n) TED, $\varepsilon = 15$, $l = 8$ (o) WM, $\varepsilon = 15$, $l = 0$ (p) WM, $\varepsilon = 15$, $l = 8$

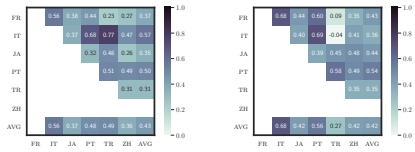


(q) TED, $\varepsilon = 30$, $l = 0$ (r) TED, $\varepsilon = 30$, $l = 8$ (s) WM, $\varepsilon = 30$, $l = 0$ (t) WM, $\varepsilon = 30$, $l = 8$



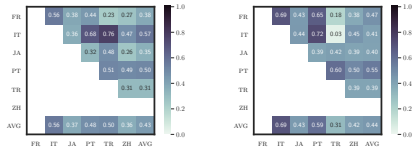
(u) TED, $\varepsilon = \infty$, $l = 0$ (v) TED, $\varepsilon = \infty$, $l = 8$ (w) WM, $\varepsilon = \infty$, $l = 0$ (x) WM, $\varepsilon = \infty$, $l = 8$

Figure 5.10: **POS** RSA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.



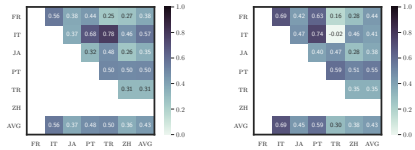
(a) $\varepsilon = 1, l = 0$

(b) $\varepsilon = 1, l = 8$



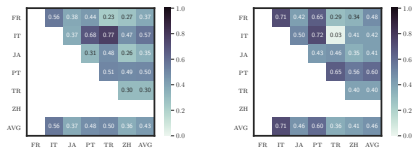
(c) $\varepsilon = 3, l = 0$

(d) $\varepsilon = 3, l = 8$



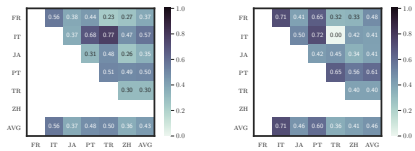
(e) $\varepsilon = 8, l = 0$

(f) $\varepsilon = 8, l = 8$



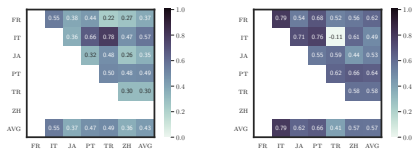
(g) $\varepsilon = 15, l = 0$

(h) $\varepsilon = 15, l = 8$



(i) $\varepsilon = 30, l = 0$

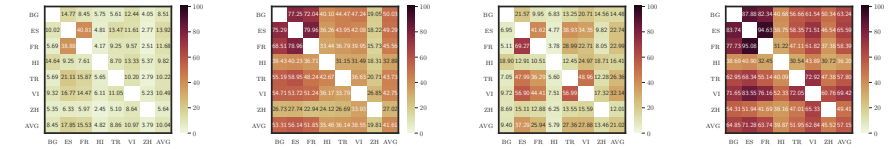
(j) $\varepsilon = 30, l = 8$



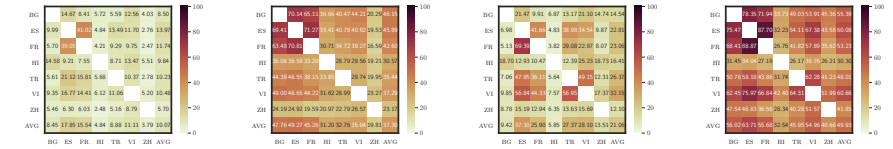
(k) $\varepsilon = \infty, l = 0$

(l) $\varepsilon = \infty, l = 8$

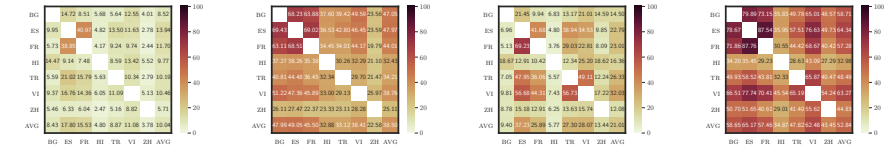
Figure 5.11: **POS** RSA results for the Tatoeba dataset and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0. Also note that, unlike in CKA (Fig. 5.9), the similarity between IT and TR is high at l=0 but low at l=8.



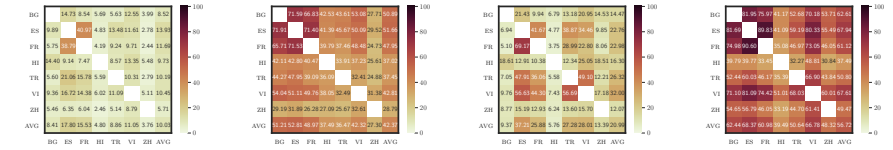
(a) TED, $\varepsilon = 1$, $l = 0$ (b) TED, $\varepsilon = 1$, $l = 8$ (c) WM, $\varepsilon = 1$, $l = 0$ (d) WM, $\varepsilon = 1$, $l = 8$



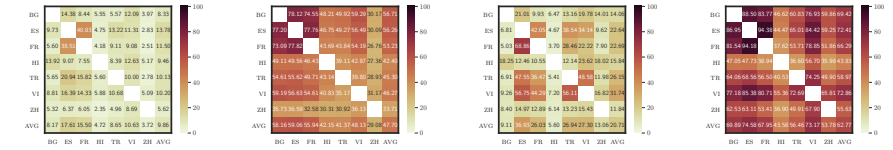
(e) TED, $\varepsilon = 3$, $l = 0$ (f) TED, $\varepsilon = 3$, $l = 8$ (g) WM, $\varepsilon = 3$, $l = 0$ (h) WM, $\varepsilon = 3$, $l = 8$



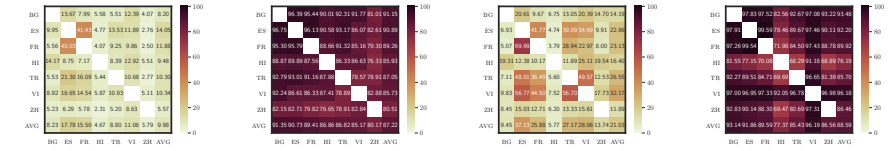
(i) TED, $\varepsilon = 8$, $l = 0$ (j) TED, $\varepsilon = 8$, $l = 8$ (k) WM, $\varepsilon = 8$, $l = 0$ (l) WM, $\varepsilon = 8$, $l = 8$



(m) TED, $\varepsilon = 15$, $l = 0$ (n) TED, $\varepsilon = 15$, $l = 8$ (o) WM, $\varepsilon = 15$, $l = 0$ (p) WM, $\varepsilon = 15$, $l = 8$

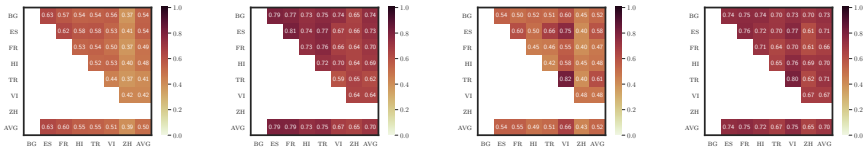


(q) TED, $\varepsilon = 30$, $l = 0$ (r) TED, $\varepsilon = 30$, $l = 8$ (s) WM, $\varepsilon = 30$, $l = 0$ (t) WM, $\varepsilon = 30$, $l = 8$

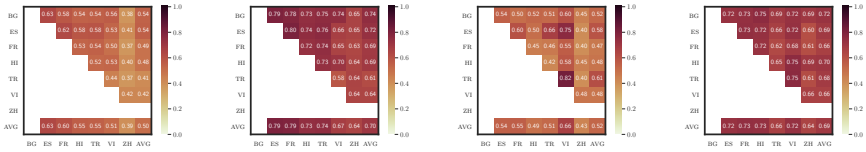


(u) TED, $\varepsilon = \infty$, $l = 0$ (v) TED, $\varepsilon = \infty$, $l = 8$ (w) WM, $\varepsilon = \infty$, $l = 0$ (x) WM, $\varepsilon = \infty$, $l = 8$

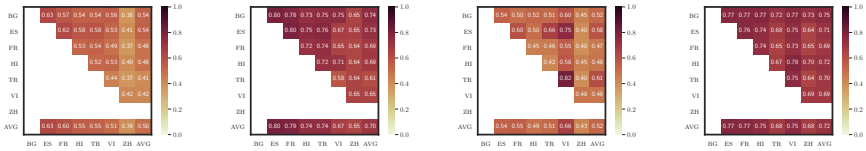
Figure 5.12: **XNLI** Sentence retrieval results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.



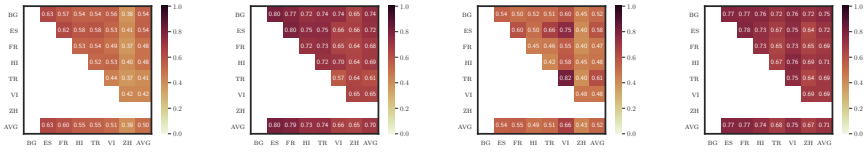
(a) TED, $\varepsilon = 1$, $l = 0$ (b) TED, $\varepsilon = 1$, $l = 8$ (c) WM, $\varepsilon = 1$, $l = 0$ (d) WM, $\varepsilon = 1$, $l = 8$



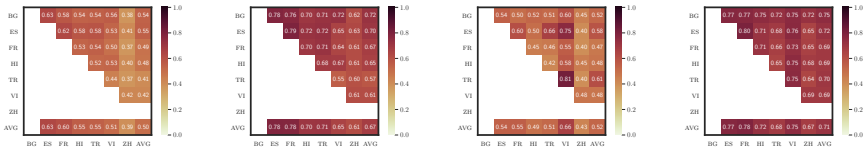
(e) TED, $\varepsilon = 3$, $l = 0$ (f) TED, $\varepsilon = 3$, $l = 8$ (g) WM, $\varepsilon = 3$, $l = 0$ (h) WM, $\varepsilon = 3$, $l = 8$



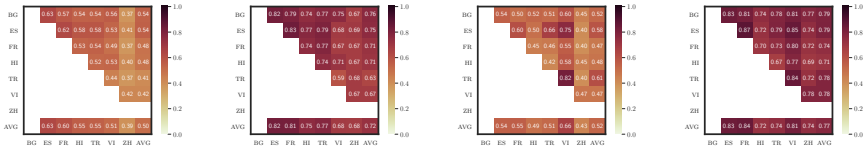
(i) TED, $\varepsilon = 8$, $l = 0$ (j) TED, $\varepsilon = 8$, $l = 8$ (k) WM, $\varepsilon = 8$, $l = 0$ (l) WM, $\varepsilon = 8$, $l = 8$



(m) TED, $\varepsilon = 15$, $l = 0$ (n) TED, $\varepsilon = 15$, $l = 8$ (o) WM, $\varepsilon = 15$, $l = 0$ (p) WM, $\varepsilon = 15$, $l = 8$

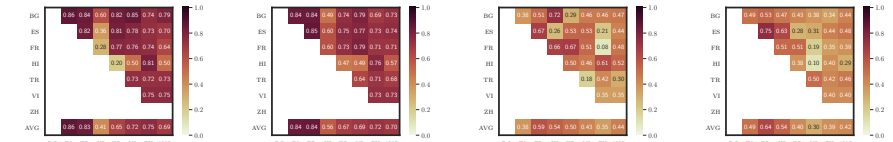


(q) TED, $\varepsilon = 30$, $l = 0$ (r) TED, $\varepsilon = 30$, $l = 8$ (s) WM, $\varepsilon = 30$, $l = 0$ (t) WM, $\varepsilon = 30$, $l = 8$

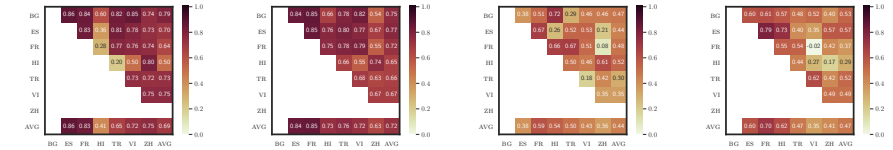


(u) TED, $\varepsilon = \infty$, $l = 0$ (v) TED, $\varepsilon = \infty$, $l = 8$ (w) WM, $\varepsilon = \infty$, $l = 0$ (x) WM, $\varepsilon = \infty$, $l = 8$

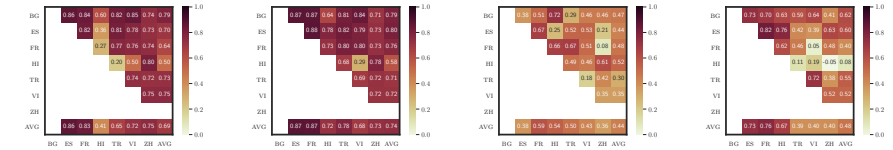
Figure 5.13: **XNLI** CKA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.



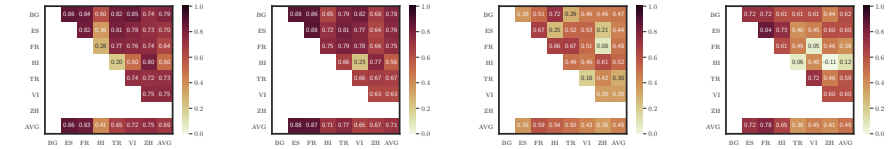
(a) TED, $\varepsilon = 1$, $l = 0$ (b) TED, $\varepsilon = 1$, $l = 8$ (c) WM, $\varepsilon = 1$, $l = 0$ (d) WM, $\varepsilon = 1$, $l = 8$



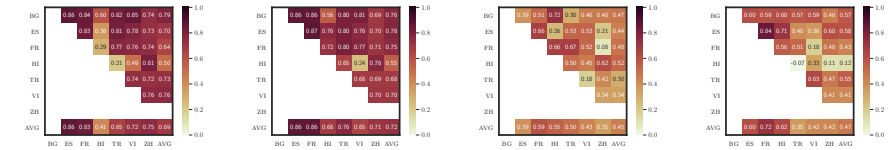
(e) TED, $\varepsilon = 3$, $l = 0$ (f) TED, $\varepsilon = 3$, $l = 8$ (g) WM, $\varepsilon = 3$, $l = 0$ (h) WM, $\varepsilon = 3$, $l = 8$



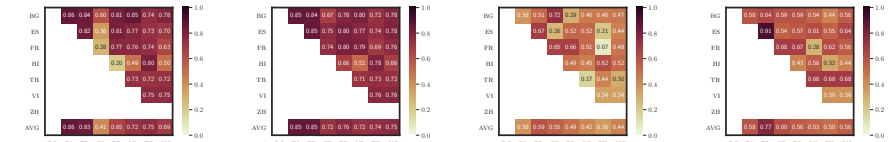
(i) TED, $\varepsilon = 8$, $l = 0$ (j) TED, $\varepsilon = 8$, $l = 8$ (k) WM, $\varepsilon = 8$, $l = 0$ (l) WM, $\varepsilon = 8$, $l = 8$



(m) TED, $\varepsilon = 15$, $l = 0$ (n) TED, $\varepsilon = 15$, $l = 8$ (o) WM, $\varepsilon = 15$, $l = 0$ (p) WM, $\varepsilon = 15$, $l = 8$



(q) TED, $\varepsilon = 30$, $l = 0$ (r) TED, $\varepsilon = 30$, $l = 8$ (s) WM, $\varepsilon = 30$, $l = 0$ (t) WM, $\varepsilon = 30$, $l = 8$



(u) TED, $\varepsilon = \infty$, $l = 0$ (v) TED, $\varepsilon = \infty$, $l = 8$ (w) WM, $\varepsilon = \infty$, $l = 0$ (x) WM, $\varepsilon = \infty$, $l = 8$

Figure 5.14: **XNLI** RSA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets (ε) and layers (l). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at l=0.

ε	TED 2020		WikiMatrix		Tatoeba	
	$l = 0$	$l = 8$	$l = 0$	$l = 8$	$l = 0$	$l = 8$
RND	0.141	0.132	0.114	0.111	0.054	0.061
PRE	0.187	0.130	0.198	0.112	0.134	0.075
1	0.188	0.054	0.199	0.046	0.135	0.033
3	0.188	0.044	0.199	0.038	0.135	0.027
8	0.187	0.045	0.197	0.038	0.133	0.027
15	0.187	0.047	0.199	0.040	0.135	0.028
30	0.187	0.047	0.199	0.040	0.135	0.028
∞	0.188	0.087	0.199	0.070	0.135	0.051

Table 5.7: **POS** IsoScores for different combinations of privacy budgets (ε) and layers (l). We show results averaged over 5 random seeds, except for RND and PRE. RND and PRE (added for comparison) denote XLM-R with randomly initialized weights and the original pretrained XLM-R, respectively. We see that the isotropy is fairly uniform across privacy budgets at layer 0 and generally higher at layer 0 than at layer 8. At layer 8, it peaks for non-private ($\varepsilon = \infty$) and our most private ($\varepsilon = 1$) models.

ε	TED 2020		WikiMatrix	
	$l = 0$	$l = 8$	$l = 0$	$l = 8$
RND	0.144	0.134	0.130	0.124
PRE	0.195	0.138	0.210	0.129
1	0.195	0.121	0.211	0.120
3	0.196	0.101	0.211	0.104
8	0.196	0.074	0.212	0.079
15	0.196	0.071	0.212	0.077
30	0.194	0.087	0.210	0.089
∞	0.195	0.182	0.211	0.166

Table 5.8: **XNLI** IsoScores for different combinations of privacy budgets (ε) and layers (l). We show results averaged over 5 random seeds, except for RND and PRE. RND and PRE (added for comparison) denote XLM-R with randomly initialized weights and the original pretrained XLM-R, respectively. We see that the isotropy is fairly uniform across privacy budgets at layer 0 and generally higher at layer 0 than at layer 8. At layer 8, it peaks for non-private ($\varepsilon = \infty$) and our most private ($\varepsilon = 1$) models.

Chapter 6

Towards Privacy-Aware Sign Language Translation at Scale

The work presented in this chapter is based on a paper that has been published as: **Phillip Rust**, Bowen Shi, Skyler Wang, Necati Cihan Camgoz, and Jean Maillard. 2024. Towards privacy-aware sign language translation at scale. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8624–8641, Bangkok, Thailand. Association for Computational Linguistics.

Abstract

A major impediment to the advancement of sign language translation (SLT) is data scarcity. Much of the sign language data currently available on the web cannot be used for training supervised models due to the lack of aligned captions. Furthermore, scaling SLT using large-scale web-scraped datasets bears privacy risks due to the presence of biometric information, which the responsible development of SLT technologies should account for. In this work, we propose a two-stage framework for privacy-aware SLT at scale that addresses both of these issues. We introduce SSVP-SLT, which leverages self-supervised video pretraining on anonymized and unannotated videos, followed by supervised SLT finetuning on a curated parallel dataset. SSVP-SLT achieves state-of-the-art finetuned and zero-shot gloss-free SLT performance on the How2Sign dataset, outperforming the strongest respective baselines by over 3 BLEU-4. Based on controlled experiments, we further discuss the advantages and limitations of self-supervised pretraining and anonymization via facial obfuscation for SLT.

facebookresearch/ssvp_slt

6.1 Introduction

Used by millions worldwide, sign languages play a crucial role in facilitating communication for many d/Deaf and hard-of-hearing individuals. Visual in nature, these languages make use of the co-articulated features of hands (i.e., finger positioning, shape, movement, palm orientation, etc.), body postures, gaze, mouth gestures, mouthings and facial expressions to convey meaning (Stokoe, 1980). Globally, there are more than 300 sign languages, each with their own grammar and vocabulary.[1] American Sign Language (ASL) alone is estimated to have more than half a million native users, ranking it among the most commonly used languages in the United States (Mitchell and Young, 2022).

Despite the prevalence of sign languages, they are still under-served by translation technology. Besides under-investment (Yin et al., 2021) and the inherent difficulty of SLT,[2] another key explanation for this imbalance is the lack of sufficiently large, clean, and labeled parallel corpora. Current state-of-the-art SLT systems require detailed and time aligned annotations (Zhou et al., 2023; Uthus et al., 2023), which is not scalable, as annotating sign language data is a labour intensive task and can only be done by proficient signers.

We argue that a promising solution to SLT's data scarcity is to utilize publicly available *unannotated* sign language data.[3] In other domains of computer vision and NLP, a common practice is to pretrain on large-scale unannotated web datasets and later finetune on curated, task-specific datasets (Devlin et al., 2019; Radford et al., 2018, 2019; Raffel et al., 2020; He et al., 2022). This practice is largely unexplored in the SLT domain and comes with additional challenges. In particular, moving to large-scale sign language processing makes it increasingly difficult to control the composition of the training data. Because sign language videos typically feature faces and upper bodies and thus are biometrically identifying, such research may exacerbate privacy risks. Hence, developing sign language technologies responsibly requires us to account for these risks and explore techniques to protect privacy.

In this work, we study the effectiveness of self-supervised video pretraining for SLT, under consideration of the aforementioned privacy risks. We first propose a *generic, scalable, and privacy-aware* two-stage framework for SLT, summarized

[1] https://www.un.org/en/observances/sign-languages-day

[2] Results of the WMT 2023 SLT task evince this difficulty; the best system only achieved ~1 BLEU (Müller et al., 2023a).

[3] For example, Uthus et al. (2023) filtered their Youtube-ASL dataset from ~88K to 11K videos based largely on the availability and quality of English captions.

Stage	I.	II.
Data		
Scale	Large	Smaller
Source	Web-mined	Hand-curated
Annotated	✗	✓
Anonymized	✓	✓/ ✗ (with explicit consent)
Training	Self-supervised	Supervised
Output	Pretrained model	Translations

Table 6.1: Our proposed generic, scalable and privacy-aware SLT framework. We make no assumptions about model architecture and anonymization method.

in Table 6.1. We introduce **SSVP-SLT** (**S**elf-**S**upervised **V**ideo **P**retraining for **S**ign **L**anguage **T**ranslation), an implementation of this framework consisting of two or optionally three stages: pretraining a continuous sign language encoder via masked autoencoding (MAE; He et al., 2022) on anonymized video, then optionally bridging the modality gap via CLIP-style video-text pretraining (Radford et al., 2021a), and finally training an SLT system via supervised finetuning using extracted features from the pretrained model. Our best performing models achieve 15.5 BLEU finetuned and 7.1 BLEU zero-shot on the How2Sign dataset (Duarte et al., 2021), surpassing SOTA in both settings by over 3 BLEU while using data anonymized via facial obfuscation. We also introduce a new ASL-to-English SLT benchmark dataset, *DailyMoth-70h*, consisting of ~70h of continuous signing in native ASL. We then evaluate the downstream performance impact and discuss the benefits and limitations of facial blurring to achieve anonymization. Through controlled ablation studies of SSVP-SLT, we identify what factors contribute to a strong pretraining and finetuning recipe. We conclude by discussing opportunities and challenges of self-supervised pretraining for sign language processing.

6.2 Background and Related Work

Gloss-free SLT Glosses are a way of representing individual signs into a written form. Being monotonically aligned to signs, they can be a useful medium between sign and spoken languages. Most SLT approaches to date rely on them (Chen et al., 2022a,b; Zhang et al., 2023). The task of predicting glosses from continuous signing is typically performed via gloss supervision either jointly or in a cascaded approach with supervised SLT finetuning (Camgöz et al., 2018; Cihan Camgoz

et al., 2020).

However, glosses are also considered an incomplete and inaccurate representation of sign language (Cihan Camgoz et al., 2020; Müller et al., 2023b). Furthermore, gloss annotation is a labour intensive task. Due to these constraints, there is a growing body of research on gloss-free SLT. Most such approaches incorporate techniques aimed at reducing the modality gap, such as training the visual encoder via sign spotting (Tarrés et al., 2023; Shi et al., 2022),[4] adding inductive bias in the attention mechanism (Yin et al., 2023), using conceptual anchor words (Lin et al., 2023), or performing visual-language pretraining (Zhou et al., 2023). Uthus et al. (2023) also benefit from a pretrained text model (T5; Raffel et al., 2020). Similar to Zhou et al. (2023), we also leverage language supervision to reduce the modality gap, albeit in conjunction with self-supervised video pretraining.

Sign Language Video Anonymization Sign language videos typically feature signers' faces, which convey essential linguistic information. However, in virtual domains, particularly in spaces involving the discussion of sensitive topics, exposing one's face (and identity) may lead to various forms of personal risks. Such exposures could even lead to harm associated with professional or insurance discrimination. For these reasons, the d/Deaf and hard-of-hearing community has long expressed interest in anonymization and privacy protection techniques for sign language content (Lee et al., 2021), and sign language video anonymization has, in recent years, become an active area of research (Isard, 2020; Xia et al., 2024).

For general-domain images and videos, simple anonymization techniques such as facial obfuscation via overlaying or blurring are widely used and accepted (Frome et al., 2009; Yang et al., 2022a). In the sign language domain, such techniques may be inadequate due to the loss of information in the facial region (Lee et al., 2021). More specifically, in signed language, non-manual features such as mouthing, eyebrow and head movements are used extensively to enrich grammar. Certain signs with similar manual features are only disambiguated through mouth morphemes. Moreover, facial expressions are often used to indicate emphasis, negation, a question, etc. (Baker-Shenk, 1985; Valli and Lucas, 2000; Neidle et al., 2000).

[4]While Tarrés et al. (2023) do not explicitly mention the use of sign spotting, they rely on features extracted from an I3D model (Carreira and Zisserman, 2017) by Duarte et al. (2022), who used an iterative sign spotting technique.

Recent work has focused on anonymizing sign language content via avatars (Tze et al., 2022; Lee et al., 2021) and transferred or synthetic human appearances (Saunders et al., 2021; Xia et al., 2022, 2024). While promising, these approaches are nascent and we are unaware of studies that determine to what extent models can learn to recover or disambiguate obfuscated linguistic information from context. That being said, human studies suggest that signers struggle to comprehend content anonymized in such a way (Lee et al., 2021).

Lacking an obvious alternative, in this work we return to the relatively straightforward technique of facial blurring. Despite its limitations, we demonstrate that blurring can raise privacy protection with little performance degradation.[5] This, we argue, can facilitate large-scale video anonymization when applied to publicly available sign language data.

Masked Autoencoding for Video and Beyond Following its success as a self-supervised learning paradigm in the image domain (He et al., 2022), MAE has been widely applied in other areas, including audio (Huang et al., 2022), language (Rust et al., 2023), and video (Feichtenhofer et al., 2022; Tong et al., 2022; Wang et al., 2023b). Considering that MAEs have been shown to be capable of acquiring both language and basic video understanding from pixels alone, it is conceivable that high-quality sign language representations can be learned directly from RGB video data via MAE, given enough data. Recently, Sandoval-Castaneda et al. (2023) explored MAE among other self-supervised learning techniques in the context of isolated sign language recognition (ISLR) and found it to be highly useful. MAE has, however, not yet been successfully applied to SLT. In SLT, videos are much longer, and learning high-quality representations requires models to capture long-range spatiotemporal dependencies. Our usage of MAE, or self-supervised pretraining in general, therefore stands in contrast to recent SLT methods, gloss-based and gloss-free methods alike, which instead fully rely on supervised training that requires annotated captions (Zhou et al., 2023; Tarrés et al., 2023; Lin et al., 2023; Uthus et al., 2023).

6.3 Generic Framework

We first outline a generic, scalable and privacy-aware two-stage transfer learning framework for SLT (see Table 6.1).

[5] In Appendix 6.8.1 we discuss issues with pose landmarks, often promoted as a privacy-preserving alternative to video.

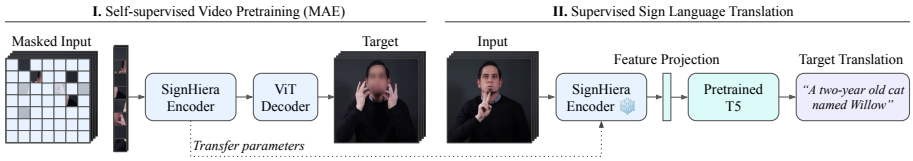


Figure 6.1: Overview of our two-stage SSVP-SLT method. The first stage consists of training a SignHiera encoder via masked autoencoding (MAE) on *blurred* video frames. In the second stage, a pretrained T5 model is finetuned for SLT while the pretrained SignHiera is kept frozen (❄). The input video in the second stage *can be unblurred*.

In **Stage I**, we train a model with the goal to learn high-quality continuous sign language representations via self-supervised learning. The data used at this stage is *always anonymized*. We make no assumptions on how the data may be anonymized, i.e., face blurring as discussed in § 6.2, or more sophisticated methods such as using synthetic appearances.

In **Stage II**, we finetune the model from the first stage in a supervised manner using a smaller and hand-curated parallel dataset. Ideally, the finetuning dataset, being manageable in size, can be unanonymized after gaining explicit consent from signers in the data to minimize information loss.

6.4 Method

The base implementation of our framework is designed as a two-step approach, termed **SSVP-SLT**. We provide a high-level overview in Figure 6.1.

Self-Supervised Video Pretraining (MAE) We first aimed to pretrain a capable sign language encoder on video data alone—no gloss, pseudo-gloss, or spoken-language text annotations—allowing us to leverage large amounts of unannotated sign language video in the future, alleviating the data scarcity issue in SLT training.

Our sign video encoder, *SignHiera*, builds on Hiera (Ryali et al., 2023), a vision transformer that combines a hierarchical architecture, shown to be crucial for learning phonetically meaningful sign representations (Sandoval-Castaneda et al., 2023) with masked autoencoding (MAE), a widely used self-supervised learning paradigm (He et al., 2022). Its hierarchical architecture also makes Hiera more efficient to train than other MAE-based video transformers such as VideoMAE

(Tong et al., 2022; Wang et al., 2023b) or MAE-ST (Feichtenhofer et al., 2022).

Hiera embeds a video into a sequence of spatio-temporal tokens. A large percentage of tokens is randomly masked, while the rest is passed through a hierarchical transformer stack with several pooling operations. The decoder receives fused multi-scale features extracted before each pooling operation and processes them via a lightweight transformer stack. A final linear projection yields pixel logits. The loss is computed as the normalized mean squared error between the original and predicted pixel values of the masked tokens.

SignHiera is initialized from the original Hiera-Base-16×224 checkpoint pretrained on Kinetics-400 (Kay et al., 2017). In order to capture longer-range spatio-temporal dependencies in signed utterances, we increased the clip length from 16 to 128 video frames, leading to an 8× sequence length, and accordingly resized and reinitialized the position embeddings. We further added attention masking to accommodate shorter videos with temporal padding and added a third Q-pooling operation after the last encoder stage to save GPU memory. We trained with a masking ratio of 0.9.

Supervised SLT Finetuning The translation network is an encoder-decoder transformer model (Vaswani et al., 2017). Our default configuration uses a pretrained T5-v1.1 (Raffel et al., 2020; Shazeer, 2020), following Uthus et al. (2023). We also experimented with BART (Lewis et al., 2020) and Tarrés et al. (2023)'s setup, training a ~10M parameter transformer from scratch.

The only difference from a text transformer is that our translation network takes in video features extracted from the pretrained SignHiera. We used SignHiera's final intermediate representations, which are of size $\mathbb{R}^{B\times\frac{T}{2}\times\frac{H}{32}\times\frac{W}{32}\times D}$, where B is the batch size, $T=128$, $H=W=224$ is the input video size, and $D=768$ is the feature size. We mean-pooled over the spatial dimensions to obtain feature vectors of size $\mathbb{R}^{B\times\frac{T}{2}\times D}$. Videos shorter than 128 frames were padded for encoding, and the padding was then removed from the extracted features. For longer videos, we used a sliding window with stride $\frac{T}{2}$ and concatenated the resulting features. A linear projection $W_{\text{proj}} \in \mathbb{R}^{D\times D'}$ mapped the extracted features to a sequence of size $\mathbb{R}^{B\times S\times D'}$, with S being the sequence length of the extracted features and D' the transformer's hidden size. This sequence was processed by the transformer as usual (Vaswani et al., 2017).

Adding Language-supervised Pretraining We also experimented with extending SSVP-SLT with a language-supervised pretraining (LSP) step to bridge the

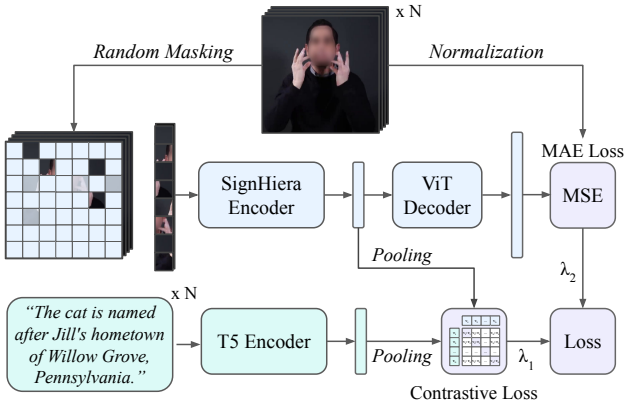


Figure 6.2: Overview of our LSP extension.

modality gap between the input videos and text translations. Bridging this gap may improve gloss-free SLT performance, as discussed in § 6.2. In what follows, we refer to SSVP-SLT with an additional LSP stage as **SSVP-SLT-LSP**. The LSP stage fits *in between* the self-supervised MAE pretraining and the supervised SLT finetuning stage. Since language supervision requires annotations, the LSP stage should be considered a part of stage II in our generic framework.

Our LSP approach is illustrated in Figure 6.2. We first initialized a CLIP model (Radford et al., 2021a) with our MAE-pretrained SignHiera encoder as the vision tower and a pretrained T5-v1.1 encoder as the text tower. We then jointly trained the CLIP model via contrastive video-text pretraining and the SignHiera model via MAE. The goal is to help the SignHiera encoder, which is involved in both tasks, learn strong continuous sign representations grounded in the target modality (text). The videos were masked with a 90% ratio even for computing the contrastive loss, which is similar to FLIP (Li et al., 2023d), and enabled end-to-end training by drastically reducing the memory footprint. The two losses (MAE and contrastive) were balanced via GradNorm (Chen et al., 2018), which helped stabilize training, compared to using fixed loss weights.[6]

[6]In contrast to Zhou et al. (2023), we did not jointly train the text decoder as doing so did not improve performance in preliminary experiments and led to training instabilities.

6.5 Experimental Setup

6.5.1 Datasets

Youtube-ASL (YT) We used Youtube-ASL (Uthus et al., 2023), the largest available ASL training dataset with ~1000h of in-the-wild video and over 2500 signers, both during pretraining and supervised finetuning.[7] We used a version in which all signers' faces are blurred for privacy.

How2Sign (H2S) We also used How2Sign (Duarte et al., 2021), an ASL dataset with ~80h of video and nine different signers in a green screen studio setting, for pretraining, finetuning, and downstream evaluation of our SLT models. Again, we used a version with blurred faces only.

DailyMoth-70h To isolate the impact of face blurring during pretraining and finetuning on SLT performance, we relied on a new dataset, which we name *DailyMoth-70h*. This dataset contains over 70h of video of a single signer from the ASL news page TheDailyMoth and was obtained through a license agreement with TheDailyMoth's host.[8] We used both unblurred and blurred dataset versions and report dataset statistics in Appendix 6.8.2.

6.5.2 Training and Evaluation Protocols

We briefly describe our training and evaluation protocols. The full configurations for pretraining and SLT training are listed in Appendix 6.8.3.1.

Face blurring We used an internal face blurring software and ensured its reliability on the YT and H2S datasets via a combination of automatic and manual verification techniques. Example frames sampled from two blurred videos from the DailyMoth-70h data are shown in Appendix Figure 6.5.

MAE pretraining We largely follow the Hiera pretraining recipe from Ryali et al. (2023). In our default configuration, we trained for 800 effective epochs (accounting for repeated sampling as in Feichtenhofer et al. (2022)) with the AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019). The

[7] For the lack of a readily available larger, unannotated dataset, Youtube-ASL fits both dimensions of our framework: the large, publicly-available, unannotated dataset and the curated, parallel dataset.

[8] https://www.dailymoth.com/

Method	Blur	FT Data	BLEU-1	BLEU-2	BLEU-3	BLEU	ROUGE-L	BLEURT
Baselines								
Lin et al. (2023)	✗	H2S	14.9	7.3	3.9	2.2	12.6	31.7
Tarrés et al. (2023)	✗	H2S	34.0	19.3	12.2	8.0	—	—
Uthus et al. (2023)	✗	H2S	15.0	5.1	2.3	1.2	—	30.0
		YT	20.9	10.4	6.1	4.0	—	35.0
		YT + H2S	36.3	23.0	16.1	11.9	—	44.8
		YT → H2S	37.8	24.1	16.9	12.4	—	46.6
Ours								
SSVP-SLT $^{\text{H2S}}_{800}$	✓	H2S	30.2	16.7	10.5	7.0	25.7	39.3
SSVP-SLT $^{\text{YT}}_{800}$	✓	H2S	38.1	23.7	16.3	11.7	33.8	44.2
		YT	29.2	16.6	10.7	7.1	28.3	41.8
		YT + H2S	41.6	27.2	19.3	14.3	36.8	48.6
		YT → H2S	41.9	27.7	19.8	14.7	37.8	49.3
SSVP-SLT $^{\text{YT+H2S}}_{800}$	✓	H2S	38.9	24.1	16.4	11.6	34.0	44.5
		YT	29.1	17.0	11.1	7.5	28.6	41.6
		YT + H2S	41.8	27.4	19.5	14.3	36.9	48.9
		YT → H2S	41.8	27.4	19.6	14.6	37.7	49.0
SSVP-SLT-LSP $^{\text{YT+H2S}}_{600\rightarrow200}$	✓	YT + H2S	**43.2**	**28.8**	**20.8**	**15.5**	**38.4**	**49.6**

Table 6.2: How2Sign test performance of SSVP-SLT in different pretraining configurations compared to baselines. The Blur column denotes whether faces in the train and eval data are blurred. FT Data indicates the finetuning configuration; respectively, YT+H2S and YT→H2S refer to training on the two datasets jointly or consecutively.

learning rate was set to $8e{-}4$ with linear warmup over 120 epochs and cosine decay to $1e{-}5$. The batch size was 2×128, with 2 being the repeated sampling factor. Similar to Zhou et al. (2023), we employed video data augmentation via random cropping, horizontal flipping, and RandAug (Cubuk et al., 2020). We used a 128×2 temporal sampling strategy, i.e., sampling 128 frames with a stride of 2, which fully accommodates ~85–95% of videos in the data.

SLT finetuning When finetuning only on How2Sign or DailyMoth-70h, we closely followed the setup of Tarrés et al. (2023), training a ~10M parameters transformer from scratch; see Appendix 6.8.3.1 for more details. For How2Sign, we reused their lowercase unigram tokenizer (vocab size 7K). For DailyMoth-70h, we trained a cased tokenizer (unigram, 7K vocab size), which we found to work better due to the large proportion of named entities in the data.

When finetuning on Youtube-ASL, as we needed a model with more capacity we relied on a pretrained T5-v1.1 with default tokenizer, following Uthus et al.

(2023). We trained for up to 100 epochs with early stopping, batch size 512 and AdamW with peak rate $5e{-}4$, linear warmup over two epochs and cosine decay to $1e{-}7$. We used dropout of 0.3 and 0.2 label smoothing in the cross-entropy loss. We did not use video data augmentation unless specified otherwise.

Language-supervised pretraining We performed 200 epochs of LSP with a batch size of 512 on top of 600 epochs of MAE-only pretraining. We did not use repeated sampling, which is incompatible with the contrastive loss. We again used AdamW, warming up to a learning rate of $1e{-}4$ over ten epochs, followed by cosine decay to $1e{-}6$. The GradNorm optimizer has a one epoch warmup, a peak learning rate of $1e{-}2$, and decays to $1e{-}4$. Data augmentation and temporal sampling are the same as for MAE pretraining.

Evaluation We used beam search with 5 beams and no length penality. We evaluated every epoch when finetuning on How2Sign or Dailymoth-70h and every 500 steps for Youtube-ASL. We kept the checkpoint with the highest validation BLEU-4 and evaluated it on the respective test set.

Notation Below, we use superscript and subscript to indicate the pretraining dataset and number of epochs, respectively. For instance, SSVP-SLT $^{\text{YT+H2S}}_{800}$ refers to 800 epochs of MAE pretraining on Youtube-ASL and How2Sign. For SSVP-SLT-LSP, 600→200 denotes 600 epochs of MAE pretraining followed by 200 epochs of LSP.

6.5.3 Baselines

Lin et al. (2023) propose to bridge the visual and text modalities via contrastive anchoring of encoded visual features to embeddings of conceptual words in the target sequence. **Tarrés et al. (2023)** is the SOTA on How2Sign without additional SLT data, training a 6+3 layer transformer from scratch on features from an I3D model (Carreira and Zisserman, 2017). The I3D model was first pretrained on Kinetics (Carreira and Zisserman, 2017) and BOBSL (Albanie et al., 2021), and finetuned on How2Sign for sign language recognition using annotations generated via iterative sign spotting (Duarte et al., 2022). **Uthus et al. (2023)** is the current SOTA on How2Sign, and finetunes a pretrained T5-v1.1-Base model for SLT directly on pose landmarks extracted from YouTube-ASL and How2Sign videos.

6.5.4 Evaluation Measures

We report BLEU via SacreBLEU (Papineni et al., 2002; Post, 2018).[9] We also report ROUGE-L (Lin, 2004) and BLEURT (Sellam et al., 2020) from the BLEURT-20 checkpoint, which has been shown to correlate well with human judgments.

6.6 Results and Discussion

Comparison against the state-of-the-art We present our main results in Table 6.2. Our best models significantly improve over the previous 12.4 BLEU state-of-the-art by Uthus et al. (2023). SSVP-SLT achieves 14.7 and 14.6 BLEU when pretraining on YT and YT+H2S respectively. Our best overall model, utilizing SSVP-SLT-LSP, achieves 15.5 BLEU, a 3.1 point improvement over the baseline. When pretraining and finetuning on YT only, we also observe a 3.1 BLEU improvement (4.0 vs. 7.1) over the previous SOTA in the zero-shot setting. These results demonstrate the overall effectiveness of SSVP-SLT and, more broadly, self-supervised pretraining for SLT.

Pretraining on YT+H2S performs almost the same as training on YT only, with the YT-only models even sometimes performing best. Given the distributional gap between the datasets (in-the-wild YT vs. studio H2S video) and the fact that the YT+H2S models consumed more data, this finding is somewhat surprising. While this may be due in part to randomness in the training dynamics, it could also mean that sufficient finetuning can compensate for not accessing the H2S data at pretraining, presumably because the pretraining set is sufficiently large and diverse. This encouraging result suggests that we can pretrain on large data independently of knowing what our finetuning and inference dataset will be—a crucial requirement for practical SLT applications.

We find that YT data is beneficial both for pretraining and finetuning, which emphasizes the importance of training on large and diverse data and suggests that we can expect further gains from scaling to large public unannotated video.

Finally, we find that bridging the modality gap via language-supervised pretraining yields a 1.2 BLEU improvement over its MAE-only counterpart. Given enough annotated data, the technique can be employed independently of self-supervised pretraining at little extra cost.

[9] `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1`

Blur Pretrain	Blur SLT	BLEU	ROUGE-L	BLEURT
✗	✗	**28.8**	**50.9**	**51.7**
✗	✓	28.1	50.6	51.4
✓	✗	28.2	50.3	51.0
✓	✓	27.5	49.6	50.4

Table 6.3: Performance on *unblurred* test data for SSVP-SLT trained and evaluated on DailyMoth-70h with or without facial blurring during pretraining and SLT.

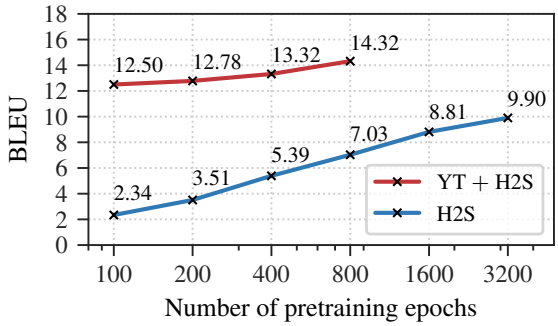


Figure 6.3: How2Sign test BLEU of SSVP-SLT after pretraining on YouTube-ASL and How2Sign or How2Sign only and finetuning on the same data.

What's the effect of blurring? We isolate the impact of facial obfuscation via blurring on SLT performance by training SSVP-SLT models on DailyMoth-70h with and without blurring during pretraining and SLT training. We report the results in Table 6.3. As expected, performance is best when not blurring (28.8 BLEU) and worst when blurring at finetuning time (28.1 and 27.5 BLEU). Crucially, some performance can be recovered after pretraining on blurred data when performing SLT on unblurred data (28.2 BLEU), validating our proposed framework (see Table 6.1).[10] This means we can pretrain in a privacy-aware manner without sacrificing too much performance.

How long should you pretrain? The sign language MAE task is intricate. The model first needs to abstract away surface-level information such as background and signer appearance. It then needs to implicitly acquire an understanding

[10]We hypothesize that even more performance could be recovered if the SignHiera video encoder was unfrozen during SLT training, allowing adaptation to the facial information.

Clip size	Pretraining epochs	BLEU	ROUGE-L	BLEURT
		H2S		
128	800	**7.0**	**25.7**	**39.3**
128	100	2.3	14.3	33.6
16	800	5.0	20.2	36.4
		YT + H2S		
128	800	**14.3**	**36.9**	**48.9**
128	100	12.5	34.2	46.1
16	800	10.4	31.7	44.3

Table 6.4: How2Sign test performance of SSVP-SLT when pretraining on (YouTube-ASL and) How2Sign with a clip size of 16 versus 128 video frames.

of ASL, capturing long-range dependencies in the video. It is therefore worth investigating basic scaling properties. In Figure 6.3, we show downstream SLT performance over the course of pretraining. Similar to Ryali et al. (2023) and He et al. (2022), we observe consistent downstream improvements as pretraining progresses, suggesting that the models are not overfitting to the training data even after extensive pretraining. These results underscore the task's effectiveness and indicate that further scaling would likely yield additional gains.

Is encoding longer clips necessary? Increasing clip length is costly due to the poor scaling of global attention, raising the question of whether encoding longer clips is needed. Our results indicate that the answer is *yes*. Table 6.4 compares the performance of the original 16-frame Hiera with our 128-frame SignHiera. While 16-frame Hiera achieves non-trivial performance after 800 epochs, it is substantially outperformed by 800 epoch SignHiera (7.0 vs 5.0 BLEU for H2S and 14.3 vs 10.4 BLEU for YT+H2S). This may be partially explained by the fact that SignHiera sees up to 8× as much data every step. However, we also see that 800-epoch 16-frame Hiera lags far behind even a 100-epoch SignHiera (which has seen roughly the same number of tokens) when training on a large dataset (12.5 vs 10.4 BLEU in the YT+H2S setting). When training on less data (H2S), 16-frame Hiera is still worse than 400-epoch SignHiera (5.39 vs 5.0 BLEU, see Figure 6.3). Overall, this suggests that certain information cannot easily be acquired from shorter clips.

Model	Param	PT	BLEU	ROUGE-L	BLEURT
BART	140M	✗	14.0	36.8	48.3
		✓	13.5	36.2	48.1
T5-v1.1	248M	✗	11.6	35.0	46.1
		✓	**14.3**	**36.9**	**48.9**

Table 6.5: How2Sign test performance of SSVP-SLT $^{\text{YT+H2S}}_{800}$ when finetuning BART and T5, initialized randomly (PT = ✗) or from the pretrained model (✓).

Aug	BLEU	ROUGE-L	BLEURT
✗	14.3	36.9	48.9
✓	**14.7**	**37.2**	**49.0**

Table 6.6: How2Sign test performance of SSVP-SLT $^{\text{YT+H2S}}_{800}$ with and without finetuning augmentation.

How to choose the text model? We investigate how the architecture and initialization of the text transformer affect performance. Table 6.5 compares pretrained and randomly initialized BART (Lewis et al., 2020), the English monolingual counterpart to mBART (Liu et al., 2020), which has previously been successfully adapted for German and Chinese sign languages (De Coster et al., 2021; Chen et al., 2022a; Zhou et al., 2023), and T5-v1.1 as used by Uthus et al. (2023).

Overall, T5 outperforms BART, possibly due to larger capacity, but the gap is small. While it is worse to finetune a randomly initialized T5 model compared to the pretrained one (corroborating findings by Uthus et al. (2023)), for BART we find the opposite result. We conclude that whether text pretraining is helpful needs to be evaluated on a case-by-case basis. It may be worth investigating in the future whether an additional pretraining or finetuning step may lead to better adaptation of the text model to sign language.

Should we augment data at finetuning? Augmentation such as flipping, cropping, and RandAug may improve generalization, but it comes at a high storage cost at finetuning time, as the video features are extracted offline. Is it worth the cost? We compared using 60 epochs of augmented videos (a 60-fold increase in storage) with not using any augmentation. The results in Table 6.6 show that using augmentation yields a reasonable 0.4 BLEU gain, suggesting that

Initialization	MAE	CLIP	BLEU	ROUGE-L	BLEURT
Hiera $^{\text{K400}}_{800}$	✗	✓	2.1	14.9	35.0
SSVP-SLT $^{\text{YT+H2S}}_{600}$	✗	✓	11.0	32.1	44.7
	✓	✗	14.3	36.9	48.9
	✓	✓	**15.5**	**38.4**	**49.6**

Table 6.7: How2Sign test performance when including (✓) or removing (✗) the MAE and CLIP objectives and pretraining from the original Hiera $^{\text{K400}}_{800}$ or SSVP-SLT $^{\text{YT+H2S}}_{600}$ checkpoint for 200 epochs on YT+H2S, followed by finetuning on the same data.

augmentation can be useful when storage is not a major concern.

Are both pretraining objectives necessary? In § 6.6, we saw that language-supervised video-text pretraining is highly effective when combined with MAE. Are both necessary? We compared pretraining for 200 epochs with either and both objectives, initializing from a 600-epoch SSVP-SLT checkpoint, and also performed 200 epochs of CLIP-only pretraining from the original pretrained Hiera. The results in Table 6.7 show that removing either objective results in a performance drop. The drop is larger when removing MAE, indicating its continued importance after 600 epochs of MAE-only training. Initializing from the original Hiera results in very poor performance (2.1 BLEU), suggesting that language-supervised pretraining alone is not useful in our setting. Considering that language supervision has previously been shown to be effective in isolation (Zhou et al., 2023), this may be due to the FLIP-style masking and the fact that we do not jointly pretrain T5. We also emphasize that language-supervised pretraining falls under stage II of our framework as it requires annotations; it can, therefore, only serve as an addition to self-supervised pretraining, but not a replacement.

6.7 Conclusion

Through controlled experiments, we studied the effectiveness of self-supervised pretraining for SLT while considering privacy risks. We introduce SSVP-SLT, a novel, scalable, and privacy-aware SLT method that leverages masked autoencoding on anonymized video. It achieves state-of-the-art ASL-to-English translation performance on the How2Sign benchmark, outperforming the best previous

models in the finetuned and zero-shot setting by over 3 BLEU.

Our results demonstrate the promise of self-supervised learning to alleviate the data scarcity issue and further scale up sign language processing in the future. We found that video anonymization, even via simple techniques such as face blurring, has relatively little negative impact on downstream performance, further proving that we can build more proficient systems without neglecting important privacy concerns. We hope that this work, alongside the code and data we release, will spur future developments that benefit the d/Deaf and hard of hearing communities.

Limitations

Compute Requirements Currently, SSVP-SLT requires access to substantial compute to train at the scale of Youtube-ASL (600K videos). This is primarily due to the high dimensionality of video data, exacerbated by the long clip length and information density in sign language content, which creates a data-loading bottleneck and increases the memory footprint, especially in combination with a transformer architecture. Our longest pretraining run in full precision (fp32) took approximately two weeks on 64 A100 GPUs. We believe that it will be important to drive down this cost in the future and make large-scale video pretraining more accessible. While many simple interventions, such as mixed precision, gradient accumulation, and gradient checkpointing, could drastically reduce the memory footprint, they usually come at the cost of training time or stability. In general, we note that this limitation is not unique to our approach but often not apparent due to training being conducted on nearly 100× smaller datasets such as RWTH-Phoenix-Weather 2014T (7K videos; Camgöz et al., 2018).

Anonymization We rely on face blurring for video anonymization, which is known to incur a loss of linguistic information (see § 6.2). In the future, it will be worth investigating more sophisticated methods, such as using synthetic appearances. Also, largely due to a lack of linguistic tools for continuous signing, we did not investigate what effects anonymization may have on the translations *qualitatively*. For instance, it would be instructive to know whether the model successfully disambiguates certain phonemes with similar manual features through context in the absence of facial information.

Languages Due to the availability of sufficiently large datasets for our pretraining experiments, we only experiment with American Sign Language and English, the two highest-resource signed and spoken languages. We aim to diversify this language selection in the future.

Ethics Statement

Regarding performance, our models may contain demographic biases and underperform for certain race, gender, and age groups. For instance, even though the YouTube-ASL dataset (a dataset we used for pretraining and supervised finetuning) features over 2500 signers, the authors did not provide demographic details of these signers. Similarly, our DailyMoth-70h dataset includes only one signer (white, male, and early middle-aged). As such, it is unclear how our models perform for underrepresented users, who, aside from having diverse identities, may introduce different accents or regional variations of ASL that our models do not adequately capture. We call for future research on SLT to be more explicit about documenting demographic biases in their datasets and models.

Lastly, we emphasize that anonymization inherently does not offer any formal privacy guarantees in contrast to frameworks such as differential privacy (Dwork, 2006), which fundamentally comes at a (often substantial) cost in utility (Geng et al., 2020). As such, while our work (and the use of facial obfuscation in general) represents an important first step towards comprehensively protecting the privacy of signers, it should not be relied on in sensitive or high-stakes applications.

Acknowledgements

We thank members of the Seamless Communication team at FAIR for helpful feedback and discussions throughout this project. We also thank Florian Metze and Amanda Duarte for their prompt assistance with our questions about the How2Sign dataset.

6.8 Appendix

6.8.1 Pose landmarks vs RGB Video

Pose landmarks (e.g., from MediaPipe Holistic) are cited as a privacy-preserving alternative to RGB video for SLT (Moryossef et al., 2021; Uthus et al., 2023).

	Train	Validation	Test	Full
Raw data				
Number of signers	—	—	—	1
Number of videos	—	—	—	496
Video duration (hours)	—	—	—	76.9
Time frame	01/21–04/23	02/22–04/23	02/22–04/23	01/21–04/23
Segmented data				
Number of clips	41 412	2789	4185	48 386
Clip duration (hours)	65.8	4.0	6.0	75.8
Vocabulary (words)	18 495	4803	6040	19 694
Duration in seconds per clip*	5.7 / 2.4 / 8.9	5.2 / 2.1 / 7.9	5.2 / 2.1 / 7.9	5.6 / 2.3 / 8.7
Characters per caption*	43.9 / 12.7 / 58.0	44.1 / 12.8 / 59.0	43.3 / 12.9 / 59.0	43.9 / 12.7 / 59.0
Words per caption*	8.6 / 2.4 / 12.0	8.7 / 2.4 / 12.0	8.5 / 2.4 / 12.0	8.6 / 2.4 / 12.0

Table 6.8: DailyMoth-70h dataset statistics. (*): mean/std/90$^{\text{th}}$ percentile

While they may indeed offer benefits in terms of efficiency and generalization, we argue that they do not offer meaningful privacy protection either. For instance, using a sufficiently large number of facial landmarks that are estimated accurately results in what is essentially a scan of the facial geometry, a biometric identifier according to legislation like the Biometric Information Privacy Act (BIPA).[11] Despite abstracting away shallow information about a person's appearance, pose information could, therefore, be (mis)used in similar ways as de-anonymized video. Analogous to facial obfuscation in video, one could reduce the number of facial landmarks or add noise to them to hinder re-identification, but doing so also results in (arguably even more) loss of information.

6.8.2 DailyMoth-70h Dataset

We introduce *DailyMoth-70h*, a dataset containing over 70h of video with aligned English captions of a single native ASL signer (white, male, and early middle-aged) from the ASL news page TheDailyMoth.[12] We obtained the data via a license agreement with the host of TheDailyMoth.

Download and License The fully self-contained dataset is available under a CC BY-NC license at facebookresearch/ssvp_slt.

Statistics We provide detailed dataset statistics in Table 6.8 and Figure 6.4.

[11] https://www.ilga.gov/legislation/ilcs/ilcs3.asp?ActID=3004&ChapterID=57
[12] https://www.dailymoth.com/

Purpose The dataset is fully self-contained and can therefore serve as a new benchmark for studying the task of single-signer translation (e.g., for building personalized systems). Furthermore, sign language translation involves overcoming several challenges such as generalizing over signers, their appearances and their signing styles as well as mapping spatio-temporal signed utterances to spoken words. DailyMoth-70h can be used to disentangle some of these challenges by eliminating the signer and style variances and allow researchers to ablate their models more focused on the sign-to-spoken language mapping.

Preprocessing We first segmented the raw videos into clips of ~5.6 seconds on average based on their aligned English captions. Each entry in the SubRip subtitle (SRT) file, which maps video timestamps to captions, was chosen to be a distinct datapoint. Accordingly, example clips are often sentence fragments rather than full sentences.

Afterwards, the segmented video clips were automatically cropped such that the signer is approximately in the center of the frame and resized to 224×224 pixels. The preprocessed clips were saved in their native framerates, which are either 24 or 30 fps.

Next, many videos had burned-in captions which, if not removed, would reduce the translation task to a simple OCR task. We, therefore, used an off-the-shelf text detection model to identify burned-in captions in the videos, and blurred the captions conservatively. Although the blurring may be imperfect due to errors made by the text detector, this intervention should nevertheless solve the concern of models shortcutting the SLT task for the most part.

Finally, we split the data into training, validation, and test splits. The proportions (85% / 6% / 9%) were chosen to approximately match How2Sign. The validation and test examples were randomly sampled from the subset of videos posted after January 2022, which avoids data leakage from Youtube-ASL or OpenASL (Shi et al., 2022), both of which have cut-off dates before/during January 2022, into the DailyMoth-70h evaluation splits. The training examples were randomly sampled from the full range of dates (January 2021 to April 2023).

6.8.3 Reproducibility

6.8.3.1 Model and Training Configurations

We report our pretraining configurations for SSVP-SLT in Table 6.9 and SSVP-SLT-LSP in Table 6.10. Our finetuning configurations are listed in Table 6.11 for Youtube-ASL (+ How2Sign) and Table 6.12 for How2Sign-only and DailyMoth-70h.

6.8.3.2 Code

Our implementation uses Python 3.10 and PyTorch 2.0.1 (Paszke et al., 2019) compiled with CUDA 11.7. The code is available under a CC BY-NC license at facebookresearch/ssvp_slt.

6.8.3.3 Hardware & Runtime

We ran our experiments on NVIDIA A100 80GB and V100 32GB GPUs. On Youtube-ASL (+ How2Sign), pretraining took ~20 minutes (SSVP-SLT) / 30 minutes (SSVP-SLT-LSP) per effective epoch on 64 A100 or 128 V100 GPUs. On How2sign or DailyMoth-70h, an effective epoch of SSVP-SLT pretraining took ~3 minutes. Finetuning and evaluating T5 and BART on Youtube-ASL (+ How2Sign) took ~8 and 4 minutes, respectively, per epoch on 32 V100 GPUs. Training T5 was slower due to training in full precision as opposed to fp16 and using a smaller batch size with gradient accumulation. Finetuning and evaluating with Tarrés et al. (2023)'s setup on How2Sign or DailyMoth-70h took ~1 minute per epoch on a single V100 GPU.

6.8.4 Qualitative Examples

In Table 6.13, we provide qualitative examples of our best-performing model (15.5 BLEU on How2Sign), compared against the best-performing models from Tarrés et al. (2023), Uthus et al. (2023), as well as the reference translations. The examples were picked from the How2Sign test split by Tarrés et al. (2023). Examples (3)–(5) are, anecdotally, more challenging than the average test example. We find that our model is mostly on-topic and matches the syntactic structure, although it can still struggle with repetitions and the mixing-up of signs. Our model's failure patterns are more similar to Uthus et al. (2023)'s models—possibly a result of finetuning the same base model (T5-v1.1) on the same

datasets (Youtube-ASL and How2Sign). For instance, in example (3), both models mistakenly predict the verb “feed” (and mispredict everything that comes after) but flawlessly match the syntactic structure of the reference translation. Overall, both baselines appear to exhibit a higher occurrence of (complete) mis-translation, which aligns with our quantitative results.

Parameter	**Value**
Decoder blocks	8
Decoder heads	8
Mask ratio	0.9
Drop path rate	0.2
Video size (T, C, H, W)	(128, 3, 224, 224)
Sampling Rate	2
Face Blur	✓
Random Crop	✓
Horizontal Flip	✓
RandAug	✓(4, 7)
Repeated sampling	2
Optimizer	AdamW
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.95$
Weight decay	0.05
Peak learning rate	$8e{-}4$
Learning rate schedule	Cosine Decay
Minimum learning rate	$1e{-}5$
Effective warmup epochs	120
Effective epochs	800
Effective batch size	256
Gradient clipping	—
Precision	fp32

Table 6.9: **SSVP-SLT** pretraining settings

Parameter	Value
CLIP Text tower	T5-v1.1-base
CLIP Feature pooling	mean
CLIP Feature projection	2-layer MLP
Decoder blocks	8
Decoder heads	8
Mask ratio	0.9
Drop path rate	0.2
Video size (T, C, H, W)	(128, 3, 224, 224)
Sampling Rate	2
Face Blur	✓
Random Crop	✓
Horizontal Flip	✓
RandAug	✓(4, 7)
Repeated sampling	1
Optimizer	AdamW
Optimizer momentum	$\beta_1 = 0.9$, $\beta_2 = 0.95$
Weight decay	0.05
GradNorm α	1.0
Peak learning rate	M = $1e{-}4$, GN = $1e{-}2$
Learning rate schedule	Cosine Decay
Minimum learning rate	M = $1e{-}6$, GN = $1e{-}4$
Effective warmup epochs	M = 10, GN = 1
Effective epochs	200
Effective batch size	512
Gradient clipping	1.0
Precision	fp32

Table 6.10: **SSVP-SLT-LSP** pretraining settings. “M” refers to the main optimizer while “GN” refers to the GradNorm optimizer.

Parameter	Value
Model & Tokenizer	T5-v1.1
Dropout probability	0.3
Label smoothing	0.2
Number of beams	5
Video size (T, C, H, W)	(T, 3, 224, 224)
Sampling Rate	2
Face Blur	✓
Random Crop	✗
Horizontal Flip	✗
RandAug	✗
Min sequence length	0
Max sequence length	1024
Max target length	128
Optimizer	AdamW
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.95$
Weight decay	$1e{-}1$
Peak learning rate	$5e{-}4$
Learning rate schedule	Cosine Decay
Minimum learning rate	$1e{-}7$
Warmup epochs	2
Epochs	100
Batch size	256
Early stopping	✓
Gradient clipping	1.0
Precision	fp32

Table 6.11: Finetuning settings for Youtube-ASL.

Parameter	Value
Encoder layers	6
Decoder layers	3
Attention heads	4
Embedding dim	256
FFN embedding dim	1024
Output dim	256
Layerdrop	0.0
Activation function	ReLU
LayerNorm Before	✓
LayerNorm Embedding	✓
Scale embeddings	✓
Decoder share embeddings	✓
Vocab size	7000
Lowercase tokenizer	H2S = ✓, DM = ✗
Truecase outputs	H2S = ✓, DM = ✗
Dropout probability	0.3
Label smoothing	0.2
Number of beams	5
Video size (T, C, H, W)	(T, 3, 224, 224)
Sampling Rate	2
Face Blur	H2S = ✓, DM = ✓/ ✗
Random Crop	✗
Horizontal Flip	✗
RandAug	✗
Min sequence length	0
Max sequence length	1024
Max target length	128
Optimizer	AdamW
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.95$
Weight decay	$1e{-}1$
Peak learning rate	$1e{-}2$
Learning rate schedule	Cosine Decay
Minimum learning rate	$1e{-}4$
Warmup epochs	10
Epochs	200
Batch size	32
Early stopping	✓
Gradient clipping	1.0
Precision	fp16

Table 6.12: Finetuning settings for How2Sign (H2S) & DailyMoth-70h (DM).

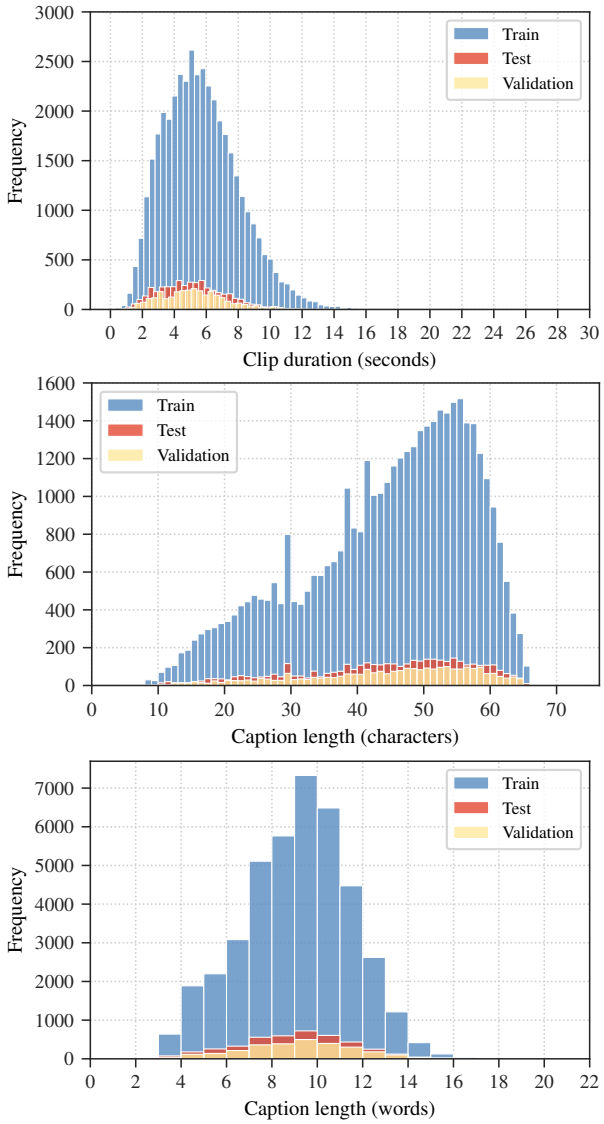
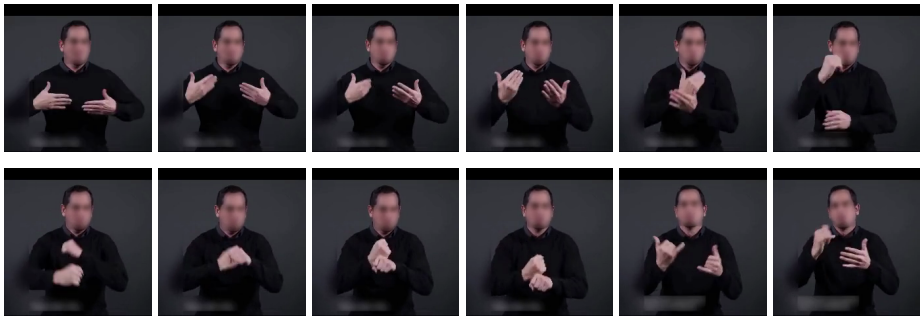


Figure 6.4: DailyMoth-70h dataset split statistics

(a) English translation: "*Hello, welcome to The Daily Moth.*"



(b) English translation: "*Happy New Year*"

Figure 6.5: Example frames sampled from two videos in the blurred version of the DailyMoth-70h training split.

(1)	**Reference**	And that's a great vital point technique for women's self defense.
	Tarrés et al.	It's really a great point for women's self defense.
	Uthus et al.	It's really great for women's self defense.
	Ours	This is a really great point for women's self defense.
(2)	**Reference**	In this clip I'm going to show you how to tape your cables down.
	Tarrés et al.	In this clip I'm going to show you how to improve push ups.
	Uthus et al.	In this clip we're going to show you how to cut a piece of clay.
	Ours	In this clip I'm going to show you how to clip the cable, the cable.
(3)	**Reference**	In this segment we're going to talk about how to load your still for distillation of lavender essential oil.
	Tarrés et al.	Ok, in this clip, we're going to talk about how to fold the ink for the lid of the oil.
	Uthus et al.	In this clip we're going to talk about how to feed a set of baiting lizards for a lava field oil.
	Ours	In this clip we're going to talk about how to feed the trail for draining clean for laborer oil.
(4)	**Reference**	You are dancing, and now you are going to need the veil and you are going to just grab the veil as far as possible.
	Tarrés et al.	So, once you're belly dancing, once you've got to have the strap, you're going to need to grab the thumb, and try to avoid it.
	Uthus et al.	Their hopping and dancing is now, they're going to need their squat and squat and they're going to be able to move independently.
	Ours	So that she's going to get her hips up as far as she can, and now she's going to lift her head up as far as possible.
(5)	**Reference**	But if you have to setup a new campfire, there's two ways to do it in a very low impact; one is with a mound fire, which we should in the campfire segment earlier and the other way to setup a low impact campfire is to have a fire pan, which is just a steel pan like the top of a trash can.
	Tarrés et al.	And other thing I'm going to talk to you is a little bit more space, a space that's what it's going to do, it's kind of a quick, and then I don't want to take a spray skirt off, and then I don't want it to take it to the top of it.
	Uthus et al.	But if you have to set up a new campfire, there are two ways to do a low impact fire, one is a cone fire, which we have to do in the tent earlier, and the other one is to set up a campfire in a fire pan.
	Ours	But if you have to set up a new campfire, this is one way to do it in a low impact. One is a monk fire. One is a campfire. The other one is to set a campfire in a campfire. That's just a post like the top of the post.
(6)	**Reference**	So, this is a very important part of the process.
	Tarrés et al.	It's a very important part of the process.
	Uthus et al.	Alright, let's get started.
	Ours	It's an important part of the process.

Table 6.13: Qualitative translation examples from our best-performing model compared to Tarrés et al. (2023), Uthus et al. (2023), and the reference translations. The examples were picked from the How2Sign test set by Tarrés et al. (2023) and do not necessarily accurately reflect progress on the task. We see that our model is mostly on-topic, but can still struggle with repetitions and the mixing-up of signs.

Chapter 7

Conclusion

This thesis was motivated by the transformative impact of modern natural language processing, particularly pretrained language models, and the concurrent imperative to ensure that these technologies are developed and deployed in ways that are both inclusive and trustworthy. As outlined in the introduction (§ 1), the prevailing paradigm based on tokenization, while powerful, faces fundamental limitations in handling the diversity and multimodality of language, and presents challenges related to trustworthiness desiderata, such as fairness and robustness. In response, we investigated how an alternative framework—specifically, visual language representation—and the broader trend toward multilingualism in NLP could help mitigate these concerns. This concluding chapter synthesizes the findings and contributions of this thesis, discusses them in the context of the rapidly evolving field, acknowledges limitations, and outlines promising directions for future research.

7.1 Discussion

The landscape of AI and NLP has continued its rapid evolution, with advances in model scale, multimodal capabilities, and alignment techniques emerging even since the core research presented in this thesis was conducted. However, the discussed challenges surrounding inclusivity and trustworthiness remain central concerns (Ovalle et al., 2024; Huang et al., 2024; Li et al., 2025). For example, issues such as the digital divide for low-resource languages (Nigatu et al., 2024), the brittleness of models to noise and distribution shifts (Hendrycks et al., 2022), and the difficulty of ensuring several trustworthiness desiderata simultaneously (Cresswell, 2025) persist. In this context, the contributions of this thesis offer relevant insights and potential solutions.

Contributions A core contribution was the exploration of visual language representation learning as an alternative to tokenization-based approaches. By proposing PIXEL (§ 2, 3), which processes rendered text images using vision transformers and masked autoencoding, we demonstrated a way to circumvent the *vocabulary bottleneck* that presents major challenges for language scaling of token-based multilingual models. This pixel-based approach inherently supports any digitally representable script or language, showing strong performance in cross-lingual and cross-script adaptation and handling code-switching without predefined subword units. Furthermore, this visual paradigm proved to be naturally suited for tackling the *digitization lag* affecting many languages and domains. In § 4, we showed that pretraining on scanned historical documents enables effective downstream task performance without relying on potentially error-prone OCR, resulting in enhanced robustness to the visual noise and degradation commonly found in such materials. This suggests a promising path for processing languages with non-standard orthographies, scripts unsupported by Unicode, or primarily available in non-digital, potentially degraded formats. This robustness to orthographic noise was also confirmed in our controlled experiments on digital text with PIXEL (§ 2). Due to its benefits, this framework for visual language representation learning has been met with excitement by the research community and has continued to be an active area of research. To mention only a few of the recent advances in visual language representation learning: Lee et al. (2023) developed a pretraining strategy for visually-situated text, learning to parse screenshots into HTML for state-of-the-art performance across several visual language understanding tasks; Salesky et al. (2023b) trained multilingual pixel-to-text translation models, highlighting their benefits for positive cross-lingual transfer and data-efficient language scaling; Tschannen et al. (2023) trained a contrastive image-text encoder exclusively on pixels, achieving strong out-of-the-box performance on tasks such as multilingual multimodal retrieval; Tai et al. (2024) proposed a pixel-based autoregressive language model, making it possible not just to understand but also generate text with PIXEL models; Gao et al. (2024) improved performance by extending PIXEL's pretraining objective with an autoregressive text decoder; Alonso et al. (2024) trained a pixel-based model achieving strong performance on table-to-text generation.

Our study examining the interactions between multilinguality, task performance, and specific trustworthiness criteria—differential privacy, linguistic fairness, and instance-interpretability–in the context of conventional text encoders (§ 5) complemented our visual approaches, which lend themselves well to

multilinguality by overcoming the vocabulary bottleneck. As the first study that explored three-way and four-way interactions between these particular criteria, we identified specific challenges in satisfying all trustworthiness dimensions simultaneously. We believe that these gained insights will help guide the development of methods that advance the Pareto frontier of trustworthiness.

We also challenged the text-centric state of NLP, highlighting that language goes beyond its written forms, therefore necessitating support for spoken and signed languages for true inclusivity. In this regard, we showed how the visual language representation learning framework naturally extends to sign language (§ 6), given its visual-manual nature. By leveraging self-supervised pretraining on large amounts of unannotated video data, our SSVP-SLT framework achieved state-of-the-art results for ASL-to-English translation, significantly reducing the gap compared to spoken language NMT and demonstrating a scalable approach to overcoming data scarcity in this historically under-served modality. Despite the fast pace of the field, SSVP-SLT continues to be a tough baseline to beat in ASL-to-English translation, having so far been surpassed only by approaches trained on significantly more data (Zhang et al., 2024; Tanzer, 2025). Crucially, our work also integrated a privacy-aware methodology, showing that effective representations can be learned from anonymized videos during pretraining, addressing privacy concerns surrounding biometric data in sign language resources. Lastly, our openly released DailyMoth-70h dataset serves as a valuable evaluation benchmark for future research on sign language processing.

Limitations Despite our promising results, it is important to acknowledge the limitations. Pixel-based methods for written language (§ 2; § 3; § 4; the follow-up research mentioned above), while demonstrating strong capabilities in specific areas (like cross-script generalization and noise robustness), have generally not yet matched the overall benchmark performance of highly optimized state-of-the-art token-based models, especially for high-resource languages where tokenizers are well-tuned. Understanding the reasons for this gap—whether due to optimization challenges, architectural suitability, data scaling requirements, or simply less cumulative research effort compared to the token paradigm—is crucial for future progress. We hypothesize that this performance difference relates to a potential *inclusivity-scaling* trade-off. In particular, methods designed for broader coverage and robustness (like pixel-, byte-, or character-level models) often process less optimally compressed information or longer sequences compared to subword tokenizers that aggressively compress frequent patterns in dominant

languages. It remains an open question how we can best balance the need for flexible applicability and robustness with the need for computational efficiency and state-of-the-art performance. We have also primarily explored visual language representations in the context of natural language understanding, while much of NLP today relies on a model's ability to generate language. It remains an open problem what benefits visual representations can bring for such generative tasks. In general, low-resource NLP is far from solved, and tasks such as sign language translation still have limited practical viability despite our advances. It will take considerable further efforts to lift NLP for low-resource languages up to the current state of high-resource NLP. Similarly, our explorations on trustworthiness criteria (§ 5) have largely been limited to advancing our conceptual understanding of synergies and trade-offs, with no practical solutions harnessing this improved understanding to better navigate these trade-offs. This reflects the broader *implementation gap* in trustworthy AI; although several more recent survey and position papers call (Ferry et al., 2023; Li et al., 2025; Cresswell, 2025)—just like us—for more research at the intersection of trustworthiness desiderata, rather than treating them in isolation, practical advances in this direction have remained scarce. Closing this implementation gap represents a key challenge towards building trustworthy AI. In our sign language work (§ 6), although providing a practical method for enhancing the privacy of signers, we emphasize the need for more sophisticated anonymization strategies beyond facial blurring, and acknowledge the fact that anonymization does not provide any formal privacy guarantees, so it should not be relied on in high-stakes applications. These acknowledged limitations, open questions, and recent developments in the field naturally motivate promising directions for future research.

7.2 Future Work

We highlight several particularly promising future directions, building on the advances of this thesis.

Scaling of pixel-based language models As discussed in the previous section, our foundational work on PIXEL (§ 2) and text rendering strategies for PIXEL models (§ 3) have hinted at potential scaling challenges (and resulting trade-offs between inclusivity and scalability) with pixel-based models, compared to the incumbent token-based ones. As Tay et al. (2023) have found, the best model architecture or inductive bias can also vary across model and compute scales. A

critical next step is, therefore, to systematically investigate the scaling properties of the visual language representation framework. Do models like PIXEL exhibit predictable scaling laws relating performance to model size, dataset size, and compute, similar to unimodal or mixed-modal token-based LLMs (Kaplan et al., 2020; Hoffmann et al., 2022; Aghajanyan et al., 2023)? How do these laws compare? Establishing these relationships is essential for determining the practical limits and potential of this approach and guiding efforts to close the performance gap with token-based models.

Hybrid approaches and multimodal LLM integration Exploring hybrid architectures that combine the strengths of token-based and pixel-based processing is another highly promising avenue. For example, this could involve models that dynamically choose the representation based on the input (e.g., using pixels for noisy text or unfamiliar scripts), something along the lines of which has recently been explored by Lotz et al. (2025)). One might also explore multi-input systems that directly create complementary views of inputs—one challenge here is minimizing the potential extra computational cost and latency of these approaches.

It also seems very promising to integrate pixel-based language modeling objectives into the pretraining or post-training stacks of multimodal LLMs. As a result of increased model scale combined with pretraining and instruction tuning on vast amounts of mixed-modal training data, these multimodal LLMs have recently demonstrated impressive zero-shot capabilities across a wide array of tasks, some related to those discussed in this thesis (e.g., OCR and reasoning over visually-situated text) (Li et al., 2023b; Zhu et al., 2024; Liu et al., 2023; Gemini Team et al., 2024; Chameleon Team, 2025). However, for processing of the textual modality itself, these models still rely on the same conventional tokenizers whose limitations our approaches seek to overcome. Their objective is generally broad multimodal understanding, not necessarily learning language through its visual form to bypass tokenization issues. As such, we see potential for synergies by integrating our visual language representation framework into these multimodal LLMs. Overall, such integration might lead to more robust models capable of grounding language in its visual form when beneficial, perhaps moving closer to modality-agnostic representations (Huh et al., 2024).

Massively multilingual visual language representations Instead of relying primarily on cross-lingual adaptation from English (§ 2; § 3; § 4) or American Sign Language (§ 6) pretraining, future work should explore pretraining visual language models directly on large-scale multilingual corpora. This could improve

performance and finetuning efficiency on a wider range of languages, either through direct exposure or positive transfer from more closely related languages. For written language, it could also improve handling of mixed-script text and code-switching, and further improve robustness to orthographic variation. Extending this further—related to multimodal LLM integration—one could investigate models pretrained jointly on diverse visual language forms, such as rendered text, document scans, sign language videos, and speech spectrograms, fostering unified representations and facilitating transfer across language modalities.

Targeting the low-resource long tail As pointed out before, the success of visual representations for historical documents (§ 4) motivates a concerted effort to apply similar techniques to extremely low-resource languages. This includes languages with limited digital text, inconsistent orthography, unencoded scripts, or where primary resources are handwritten manuscripts or degraded prints. Visual models' ability to bypass OCR and potentially handle severe noise could be transformative for documenting, preserving, and building NLP tools for these languages. Again, this exploration could include integrating our methods within the broader multimodal LLM framework.

Deepening trust The unique nature of visual language representations calls for deeper investigation into their trustworthiness properties—ideally focusing on several dimensions at once. This includes conducting fine-grained fairness audits (e.g., how do models handle visual variations corresponding to different dialects or sociolects?), assessing robustness against visually-grounded adversarial attacks, and applying nascent mechanistic interpretability techniques (Olah et al., 2020; Bricken et al., 2023) to understand how these models learn linguistic structures from pixels (expanding on "pixology" work (Tatariya et al., 2024a)), potentially revealing internal mechanisms different from token-based models.

7.3 Closing Remarks

The methods developed and analyzed in this thesis represent steps towards bridging the gap between the current state of NLP and the pressing need for technologies that embrace the world's diverse linguistic landscape safely and effectively. However, this work is not the end of the story. Most importantly, I hope that this research encourages a curiosity and willingness within the community to continually challenge established paradigms and explore alternative pathways.

Bibliography

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, Vienna, Austria. ACM.

Judit Ács. 2019. Exploring BERT's Vocabulary. *Blog Post*.

Ife Adebara and Muhammad Abdul-Mageed. 2022. Towards afrocentric NLP for African languages: Where we are and where we can go. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3814–3841, Dublin, Ireland. Association for Computational Linguistics.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, and 42 others. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Milind Agarwal and Antonios Anastasopoulos. 2024. A concise survey of OCR for low-resource languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 88–102, Mexico City, Mexico. Association for Computational Linguistics.

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. 2023. Scaling laws for generative mixed-modal language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Philippe Aghion, Benjamin F. Jones, and Charles I. Jones. 2018. *Artificial In-*

telligence and Economic Growth, pages 237–282. Volume 9 of Agrawal et al. (2019).

Ajay Agrawal, Joshua Gans, and Avi Goldfarb. 2019. *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Alham Fikri Aji, Jessica Zosa Forde, Alyssa Marie Loo, Lintang Sutawika, Skyler Wang, Genta Indra Winata, Zheng-Xin Yong, Ruochen Zhang, A. Seza Doğruöz, Yin Lin Tan, and Jan Christian Blaise Cruz. 2023. Current status of NLP in south East Asia with insights from multilingualism and language diversity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: Tutorial Abstract*, pages 8–13, Nusa Dua, Bali. Association for Computational Linguistics.

Alham Fikri Aji, Genta Indra Winata, Fajri Koto, Samuel Cahyawijaya, Ade Romadhony, Rahmad Mahendra, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Timothy Baldwin, Jey Han Lau, and Sebastian Ruder. 2022. One country, 700+ languages: NLP challenges for underrepresented languages and dialects in Indonesia. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7226–7249, Dublin, Ireland. Association for Computational Linguistics.

Ahmad Al-Sallab, Ramy Baly, Hazem Hajj, Khaled Bashir Shaban, Wassim El-Hajj, and Gilbert Badaro. 2017. Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 16(4).

Khetam Al Sharou, Zhenhao Li, and Lucia Specia. 2021. Towards a better understanding of noise in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62, Held Online. INCOMA Ltd.

Samuel Albanie, Gül Varol, Liliane Momeni, Hannah Bull, Triantafyllos Afouras, Himel Chowdhury, Neil Fox, Bencie Woll, Rob Cooper, Andrew McParland,

and Andrew Zisserman. 2021. Bbc-oxford british sign language dataset. *arXiv preprint*.

Iñigo Alonso, Eneko Agirre, and Mirella Lapata. 2024. PixT3: Pixel-based table-to-text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6721–6736, Bangkok, Thailand. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2021. How suitable are subword segmentation strategies for translating non-concatenative morphology? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 689–705, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 993–1003.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020a. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020b. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online. Association for Computational Linguistics.

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

Nishanth Arun, Nathan Gaw, Praveer Singh, Ken Chang, Mehak Aggarwal, Bryan Chen, Katharina Hoebel, Sharut Gupta, Jay Patel, Mishka Gidwani, Julius Adebayo, Matthew D. Li, and Jayashree Kalpathy-Cramer. 2020. Assessing the (un)trustworthiness of saliency maps for localizing abnormalities in medical imaging. *medRxiv*.

Masayuki Asahara, Hiroshi Kanayama, Takaaki Tanaka, Yusuke Miyao, Sumire Uematsu, Shinsuke Mori, Yuji Matsumoto, Mai Omura, and Yugo Murawaki. 2018. Universal Dependencies version 2 for Japanese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint*.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. 2019. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 15453–15462, Vancouver, BC, Canada. Curran Associates, Inc.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint*.

Charlotte Baker-Shenk. 1985. The facial behavior of deaf signers: Evidence of a complex language. *American Annals of the Deaf*, 130(4).

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China. Association for Computational Linguistics.

Dana H. Ballard. 1987. Modular learning in neural networks. In *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*, AAAI'87, page 279–284. AAAI Press.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2022. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*.

Blouin Baptiste, Benoit Favre, Jeremy Auguste, and Christian Henriot. 2021. Transferring modern named entity recognition to the historical domain: How to take the step? In *Workshop on Natural Language Processing for Digital Humanities (NLP4DH)*.

Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, Kathleen Fisher, Tatsunori Hashimoto, Dan Hendrycks, Somesh Jha, Daniel Kang, Florian Kerschbaum, Eric Mitchell, John Mitchell, Zulfikar Ramzan, and 4 others. 2023. Identifying and mitigating the security risks of generative ai. *Foundations and Trends in Privacy and Security*, 6(1):1–52.

Samyadeep Basu, Xuchen You, and Soheil Feizi. 2020. On second-order group influence functions for black-box predictions. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 715–724, Online. PMLR.

Emily M. Bender. 2011. On achieving and evaluating language-independence in nlp. *Linguistic Issues in Language Technology*, 6.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic interpretability for ai safety - a review. *Transactions on Machine Learning Research*.

Richard A. Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2018. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50:3 – 44.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Gregory Anthony, Shivanshu Purohit, and Edward Raff. 2023. Emergent and predictable memorization in large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Reuben Binns. 2020. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 514–524, New York, NY, USA. Association for Computing Machinery.

Steven Bird. 2020. Decolonising speech and language technology. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chris M. Bishop. 1995. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116.

Maxwell Troy Bland, Anushya Iyer, and Kirill Levchenko. 2022. Story beyond the eye: Glyph positions break PDF text redaction. *arXiv preprint*.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2022. Analyzing the mono-and cross-lingual pretraining dynamics of multilingual language models. *arXiv preprint*.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Marcel Bollmann. 2019. A large-scale comparison of historical text normalization systems. In *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 3885–3898, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcel Bollmann, Anders Søgaard, and Joachim Bingel. 2018. Multi-task learning for historical text normalization: Size matters. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 19–24.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NeurIPS '16, page 4356–4364, Red Hook, NY, USA. Curran Associates Inc.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, and 95 others. 2021. On the opportunities and risks of foundation models. *arXiv preprint*.

Rishi Bommasani, Kevin Klyman, Shayne Longpre, Betty Xiong, Sayash Kapoor, Nestor Maslej, Arvind Narayanan, and Percy Liang. 2024. Foundation model transparency reports. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 7(1):181–195.

Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, Mark Ibrahim, Melissa Hall, Yunyang Xiong, Jonathan Lebensold, Candace Ross, Srihari Jayakumar, Chuan Guo, Diane Bouchacourt, Haider Al-Tahan, and 22 others. 2024. An introduction to vision-language modeling. *arXiv preprint*.

Nadav Borenstein, Natalia da Silva Perez, and Isabelle Augenstein. 2023a. Multilingual event extraction from historical newspaper adverts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada. Association for Computational Linguistics.

Nadav Borenstein, Phillip Rust, Desmond Elliott, and Isabelle Augenstein. 2023b. PHD: Pixel-based language modeling of historical documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 87–107, Singapore. Association for Computational Linguistics.

Nadav Borenstein, Karolina Stańczak, Thea Rolskov, Natacha Klein Käfer, Natalia da Silva Perez, and Isabelle Augenstein. 2023c. Measuring intersectional biases in historical documents. *Association for Computational Linguistics*.

Diane Bouchacourt and Marco Baroni. 2018. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 981–985, Brussels, Belgium. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Stephanie Brandl, Emanuele Bugliarello, and Ilias Chalkidis. 2024. On the interplay between fairness and explainability. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 94–108, Mexico City, Mexico. Association for Computational Linguistics.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Samuel Broscheit. 2018. Learning distributional token representations from visual features. In *Proceedings of the Third Workshop on Representation Learning for NLP*, pages 187–194, Melbourne, Australia. Association for Computational Linguistics.

Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. 2021. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory*

of Computing, STOC 2021, page 123–132, New York, NY, USA. Association for Computing Machinery.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian Hadfield, Heidy Khlaaf, Jingying Yang, Helen Toner, Ruth Fong, Tegan Maharaj, Pang Wei Koh, Sara Hooker, Jade Leung, Andrew Trask, Emma Bluemke, Jonathan Lebensold, Cullen O'Keefe, Mark Koren, and 40 others. 2020. Toward trustworthy ai development: Mechanisms for supporting verifiable claims. *arXiv preprint*.

Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vulić. 2022. IGLUE: A benchmark for transfer learning across modalities, tasks, and languages. In *Proceedings of the 39th International Conference on Machine Learning*, Balitmore, MA. PMLR.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*.

Nitay Calderon, Naveh Porat, Eyal Ben-David, Alexander Chapanin, Zorik Gekhman, Nadav Oved, Vitaly Shalumov, and Roi Reichart. 2024. Measuring the robustness of NLP models to domain shifts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 126–154, Miami, Florida, USA. Association for Computational Linguistics.

Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

Clément L. Canonne, Gautam Kamath, and Thomas Steinke. 2020. The discrete gaussian for differential privacy. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 15676–15688, Online. Curran Associates, Inc.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint*.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733.

Lewis Carroll. 1865a. *Alice's Adventures in Wonderland*. Macmillan.

Lewis Carroll. 1865b. *Alice's Adventures Under Ground*. British Library.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona, Spain (Online). Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Chadwyck. 1998. Early english books online : Eebo.

Chameleon Team. 2025. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint*.

H. Chang and R. Shokri. 2021. On the privacy risks of algorithmic fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 292–303, Los Alamitos, CA, USA. IEEE Computer Society.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Ben Bergen. 2024. When is multilinguality a curse? language modeling for 250 high- and low-resource languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural*

Language Processing, pages 4074–4096, Miami, Florida, USA. Association for Computational Linguistics.

Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. 2019. Input similarity from the neural network perspective. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 5343–5352, Vancouver, BC, Canada. Curran Associates, Inc.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

Hila Chefer, Shir Gur, and Lior Wolf. 2021. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. 2025. Deconstructing denoising diffusion models for self-supervised learning. In *The Thirteenth International Conference on Learning Representations*.

Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022a. A simple multi-modality transfer learning baseline for sign language translation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5110–5120.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. 2022b. Two-stream network for sign language recognition and translation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NeurIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine*

Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 794–803. PMLR.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Rochelle Choenni, Sara Rajaee, Christof Monz, and Ekaterina Shutova. 2024a. On the evaluation practices in multilingual nlp: Can machine translation offer an alternative to human translations? *arXiv preprint*.

Rochelle Choenni and Ekaterina Shutova. 2020. What does it mean to be language-agnostic? probing multilingual sentence encoders for typological properties. *arXiv preprint*.

Rochelle Choenni, Ekaterina Shutova, and Dan Garrette. 2024b. Examining modularity in multilingual LMs via language-specialized subnetworks. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 287–301, Mexico City, Mexico. Association for Computational Linguistics.

Monojit Choudhury and Amit Deshpande. 2021. How linguistically fair are multilingual pre-trained language models? In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 12710–12718, Online. AAAI Press.

Grzegorz Chrupała. 2019. Symbolic inductive bias for visually grounded learning of spoken language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6462, Florence, Italy. Association for Computational Linguistics.

Grzegorz Chrupała and Afra Alishahi. 2019. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962, Florence, Italy. Association for Computational Linguistics.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*.

Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. 2018. Building Universal Dependency treebanks in Korean. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online. Association for Computational Linguistics.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 575–593, Toronto, Canada. Association for Computational Linguistics.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation . In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10020–10030, Los Alamitos, CA, USA. IEEE Computer Society.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Jesse C. Cresswell. 2025. Trustworthy ai must account for intersectionality. *ICLR 2025 Workshop on Bidirectional Human-AI Alignment*.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. Randaugment: practical automated data augmentation with a reduced search space. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. 2019. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, UMAP'19 Adjunct, page 309–315, New York, NY, USA. Association for Computing Machinery.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. 2017. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100. Language in Vision.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. 2023. End-to-end document recognition and understanding with dessurt. In *Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, page 280–296, Berlin, Heidelberg. Springer-Verlag.

Mathieu De Coster, Karel D'Oosterlinck, Marija Pizurica, Paloma Rabaey, Severine Verlinden, Mieke Van Herreweghe, and Joni Dambre. 2021. Frozen pretrained transformers for neural sign language translation. In *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*, pages 88–97, Virtual. Association for Machine Translation in the Americas.

Anderson Santana de Oliveira, Caelin Kaplan, Khawla Mallat, and Tanmay Chakraborty. 2024. An empirical analysis of fairness notions under differential privacy. *The Fourth AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI-23)*.

Chameera De Silva and Thilina Halloluwa. 2025. Augmenting human potential: The role of llms in shaping the future of hci. *Interactions*, 32(2):42–45.

Francesco De Toni, Christopher Akiki, Javier De La Rosa, Clémentine Fourrier, Enrique Manjavacas, Stefan Schweter, and Daniel Van Strien. 2022. Entities, dates, and languages: Zero-shot on historical texts with t0. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 75–83, virtual+Dublin. Association for Computational Linguistics.

Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. 2020. What's so special about BERT's layers? a closer look at the NLP pipeline in monolingual and multilingual models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350, Online. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang,

Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint*.

Thomas Delteil, Edouard Belval, Lei Chen, Luis Goncalves, and Vijay Mahadevan. 2022. MATrIX – Modality-Aware Transformer for Information eXtraction. *arXiv preprint*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Sunipa Dev, Vinodkumar Prabhakaran, David Ifeoluwa Adelani, Dirk Hovy, and Luciana Benotti, editors. 2023. *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*. Association for Computational Linguistics, Dubrovnik, Croatia.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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.

Harnoor Dhingra, Preetiha Jayashanker, Sayali Moghe, and Emma Strubell. 2023. Queer people are people first: Deconstructing sexual identity stereotypes in large language models. *arXiv preprint*.

Jörn Diedrichsen and Nikolaus Kriegeskorte. 2017. Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLOS Computational Biology*, 13(4):1–33.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *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 4537–4546, Hong Kong, China. Association for Computational Linguistics.

Yue Ding, Karolis Martinkus, Damian Pascual, Simon Clematide, and Roger Wattenhofer. 2022. On isotropy calibration of transformer models. In *Proceedings*

of the Third Workshop on Insights from Negative Results in NLP, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Senka Drobac, Pekka Kauppinen, and Krister Lindén. 2017. Ocr and post-correction of historical finnish texts. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 70–76.

Tianqi Du, Yifei Wang, and Yisen Wang. 2024. On the role of discrete tokenization in visual representation learning. In *The Twelfth International Conference on Learning Representations*.

Amanda Cardoso Duarte, Samuel Albanie, Xavier Giró-i-Nieto, and Gül Varol. 2022. Sign language video retrieval with free-form textual queries. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14074–14084.

Amanda Cardoso Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giró-i-Nieto. 2021. How2sign: A large-scale multimodal dataset for continuous american sign language. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2734–2743.

Jeppe Klok Due, Marianne Giørtz Pedersen, Sussie Antonsen, Joen Rommedahl, Esben Agerbo, Preben Bo Mortensen, Henrik Toft Sørensen, Jonas Færch Lotz, Laura Cabello Piqueras, Constanza Fierro, Antonia Karamolegkou, Christian Igel, Phillip Rust, Anders Søgaard, and Carsten Bøcker Pedersen. 2024. Towards more comprehensive nationwide familial aggregation studies in denmark: The danish civil registration system versus the lite danish multi-generation register. *Scandinavian Journal of Public Health*, 52(4):528–538. PMID: 37036022.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Cynthia Dwork. 2006. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP), Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12, Venice, Italy. Springer.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4555–4567, Online. Association for Computational Linguistics.

Shimon Edelman. 1998. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21(4):449–467.

Steffen Eger and Yannik Benz. 2020. From hero to zéroe: A benchmark of low-level adversarial attacks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 786–803, Suzhou, China. Association for Computational Linguistics.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *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 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *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 55–65, Hong Kong, China. Association for Computational Linguistics.

European Parliament and Council of the European Union. 2016. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (general data protection regulation). Accessed: 2025-04-28.

Agnieszka Faleńska, Christine Basta, Marta Costa-jussà, Seraphina Goldfarb-Tarrant, and Debora Nozza, editors. 2024. *Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*. Association for Computational Linguistics, Bangkok, Thailand.

Minghong Fan. 2024. Llms in banking: Applications, challenges, and approaches. In *Proceedings of the International Conference on Digital Economy, Blockchain and Artificial Intelligence*, DEBAI '24, page 314–321, New York, NY, USA. Association for Computing Machinery.

Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. 2023. Fate-llm: A industrial grade federated learning framework for large language models. *Symposium on Advances and Open Problems in Large Language Models (LLM@IJCAI'23)*.

Zahra Fatemi, Chen Xing, Wenhao Liu, and Caimming Xiong. 2023. Improving gender fairness of pre-trained language models without catastrophic forgetting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1249–1262, Toronto, Canada. Association for Computational Linguistics.

Christoph Feichtenhofer, Haoqi Fan, and Yanghao Li Kaiming He. 2022. Masked autoencoders as spatiotemporal learners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NeurIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Julien Ferry, Ulrich Aïvodji, Sébastien Gambs, Marie-José Huguet, and Mohamed Siala. 2023. Sok: Taming the triangle – on the interplays between fairness, interpretability and privacy in machine learning. *arXiv preprint*.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2021. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*.

W. Nelson Francis and Henry Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Andrea Frome, German Cheung, Ahmad Abdulkader, Marco Zennaro, Bo Wu, Alessandro Bissacco, Hartwig Adam, Hartmut Neven, and Luc Vincent. 2009. Large-scale privacy protection in google street view. In *ICCV*.

Alejandro Fuster Baggetto and Victor Fresno. 2022. Is anisotropy really the cause of BERT embeddings not being semantic? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4271–4281, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Esther Gan, Yiran Zhao, Liying Cheng, Mao Yancan, Anirudh Goyal, Kenji Kawaguchi, Min-Yen Kan, and Michael Shieh. 2024. Reasoning robustness of LLMs to adversarial typographical errors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10449–10459, Miami, Florida, USA. Association for Computational Linguistics.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, and 17 others. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint*.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019a. Representation degeneration problem in training natural language generation models. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA. OpenReview.net.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019b. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint*.

Tianyu Gao, Zirui Wang, Adithya Bhaskar, and Danqi Chen. 2024. Improving language understanding from screenshots. *arXiv preprint*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, Atlanta, Georgia. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM*, 64(12):86–92.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1331 others. 2024. Gemini: A family of highly capable multimodal models. *arXiv preprint*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill,

Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *arXiv preprint*.

Quan Geng, Wei Ding, Ruiqi Guo, and Sanjiv Kumar. 2020. Tight analysis of privacy and utility tradeoff in approximate differential privacy. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pages 89–99, Online. PMLR.

Anne Gerritsen. 2012. Scales of a local: the place of locality in a globalizing world. *A Companion to World History*, pages 213–226.

Goran Glavaš and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online. Association for Computational Linguistics.

Trystan S. Goetze. 2022. Mind the gap: Autonomous systems, the responsibility gap, and moral entanglement. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 390–400, New York, NY, USA. Association for Computing Machinery.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *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 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: frequency-agnostic word representation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NeurIPS'18, page 1341–1352, Red Hook, NY, USA. Curran Associates Inc.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Comput. Surv.*, 55(14s).

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *arXiv preprint*.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. 2024. Alignment faking in large language models. *arXiv preprint*.

Michiel van Groesen. 2015. Digital gatekeeper of the past: Delpher and the emergence of the press in the dutch golden age. *Tijdschrift voor Tijdschriftstudies*, 38:9–19.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. Studying large language model generalization with influence functions.

Ivan Habernal. 2021. When differential privacy meets NLP: The devil is in the detail. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, page 1735–1742, USA. IEEE Computer Society.

Jan Hajič, Otakar Smrž, Petr Zemánek, Petr Pajas, Jan Šnaidauf, Emanuel Beška, Jakub Kracmar, and Kamila Hassanová. 2009. Prague arabic dependency treebank 1.0.

Xiaochuang Han and Yulia Tsvetkov. 2021. Influence tuning: Demoting spurious correlations via instance attribution and instance-driven updates. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4398–4409, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Victor Hansen, Atula Neerkaje, Ramit Sawhney, Lucie Flek, and Anders Søgaard. 2024. The impact of differential privacy on group disparity mitigation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3952–3965, Mexico City, Mexico. Association for Computational Linguistics.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021c. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2208–2222, Online. Association for Computational Linguistics.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. 2022. Unsolved problems in ml safety. *arXiv preprint*.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarello, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, Phillip Rust, and Anders Søgaard. 2022a. Challenges and strategies in cross-cultural NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6997–7013, Dublin, Ireland. Association for Computational Linguistics.

Daniel Hershcovich, Nicolas Webersinke, Mathias Kraus, Julia Bingler, and Markus Leippold. 2022b. Towards climate awareness in NLP research. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2480–2494, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mark J Hill and Simon Hengchen. 2019. Quantifying the impact of dirty ocr on historical text analysis: Eighteenth century collections online as a case study. *Digital Scholarship in the Humanities*, 34(4):825–843.

Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science*, 349(6245):261–266.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NeurIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Maxwell Horton, Sachin Mehta, Ali Farhadi, and Mohammad Rastegari. 2023. Bytes are all you need: Transformers operating directly on file bytes. *arXiv preprint*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. 2022. Masked autoencoders that listen. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NeurIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, and 51 others. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark. Association for Computational Linguistics.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20617–20642. PMLR.

Oana Ignat, Jean Maillard, Vishrav Chaudhary, and Francisco Guzmán. 2022. OCR improves machine translation for low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1164–1174, Dublin, Ireland. Association for Computational Linguistics.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint*.

Amy Isard. 2020. Approaches to the anonymisation of sign language corpora. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pages 95–100, Marseille, France. European Language Resources Association (ELRA).

Maor Ivgi, Yair Carmon, and Jonathan Berant. 2022. Scaling laws under the microscope: Predicting transformer performance from small scale experiments. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7354–7371, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Andrew Brock, Evan Shelhamer, Olivier J.

H'enaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. 2021. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint*.

Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi Malvajerdi, and Jonathan Ullman. 2019. Differentially private fair learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 3000–3008, Long Beach, CA, USA. PMLR.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. PromptBERT: Improving BERT sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aravind K. Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020a. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020b. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Karthikeyan K and Anders Søgaard. 2021. Revisiting methods for finding influential examples. *arXiv preprint*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, and 40 others. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint*.

Antonia Karamolegkou, Oliver Eberle, Phillip Rust, Carina Kauf, and Anders Søgaard. 2025a. Trick or Neat: Adversarial ambiguity and language model evaluation. *under review*.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. 2023. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412, Singapore. Association for Computational Linguistics.

Antonia Karamolegkou, Malvina Nikandrou, Georgios Pantazopoulos, Danae Sanchez Villegas, Phillip Rust, Ruchira Dhar, Daniel Hershcovich, and Anders Søgaard. 2025b. Evaluating multimodal language models as visual assistants for visually impaired users. *arXiv preprint (under review)*.

Antonia Karamolegkou, Phillip Rust, Ruixiang Cui, Yong Cao, Anders Søgaard, and Daniel Hershcovich. 2024. Vision-language models under cultural and inclusive considerations. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, pages 53–66, Bangkok, Thailand. ACL.

Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The kinetics human action video dataset. *arXiv preprint*.

Yannik Keller, Jan Mackensen, and Steffen Eger. 2021. BERT-defense: A probabilistic model based on BERT to combat cognitively inspired orthographic adversarial attacks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1616–1629, Online. Association for Computational Linguistics.

Omri Keren, Tal Avinari, Reut Tsarfaty, and Omer Levy. 2022. Breaking character: Are subwords good enough for mrls after all? *arXiv preprint*.

Gavin Kerrigan, Dylan Slack, and Jens Tuyls. 2020. Differentially private language models benefit from public pre-training. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 39–45, Online. Association for Computational Linguistics.

Muhammad Khalifa, Muhammad Abdul-Mageed, and Khaled Shaalan. 2021. Self-training pre-trained language models for zero- and few-shot multi-dialectal Arabic sequence labeling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 769–782, Online. Association for Computational Linguistics.

Simran Khanuja, Sebastian Ruder, and Partha Talukdar. 2023. Evaluating the diversity, equity, and inclusion of NLP technology: A case study for Indian languages. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1763–1777, Dubrovnik, Croatia. Association for Computational Linguistics.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, page 498–517, Berlin, Heidelberg. Springer-Verlag.

Minbeom Kim, Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. 2023. Critic-guided decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4598–4612, Toronto, Canada. Association for Computational Linguistics.

Sohyung Kim, Arianna Bisazza, and Fatih Turkmen. 2021. Using confidential data for domain adaptation of neural machine translation. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*, pages 46–52, Online. Association for Computational Linguistics.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, D. Erhan, and Been Kim. 2019. The (un)reliability of saliency methods. In *Explainable AI*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Michael Bean, Katerina Margatina, Rafael Mosquera, Juan Manuel Ciro, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. 2024. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Stav Klein and Reut Tsarfaty. 2020. Getting the ##life out of living: How adequate are word-pieces for modelling complex morphology? In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 204–209, Online. Association for Computational Linguistics.

Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. 2021. Crypten: secure multi-party computation meets machine learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NeurIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. 2019. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 5255–5265, Vancouver, BC, Canada. Curran Associates, Inc.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894, Sydney, NSW, Australia. PMLR.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. 2020. Big transfer (bit): General visual

representation learning. In *Computer Vision – ECCV 2020*, pages 491–507, Cham. Springer International Publishing.

Zhifeng Kong and Kamalika Chaudhuri. 2021. Understanding instance-based interpretability of variational auto-encoders. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS)*, Online. Curran Associates, Inc.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529, Long Beach, CA, USA. PMLR.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, and 33 others. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Artur Kulmizev, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gertjan van Noord, Barbara Plank, and Martijn Wieling. 2017. The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, Copenhagen, Denmark. Association for Computational Linguistics.

Deepak Kumar, Oleg Lesota, George Zerveas, Daniel Cohen, Carsten Eickhoff, Markus Schedl, and Navid Rekabsaz. 2023. Parameter-efficient modularised bias mitigation via AdapterFusion. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2738–2751, Dubrovnik, Croatia. Association for Computational Linguistics.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip H. S. Torr, Fahad Shahbaz Khan, and Salman Khan. 2025. Llm post-training: A deep dive into reasoning large language models.

Viet Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. 2021. Event extraction from historical texts: A new dataset for black rebellions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2390–2400.

Julia Laite. 2020. The emmet's inch: Small history in a digital age. *Journal of Social History*, 53(4):963–989.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Joon-Woo Lee, Hyungchul Kang, Yongwoo Lee, Woosuk Choi, Jieun Eom, Maxim Deryabin, Eunsang Lee, Junghyun Lee, Donghoon Yoo, Young-Sik Kim, and Jong-Seon No. 2022. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Sooyeon Lee, Abraham Glasser, Becca Dingman, Zhaoyang Xia, Dimitris Metaxas, Carol Neidle, and Matt Huenerfauth. 2021. American sign language video anonymization to support online participation of deaf and hard of hearing users. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '21, New York, NY, USA. Association for Computing Machinery.

Michael Lepori and R. Thomas McCoy. 2020. Picking BERT's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2023a. Trustworthy ai: From principles to practices. *ACM Comput. Surv.*, 55(9).

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022a. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 3530–3539, New York, NY, USA. Association for Computing Machinery.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023c. Trocr: transformer-based optical character recognition with pre-trained models. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Ninghui Li, Wahbeh Qardaji, and Dong Su. 2012. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, page 32–33, New York, NY, USA. Association for Computing Machinery.

Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. 2021. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5652–5660.

Qiongxiu Li, Xiaoyu Luo, Yiyi Chen, and Johannes Bjerva. 2025. Trustworthy machine learning via memorization and the granular long-tail: A survey on interactions, tradeoffs, and beyond. *arXiv preprint*.

Tiancheng Li and Ninghui Li. 2009. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD International Conference*

on Knowledge Discovery and Data Mining, KDD '09, page 517–526, New York, NY, USA. Association for Computing Machinery.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2022b. Large language models can be strong differentially private learners. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. 2023d. Scaling language-image pre-training via masking. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23390–23400.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023e. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, ICAIF '23, page 374–382, New York, NY, USA. Association for Computing Machinery.

Yueqi Li and Sanjay Goel. 2025. Artificial intelligence auditability and auditor readiness for auditing artificial intelligence systems. *International Journal of Accounting Information Systems*, 56:100739.

Feng Liang, Yangguang Li, and Diana Marculescu. 2022. Supmae: Supervised masked autoencoders are efficient vision learners. *arXiv preprint*.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1.0: Korean QA dataset for machine reading comprehension. *arXiv preprint*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kezhou Lin, Xiaohan Wang, Linchao Zhu, Ke Sun, Bang Zhang, and Yi Yang. 2023. Gloss-free end-to-end sign language translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12904–12916, Toronto, Canada. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramón Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. 2015a. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015b. Character-based neural machine translation. *arXiv preprint*.

Zachary C. Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021a. When machine learning meets privacy: A survey and outlook. *ACM Comput. Surv.*, 54(2).

Bo Liu, Li-Ming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2024. How good are LLMs at out-of-distribution detection? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8211–8222, Torino, Italia. ELRA and ICCL.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Jingcheng Liu and Kunal Talwar. 2019. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, page 298–309, New York, NY, USA. Association for Computing Machinery.

Wenyan Liu, Xiangfeng Wang, Xingjian Lu, Junhong Cheng, Bo Jin, Xiaoling Wang, and Hongyuan Zha. 2021b. Fair differential privacy can mitigate the disparate impact on model accuracy. *Submitted to the 9th International Conference on Learning Representations (ICLR)*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint*.

Zihan Liu, Genta Indra Winata, Andrea Madotto, and Pascale Fung. 2021c. Preserving cross-linguality of pre-trained models via continual learning. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 64–71, Online. Association for Computational Linguistics.

Shayne Longpre, Kevin Klyman, Ruth E. Appel, Sayash Kapoor, Rishi Bommasani, Michelle Sahar, Sean McGregor, Avijit Ghosh, Borhane Blili-Hamelin, Nathan Butters, Alondra Nelson, Amit Elazari, Andrew Sellars, Casey John Ellis, Dane Sherrets, Dawn Song, Harley Geiger, Ilona Cohen, Lauren McIlvenny, and 15 others. 2025. In-house evaluation is not enough: Towards robust third-party flaw disclosure for general-purpose ai. *arXiv preprint*.

Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, Xinyi (Alexis) Wu, Enrico Shippole, Kurt Bollacker, Tongshuang Wu, Luis Villa, Sandy Pentland, and Sara Hooker. 2024. A large-scale audit of dataset licensing and attribution in ai. *Nature Machine Intelligence*, 6(8):975–987.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA. OpenReview.net.

Jonas Lotz, Elizabeth Salesky, Phillip Rust, and Desmond Elliott. 2023. Text rendering strategies for pixel language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10155–10172, Singapore. Association for Computational Linguistics.

Jonas F. Lotz, Hendra Setiawan, Stephan Peitz, and Yova Kementchedjhieva. 2025. Overcoming vocabulary constraints with pixel-level fallback. *arXiv preprint*.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Sophie Lythreatis, Sanjay Kumar Singh, and Abdul-Nasser El-Kassar. 2022. The digital divide: A review and future research agenda. *Technological Forecasting and Social Change*, 175:121359.

Lijun Lyu, Maria Koutraki, Martin Krickl, and Besnik Fetahu. 2021. Neural ocr post-hoc correction of historical corpora. *Transactions of the Association for Computational Linguistics*, 9:479–493.

Lingjuan Lyu, Xuanli He, and Yitong Li. 2020. Differentially private representation for NLP: Formal guarantee and an empirical study on privacy and fairness. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2355–2365, Online. Association for Computational Linguistics.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in NLP: A survey. *Computational Linguistics*, 50(2):657–723.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware pre-trained language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 39–50, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Manuel Mager, Abteen Ebrahimi, Shruti Rijhwani, Arturo Oncevay, Luis Chiruzzo, Robert Pugh, and Katharina von der Wense, editors. 2024. *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*. Association for Computational Linguistics, Mexico City, Mexico.

Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. 2018. Challenges of language technologies for the indigenous languages of the Americas. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 55–69, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Christos A. Makridis, Joshua Mueller, Theo Tiffany, Andrew A. Borkowski, John Zachary, and Gil Alterovitz. 2024. From theory to practice: Harmonizing taxonomies of trustworthy ai. *Health Policy OPEN*, 7:100128.

Enrique Manjavacas and Lauren Fonteyn. 2022. Adapting vs. Pre-training Language Models for Historical Languages. *Journal of Data Mining & Digital Humanities*, NLP4DH.

Elman Mansimov, Mitchell Stern, Mia Chen, Orhan Firat, Jakob Uszkoreit, and Puneet Jain. 2020. Towards end-to-end in-image neural machine translation. In *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*, pages 70–74, Online. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Antonis Maronikolakis, Philipp Dufter, and Hinrich Schütze. 2021. Wine is not v i n. on the compatibility of tokenizations across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2382–2399, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada. Association for Computational Linguistics.

Cleo Matzken, Steffen Eger, and Ivan Habernal. 2023. Trade-offs between fairness and privacy in language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6948–6969, Toronto, Canada. Association for Computational Linguistics.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada. OpenReview.net.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6).

Xiangbin Meng, Xiangyu Yan, Kuo Zhang, Da Liu, Xiaojuan Cui, Yaodong Yang, Muhan Zhang, Chunxia Cao, Jingjia Wang, Xuliang Wang, Jun Gao, Yuan-Geng-Shuo Wang, Jia ming Ji, Zifeng Qiu, Muzi Li, Cheng Qian, Tianze Guo, Shuangquan Ma, Zeying Wang, and 6 others. 2024. The application of large language models in medicine: A scoping review. *iScience*, 27(5):109713.

Thomas Miconi. 2017. The impossibility of "fairness": A generalized impossibility result for decisions. *arXiv preprint*.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. *arXiv preprint*.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Ilya Mironov. 2017. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, (CSF)*, pages 263–275, Santa Barbara, CA, USA. IEEE Computer Society.

Ilya Mironov, Kunal Talwar, and Li Zhang. 2019. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint*.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229, New York, NY, USA. Association for Computing Machinery.

Ross E Mitchell and Travas A Young. 2022. How Many People Use Sign Language? A National Health Survey-Based Estimate. *The Journal of Deaf Studies and Deaf Education*, 28(1).

Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. 2024. Auditing large language models: a three-layered approach. *AI and Ethics*, 4(4):1085–1115.

Ibraheem Muhammad Moosa, Mahmud Elahi Akhter, and Ashfia Binte Habib. 2023. Does transliteration help multilingual language modeling? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 670–685, Dubrovnik, Croatia. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Amit Moryossef, Ioannis Tsochantaridis, Joe Dinn, Necati Cihan Camgoz, Richard Bowden, Tao Jiang, Annette Rios, Mathias Muller, and Sarah Ebling. 2021. Evaluating the immediate applicability of pose estimation for sign language recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3429–3435.

Janalyn Moss. 2009. Guides: News and newspapers: Historical newspaper collections. *Iowa's University Libraries*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *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.

Mathias Müller, Malihe Alikhani, Eleftherios Avramidis, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Sarah Ebling, Cristina España-Bonet, Anne Göhring, Roman Grundkiewicz, Mert Inan, Zifan Jiang, Oscar Koller, Amit Moryossef, Annette Rios, Dimitar Shterionov, Sandra Sidler-Miserez, Katja Tissi, and Davy Van Landuyt. 2023a. Findings of the second WMT shared task on sign language translation (WMT-SLT23). In *Proceedings of the Eighth Conference on*

Machine Translation, pages 68–94, Singapore. Association for Computational Linguistics.

Mathias Müller, Zifan Jiang, Amit Moryossef, Annette Rios, and Sarah Ebling. 2023b. Considerations for meaningful sign language machine translation based on glosses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 682–693, Toronto, Canada. Association for Computational Linguistics.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.

Rakshit Naidu, Aman Priyanshu, Aadith Kumar, Sasikanth Kotti, Haofan Wang, and Fatemehsadat Mireshghallah. 2021. When differential privacy meets interpretability: A case study. In *CVPR 2021 Workshop for Responsible Computer Vision (RCV)*.

Carol J. Neidle, Judy Kegl, Benjamin Bahan, Dawn MacLaughlin, and Robert G. Lee. 2000. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure*. MIT Press.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, and 28 others. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Jessica Newman. 2023. A taxonomy of trustworthiness for artificial intelligence. *Center for Long-Term Cybersecurity, United States of America*.

Simon P. Newman, Stephen Mullen, Nelson Mundell, and Roslyn Chapman. 2019. Runaway Slaves in Britain: bondage, freedom and race in the eighteenth century. `https://www.runaways.gla.ac.uk`. Accessed: 2022-12-10.

Phuong-Thai Nguyen, Xuan-Luong Vu, Thi-Minh-Huyen Nguyen, Van-Hiep Nguyen, and Hong-Phuong Le. 2009. Building a large syntactically-annotated corpus of Vietnamese. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 182–185, Suntec, Singapore. Association for Computational Linguistics.

Hellina Hailu Nigatu, Atnafu Lambebo Tonja, Benjamin Rosman, Thamar Solorio, and Monojit Choudhury. 2024. The zeno's paradox of 'low-resource' languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17753–17774, Miami, Florida, USA. Association for Computational Linguistics.

NIST. 2023. Artificial intelligence risk management framework (ai rmf 1.0). NIST AI 100-1, National Institute of Standards and Technology.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint*.

Claudio Novelli, Mariarosaria Taddeo, and Luciano Floridi. 2024. Accountability in artificial intelligence: what it is and how it works. *AI & Society*, 39(4):1871–1882.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. KinyaBERT: a morphology-aware Kinyarwanda language model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363, Dublin, Ireland. Association for Computational Linguistics.

OECD. 2024. Recommendation of the council on artificial intelligence. Accessed: 2025-04-28.

V. Ojewale, R. Steed, B. Vecchione, A. Birhane, and I. D. Raji. 2025. Towards ai accountability infrastructure: Gaps and opportunities in ai audit tooling. In

CHI Conference on Human Factors in Computing Systems (CHI '25), pages 1–29, New York, NY, USA. ACM.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt. Accessed: 2025-04-29.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *arXiv preprint*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, and 7 others. 2024. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. Featured Certification.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Anaelia Ovalle, Kai-Wei Chang, Yang Trista Cao, Ninareh Mehrabi, Jieyu Zhao, Aram Galstyan, Jwala Dhamala, Anoop Kumar, and Rahul Gupta, editors. 2024. *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*. Association for Computational Linguistics, Mexico City, Mexico.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srinivasan Iyer. 2024. Byte latent transformer: Patches scale better than tokens. *arXiv preprint*.

Martha Palmer, Owen Rambow, Rajesh Bhatt, Dipti Misra Sharma, Bhuvana Narasimhan, and F. Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *Proceedings of ICON-2009: 7th International Conference on Natural Language Processing*, India. Macmillan Publishers.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Wang. 2023. On the risk of misinformation pollution with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1389–1403, Singapore. Association for Computational Linguistics.

Ashwinee Panda, Xinyu Tang, Christopher A. Choquette-Choo, Milad Nasr, and Prateek Mittal. 2025. Privacy auditing of large language models. In *The Thirteenth International Conference on Learning Representations*.

Marlotte Pannekoek and Giacomo Spigler. 2021. Investigating trade-offs in utility, fairness and differential privacy in neural networks. *arXiv preprint*.

Nicolas Papernot and Thomas Steinke. 2022. Hyperparameter tuning with renyi differential privacy. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *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 8024–8035.

Vaidehi Patil, Partha Talukdar, and Sunita Sarawagi. 2022. Overlap-based vocabulary generation improves cross-lingual transfer among related languages. In

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 219–233, Dublin, Ireland. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36, pages 36963–36990. Curran Associates, Inc.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Ponti. 2023. Modular deep learning. *Transactions on Machine Learning Research*. Survey Certification.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jason Phang, Michael Lampe, Lama Ahmad, Sandhini Agarwal, Cathy Mengying Fang, Auren R. Liu, Valdemar Danry, Eunhae Lee, Samantha W. T. Chan, Pat Pataranutaporn, and Pattie Maes. 2025. Investigating affective use and emotional well-being on chatgpt. *arXiv preprint*.

Jason Phang, Haokun Liu, and Samuel R. Bowman. 2021. Fine-tuned transformers show clusters of similar representations across layers. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 529–538, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Laura Cabello Piqueras*, Constanza Fierro*, Jonas F. Lotz*, Phillip Rust*, Joen Rommedahl, Jeppe Klok Due, Christian Igel, Desmond Elliott, Carsten B. Pedersen, Israfel Salazar, and Anders Søgaard. 2022. Date recognition in historical parish records. In *Frontiers in Handwriting Recognition*, pages 49–64, Cham. Springer International Publishing.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Esther Ploeger, Wessel Poelman, Miryam de Lhoneux, and Johannes Bjerva. 2024. What is "typological diversity" in NLP? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5681–5700, Miami, Florida, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Yada Pruksachatkun, Anil Ramakrishna, Kai-Wei Chang, Satyapriya Krishna, Jwala Dhamala, Tanaya Guha, and Xiang Ren, editors. 2021. *Proceedings of the First Workshop on Trustworthy Natural Language Processing*. Association for Computational Linguistics, Online.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, Online. Curran Associates, Inc.

Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. 2024. Tokenflow: Unified image tokenizer for multimodal understanding and generation. *arXiv preprint*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021a. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021b. Learning transferable visual models from natural language supervision. In *ICML*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Technical Report*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. 2020. Understanding and mitigating the tradeoff between robustness and accuracy. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Md Mushfiqur Rahman, Fardin Ahsan Sakib, Fahim Faisal, and Antonios Anastasopoulos. 2023. To token or not to token: A comparative study of text representations for cross-lingual transfer. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 67–84, Singapore. Association for Computational Linguistics.

Sara Rajaee and Christof Monz. 2024. Analyzing the evaluation of cross-lingual knowledge transfer in multilingual language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2895–2914, St. Julian's, Malta. Association for Computational Linguistics.

Sara Rajaee and Mohammad Taher Pilehvar. 2021. A cluster-based approach for improving isotropy in contextual embedding space. In *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 575–584, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Loganathan Ramasamy and Zdeněk Žabokrtský. 2012. Prague dependency style treebank for Tamil. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1888–1894, Istanbul, Turkey. European Language Resources Association (ELRA).

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Vinit Ravishankar and Anders Søgaard. 2021. The impact of positional encodings on multilingual compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 763–777, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 431–445, Singapore. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. 2021. Data augmentation can improve robustness. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NeurIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Sandy Ritchie, Daan van Esch, Uche Okonkwo, Shikhar Vashishth, and Emily Drummond. 2024. LinguaMeta: Unified metadata for thousands of languages.

In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10530–10538, Torino, Italia. ELRA and ICCL.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Alexander Robertson and Sharon Goldwater. 2018. Evaluating historical text normalization systems: How well do they generalize? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 720–725, New Orleans, Louisiana. Association for Computational Linguistics.

Sebastian Ruder. 2020. Why You Should Do NLP Beyond English. `http://ruder.io/nlp-beyond-english`. Accessed: 2025-04-29.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2022. Square one bias in NLP: Towards a multi-dimensional exploration of the research manifold. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2340–2354, Dublin, Ireland. Association for Computational Linguistics.

William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff. 2022. IsoScore: Measuring the uniformity of embedding space utilization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3325–3339, Dublin, Ireland. Association for Computational Linguistics.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels.

In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda.

Phillip Rust, Bowen Shi, Skyler Wang, Necati Cihan Camgoz, and Jean Maillard. 2024. Towards privacy-aware sign language translation at scale. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8624–8641, Bangkok, Thailand. Association for Computational Linguistics.

Phillip Rust and Anders Søgaard. 2023. Differential privacy, linguistic fairness, and training data influence: Impossibility and possibility theorems for multilingual language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29354–29387. PMLR.

Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer. 2023. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29441–29454. PMLR.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *International Conference on Learning Representations*.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7235–7252, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023a. Multilingual pixel representations for translation and effective cross-lingual transfer. *arXiv preprint*.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023b. Multilingual pixel representations for translation and effective cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore. Association for Computational Linguistics.

Marcelo Sandoval-Castaneda, Yanhong Li, Diane Brentari, Karen Livescu, and Gregory Shakhnarovich. 2023. Self-supervised video transformers for isolated sign language recognition. *arXiv preprint*.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2021. Anonysign: Novel human appearance synthesis for sign language video anonymisation. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–8.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, and 372 others. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, and 49 others. 2023. Seamlessm4t: Massively multilingual & multimodal machine translation. *arXiv preprint*.

Amit Seker and Reut Tsarfaty. 2020. A pointer network architecture for joint morphological segmentation and tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4368–4378, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the*

Association for Computational Linguistics, pages 7881–7892, Online. Association for Computational Linguistics.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. 2022. The multiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359.

Navoda Senavirathne and Vicenç Torra. 2020. On the role of data anonymization in machine learning privacy. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 664–675.

Manuel Senge, Timour Igamberdiev, and Ivan Habernal. 2022. One size does not fit all: Investigating strategies for differentially-private learning across NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7340–7353, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rohin Shah, Alex Irpan, Alexander Matt Turner, Anna Wang, Arthur Conmy, David Lindner, Jonah Brown-Cohen, Lewis Ho, Neel Nanda, Raluca Ada Popa, Rishub Jain, Rory Greig, Samuel Albanie, Scott Emmons, Sebastian Farquhar, Sébastien Krier, Senthooran Rajamanoharan, Sophie Bridgers, Tobi Ijitoye, and 11 others. 2025. An approach to technical agi safety and security. *arXiv preprint*.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, and 10 others. 2025. Open problems in mechanistic interpretability. *arXiv preprint*.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint*.

Mo Shen, Ryan McDonald, Daniel Zeman, and Peng Qi. 2016. Ud_chinese-gsd. *GitHub repository*.

Bowen Shi, Diane Brentari, Gregory Shakhnarovich, and Karen Livescu. 2022. Open-domain sign language translation learned from online video. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6365–6379, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Reza Shokri, Martin Strobel, and Yair Zick. 2021. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 231–241, New York, NY, USA. Association for Computing Machinery.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. 2019. BERT is not an interlingua and the bias of tokenization. In *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.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11521–11567, Bangkok, Thailand. Association for Computational Linguistics.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Ray Smith. 2023. tesseract: Open source ocr engine. `https://github.com/tesseract-ocr/tesseract`.

Nathalie A. Smuha. 2019. The eu approach to ethics guidelines for trustworthy artificial intelligence. *Computer Law Review International*, 20(4):97–106.

ByungHoon So, Kyuhong Byun, Kyungwon Kang, and Seongjin Cho. 2022. JaQuAD: Japanese question answering dataset for machine reading comprehension. *ArXiv preprint*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Anders Søgaard. 2021. Explainable natural language processing. *Synthesis Lectures on Human Language Technologies*, 14(3):1–123.

Anders Søgaard. 2022. Should we ban English NLP for a year? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5260, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anders Søgaard. 2023. On the opacity of deep neural networks. *Canadian Journal of Philosophy*, 53(3):224–239.

Anders Søgaard. 2025. Can machines be trustworthy? *AI and Ethics*, 5(1):313–321.

Nikita Soni, Lucie Flek, Ashish Sharma, Diyi Yang, Sara Hooker, and H. Andrew Schwartz, editors. 2024. *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*. ACL, TBD.

Paul Soulos, Sudha Rao, Caitlin Smith, Eric Rosen, Asli Celikyilmaz, R. Thomas McCoy, Yichen Jiang, Coleman Haley, Roland Fernandez, Hamid Palangi, Jianfeng

Gao, and Paul Smolensky. 2021. Structural biases for improving transformers on translation into morphologically rich languages. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*, pages 52–67, Virtual. Association for Machine Translation in the Americas.

Elizabeth C. Stade, Shannon Wiltsey Stirman, Lyle H. Ungar, Cody L. Boland, H. Andrew Schwartz, David B. Yaden, João Sedoc, Robert J. DeRubeis, Robb Willer, and Johannes C. Eichstaedt. 2024. Large language models could change the future of behavioral healthcare: a proposal for responsible development and evaluation. *npj Mental Health Research*, 3(1):12.

William C Stokoe. 1980. Sign language structure. *Annual review of anthropology*, 9(1).

Martin Strobel and Reza Shokri. 2022. Data privacy and trustworthy machine learning. *IEEE Security & Privacy*, 20(5):44–49.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Baohua Sun, Lin Yang, Catherine Chi, Wenhan Zhang, and Michael Lin. 2019. Squared english word: A method of generating glyph to use super characters for sentiment analysis. In *AffCon@AAAI*.

Jimin Sun, Patrick Fernandes, Xinyi Wang, and Graham Neubig. 2023. A multi-dimensional evaluation of tokenizer-free multilingual pretrained models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1725–1735, Dubrovnik, Croatia. Association for Computational Linguistics.

Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip S. Yu, and Caiming Xiong. 2020. Adv-bert: BERT is not robust on misspellings! generating nature adversarial samples on BERT. *arXiv preprint*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Yintao Tai, Xiyang Liao, Alessandro Suglia, and Antonio Vergari. 2024. PIXAR: Auto-regressive language modeling in pixel space. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14673–14695, Bangkok, Thailand. Association for Computational Linguistics.

Garrett Tanzer. 2025. Fingerspelling within sign language translation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 385–464, Albuquerque, New Mexico. Association for Computational Linguistics.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7564–7578, Dublin, Ireland. Association for Computational Linguistics.

Laia Tarrés, Gerard I. Gállego, Amanda Cardoso Duarte, Jordi Torres, and Xavier Giró-i-Nieto. 2023. Sign language translation from instructional videos. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5625–5635.

Kushal Tatariya, Vladimir Araujo, Thomas Bauwens, and Miryam de Lhoneux. 2024a. Pixology: Probing the linguistic and visual capabilities of pixel-based language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3307–3320, Miami, Florida, USA. Association for Computational Linguistics.

Kushal Tatariya, Artur Kulmizev, Wessel Poelman, Esther Ploeger, Marcel Bollmann, Johannes Bjerva, Jiaming Luo, Heather Lent, and Miryam de Lhoneux. 2024b. How good is your wikipedia? *arXiv preprint*.

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Tran, Dani Yogatama, and Donald Metzler. 2023. Scaling laws vs model architectures: How does inductive bias influence scaling? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12342–12364, Singapore. Association for Computational Linguistics.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*.

Owen Taylor. 2004. Pango, an open-source unicode text layout engine. In *Proceedings of the 25th Internationalization and Unicode Conference*, Washington, D.C., USA. The Unicode Consortium.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NeurIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Antonio Torralba, Phillip Isola, and William T. Freeman. 2024. *Foundations of Computer Vision*. Adaptive Computation and Machine Learning series. MIT Press.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. 2021. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. 2019. Fixing the train-test resolution discrepancy. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Florian Tramèr and Dan Boneh. 2021. Differentially private learning needs better features (or much more data). In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.

Asher Trockman and J Zico Kolter. 2023. Patches are all you need? *Transactions on Machine Learning Research*. Featured Certification.

Reut Tsarfaty, Dan Bareket, Stav Klein, and Amit Seker. 2020. From SPMRL to NMRL: What did we learn (and unlearn) in a decade of parsing morphologically-rich languages (MRLs)? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7396–7408, Online. Association for Computational Linguistics.

Michael Tschannen, Basil Mustafa, and Neil Houlsby. 2023. Image-and-language understanding from pixels only. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint*.

Christina O. Tze, Panagiotis P. Filntisis, Anastasios Roussos, and Petros Maragos. 2022. Cartoonized anonymization of sign language videos. In *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

David Uthus, Garrett Tanzer, and Manfred Georg. 2023. Youtube-asl: a large-scale, open-domain american sign language-english parallel corpus. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Bram Vaassen. 2022. Ai, opacity, and personal autonomy. *Philosophy & Technology*, 35(4):88.

Clayton Valli and Ceil Lucas. 2000. *Linguistics of American Sign Language: An Introduction*. Gallaudet University Press.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Daan van Esch, Elnaz Sarbar, Tamar Lucassen, Jeremy O'Brien, Theresa Breiner, Manasa Prasad, Evan Crew, Chieu Nguyen, and Françoise Beaufays. 2019. Writing across the world's languages: Deep internationalization for gboard, the google keyboard. *Technical report*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sahil Verma and Julia Sass Rubin. 2018. Fairness definitions explained. *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7.

Giorgos Vernikos and Andrei Popescu-Belis. 2021. Subword mapping and anchoring across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2633–2647, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA. Association for Computing Machinery.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *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 4396–4406, Hong Kong, China. Association for Computational Linguistics.

Ada Wan. 2022. Fairness in representation for multilingual NLP: Insights from controlled experiments on conditional language modeling. In *International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020a. Neural machine translation with byte-level subwords. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9154–9160. AAAI Press.

Cindy Wang and Michele Banko. 2021. Practical transformer-based multilingual text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 121–129, Online. Association for Computational Linguistics.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, and 11 others. 2023a.

Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. 2023b. Videomae v2: Scaling video masked autoencoders with dual masking. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14549–14560.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020b. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020c. Improving neural language generation with spectrum control. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Junqiu Wei, Qun Liu, Yinpeng Guo, and Xin Jiang. 2021. Training multilingual pre-trained language model with byte-level subwords. *arXiv preprint*.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *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 11–20, Hong Kong, China. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas. Association for Computational Linguistics.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Robert C. Williamson and Aditya Krishna Menon. 2019. Fairness risk measures. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 6786–6797, Long Beach, CA, USA. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *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 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Wei Wu, Yuxian Meng, Fei Wang, Qinghong Han, Muyu Li, Xiaoya Li, Jie Mei, Ping Nie, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *Neural Information Processing Systems*.

Zhaoyang Xia, Yuxiao Chen, Qilong Zhangli, Matt Huenerfauth, Carol Neidle, and Dimitri Metaxas. 2022. Sign language video anonymization. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources*, pages 202–211, Marseille, France. European Language Resources Association.

Zhaoyang Xia, Yang Zhou, Ligong Han, Carol Neidle, and Dimitris N. Metaxas. 2024. Diffusion models for sign language video anonymization. In *Proceedings of the LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages: Evaluation of Sign Language Resources*, pages 395–407, Torino, Italia. ELRA and ICCL.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. 2023. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1).

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *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.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint*.

Hao Yang, Yanyan Zhao, and Bing Qin. 2022a. Face-sensitive image-to-emotional-text cross-modal translation for multimodal aspect-based sentiment analysis. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language*

Processing, pages 3324–3335, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. 2022b. Enhancing cross-lingual transfer by manifold mixup. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.

Kaiyu Yang, Jacqueline H. Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. 2022c. A study of face obfuscation in ImageNet. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25313–25330. PMLR.

Linyi Yang, Yaoxian Song, Xuan Ren, Chenyang Lyu, Yidong Wang, Jingming Zhuo, Lingqiao Liu, Jindong Wang, Jennifer Foster, and Yue Zhang. 2023. Out-of-distribution generalization in natural language processing: Past, present, and future. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4533–4559, Singapore. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Chih-Kuan Yeh, Joon Sik Kim, Ian En-Hsu Yen, and Pradeep Ravikumar. 2018. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 9311–9321, Montréal, Canada. Curran Associates, Inc.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting . In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, Los Alamitos, CA, USA. IEEE Computer Society.

Yagmur Yigit, Mohamed Amine Ferrag, Mohamed C. Ghanem, Iqbal H. Sarker, Leandros A. Maglaras, Christos Chrysoulas, Naghmeh Moradpoor, Norbert Tihanyi, and Helge Janicke. 2025. Generative ai and llms for critical infrastructure protection: Evaluation benchmarks, agentic ai, challenges, and opportunities. *Sensors*, 25(6):1666.

Aoxiong Yin, Tianyun Zhong, Li Tang, Weike Jin, Tao Jin, and Zhou Zhao. 2023. Gloss Attention for Gloss-free Sign Language Translation . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2551–2562, Los Alamitos, CA, USA. IEEE Computer Society.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including signed languages in natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360, Online. Association for Computational Linguistics.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2021. Opacus: User-friendly differential privacy library in pytorch. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, Online.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022a. Differentially private fine-tuning of language models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022b. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. Megabyte: Predicting million-byte sequences with multiscale transformers. *arXiv preprint*.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, FangYuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in NLP: Benchmarks, analysis, and LLMs evaluations. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham. Springer International Publishing.

Amir Zeldes and Mitchell Abrams. 2018. The Coptic Universal Dependency treebank. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 192–201, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Erik Andersen, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, and 483 others. 2022. Universal dependencies 2.10. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, Hı̂órunn Arnardóttir, and 436 others. 2021. Universal dependencies 2.8. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. 2022. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, New Orleans, USA.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952.

Biao Zhang, Mathias Müller, and Rico Sennrich. 2023. SLTUNET: A simple unified model for sign language translation. In *The Eleventh International Conference on Learning Representations*.

Biao Zhang, Garrett Tanzer, and Orhan Firat. 2024. Scaling sign language translation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shiyue Zhang, Vishrav Chaudhary, Naman Goyal, James Cross, Guillaume Wenzek, Mohit Bansal, and Francisco Guzmán. 2022. How robust is neural machine translation to language imbalance in multilingual tokenizer training? *arXiv preprint*.

Wei Zhang, Ziming Huang, Yada Zhu, Guangnan Ye, Xiaodong Cui, and Fan Zhang. 2021. On sample based explanation methods for NLP: Faithfulness, efficiency and semantic evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5399–5411, Online. Association for Computational Linguistics.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2).

Yuting Zhao and Ioan Calapodescu. 2022. Multimodal robustness for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8505–8516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, and Nils Hammerla. 2019. Correlation coefficients and semantic textual similarity. In *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 951–962, Minneapolis, Minnesota. Association for Computational Linguistics.

Benjia Zhou, Zhigang Chen, Albert Clapés, Jun Wan, Yanyan Liang, Sergio Escalera, Zhen Lei, and Du Zhang. 2023. Gloss-free sign language translation: Improving from visual-language pretraining. In *ICCV*.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)*.

Tianyuan Zhou, João Sedoc, and Jordan Rodu. 2019. Getting in shape: Word embedding subspaces. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5478–5484, Macao, China. ijcai.org.

Wenxuan Zhou, Bill Yuchen Lin, and Xiang Ren. 2021a. Isobn: Fine-tuning BERT with isotropic batch normalization. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 14621–14629, Online. AAAI Press.

Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. 2021b. Do feature attribution methods correctly attribute features? In *XAI 4 Debugging Workshop at NeurIPS 2021*, Online. OpenReview.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

Daniel Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Benjamin Weinstein-Raun, Daniel de Haas, Buck Shlegeris, and Nate Thomas. 2022. Adversarial training for high-stakes reliability. In *Advances in Neural Information Processing Systems*.

George Kingsley Zipf. 1935. *The psycho-biology of language : an introduction to dynamic philology*. The psycho-biology of language: an introduction to dynamic philology. Houghton Mifflin, Oxford, England.

George Kingsley Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint*.