Multilingual Pretraining for Pixel Language Models

Ilker Kesen[†] **Jonas F. Lotz**[†,‡] **Ingo Ziegler**[†] **Phillip Rust**[†] **Desmond Elliott**[†]
[†]Department of Computer Science, University of Copenhagen
[‡]ROCKWOOL Foundation Research Unit
ilke@di.ku.dk

Abstract

Pixel language models operate directly on images of rendered text, eliminating the need for a fixed vocabulary. While these models have demonstrated strong capabilities for downstream cross-lingual transfer, multilingual pretraining remains underexplored. We introduce PIXEL-M4, a model pretrained on four visually and linguistically diverse languages: English, Hindi, Ukrainian, and Simplified Chinese. Multilingual evaluations on semantic and syntactic tasks show that PIXEL-M4 outperforms an English-only counterpart on non-Latin scripts. Word-level probing analyses confirm that PIXEL-M4 captures rich linguistic features, even in languages not seen during pretraining. Furthermore, an analysis of its hidden representations shows that multilingual pretraining yields a semantic embedding space closely aligned across the languages used for pretraining. This work demonstrates that multilingual pretraining substantially enhances the capability of pixel language models to effectively support a diverse set of languages.

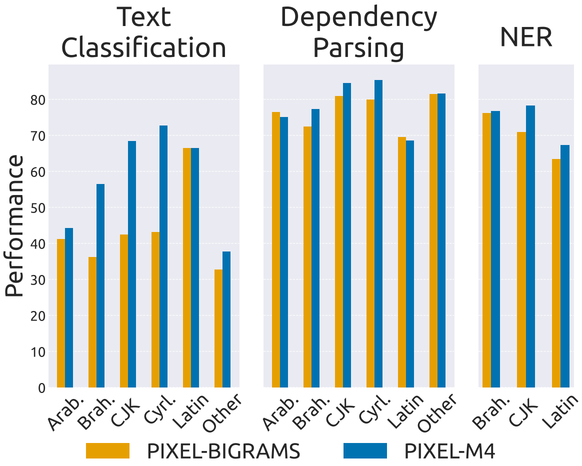

Figure 1: Average performance across tasks comparing PIXEL-M4 and PIXEL-BIGRAMS grouped by scripts: Arabic, Brahmic, Chinese-Japanese-Korean, Cyrillic, Latin, and others. Both models share the same architecture and hyperparameters, but PIXEL-M4 is pretrained in four visually and linguistically diverse languages: English, Hindi, Ukrainian and Simplified Chinese. PIXEL-M4 performs better in almost all non-Latin script languages without sacrificing Latin-script performance.

1 Introduction

Visually-rendered text has emerged as an alternative to sub-word tokenization for language models (Salesky et al., 2021; Rust et al., 2023). In comparison to sub-word tokenization, processing visually-rendered text enables models to transfer to unseen languages without needing to initialize new embeddings (Dobler and de Melo, 2023), or relying on back-off mechanisms based on bytes (Xue et al., 2022) or characters (Clark et al., 2022). Previous work on pixel-based language models has predominantly focused on monolingual pretraining on English data (Rust et al., 2023; Lotz et al., 2023), with related efforts extending to multilingual pretraining for machine translation (Salesky et al., 2023). Given evidence that pixel-based models facilitate positive transfer through visual similarity (Lotz et al., 2025; Muñoz-Ortiz et al., 2025), we investigate multilingual pretraining for general-purpose representation learning specifically by selecting only one language per script. This approach is particularly valuable for low-resource languages that can benefit from transfer via visually similar, high-resource languages.

We present PIXEL-M4: a multilingual version of PIXEL (Rust et al., 2023). PIXEL-M4 is pretrained on four equally-sized amounts of visually diverse scripts sourced from mC4 (Xue et al., 2021): English (Latin script), Hindi (Devanagari script), Simplified Chinese (Han script), and Ukrainian (Cyrillic script). These scripts were chosen to represent abugida, alphabetic, logographic/logosyllabic writing systems, covering billions of speakers. Furthermore, not only do these scripts represent visual diversity, they also represent grammatical diversity,

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 29594–29611
November 4-9, 2025 ©2025 Association for Computational Linguistics

covering Balto-Slavic, Indo-Iranian, Germanic, and Sino-Tibetan languages.

In downstream task experiments, we investigate the ability of PIXEL-M4 to transfer to new languages in three conditions (i) same-script; (ii) related-script; and (iii) unrelated scripts to better understand what is gained by multilingual pretraining.[1] The same-script experiments focus on Simplified Chinese (Han), Hindi (Devanagari), and various Latin and Cyrillic script languages. The related-script experiments include Japanese and Brahmic script languages; while the unrelated-script experiments focus on Armenian, Greek, Korean and languages using the abjad writing system (e.g. Arabic and Hebrew). Compared to its monolingually-pretrained equivalent, PIXEL-BIGRAMS (Lotz et al., 2023), we find consistent improvements in performance for almost all non-Latin script languages on text classification, dependency parsing and named entity recognition.

We conduct word-level probing experiments using LINSPECTOR (Şahin et al., 2020) to compare differences in linguistic understanding across 15 languages from multilingual versus monolingual pretraining. We find that PIXEL-M4 captures linguistic features more effectively than PIXEL-BIGRAMS, both for seen scripts (e.g., Russian and Macedonian) and unseen scripts (Arabic, Armenian, Greek). Additionally, an exploration of PIXEL-M4's embedding space reveals that earlier layers primarily encode visual information, while deeper layers shift toward semantic understanding, corroborating earlier observations by Tatariya et al. (2024). Through cross-lingual retrieval experiments, we find that PIXEL-M4 has learned a semantic representation space that is shared across the pretraining languages.

In short, the main contributions of this paper are:

- We present the first multilingually-pretrained general-purpose pixel language model,[2] trained over four visually and linguistically diverse languages.

- Experiments on syntactic and semantic tasks show consistent improvements for non-Latin script languages compared to previous PIXEL language models.

- Word-level probing analyses show that multilingual pretraining produces representations that capture more linguistic features across languages, such as case marking, part-of-speech tags, and verb tense.

- Sentence-level analyses of the learned hidden representations reveal that PIXEL-M4 has learned a representation space highly aligned between a subset of its pretraining languages.

2 PIXEL-M4

2.1 Pretraining Data

Following our motivation to explore multilingual pretraining through a diverse selection of scripts rather than a large range of languages, PIXEL-M4 is pretrained on text written in Latin (English), Cyrillic (Ukrainian), Simplified Chinese characters (Chinese), and Devanagari (Hindi). For each script, a corresponding subset of the mC4 (Xue et al., 2021) corpus is rendered into images, following the strategy of rendering two characters per image patch from Lotz et al. (2023). With a sequence length of 529 image patches and a batch size of 256, the model observes approximately 135 billion image patches over 1 million pretraining steps – this is the same total amount of data as the original PIXEL and PIXEL-BIGRAMS models. However, PIXEL-M4 is trained on an order-of-magnitude more unique samples than PIXEL-BIGRAMS. This difference is due to the fact that PIXEL-BIGRAMS was trained by iterating 10 times over the English-only Wikipedia + BookCorpus datasets (Zhu et al., 2015), whereas PIXEL-M4 processes each sample in our subset of mC4 only once across the four pretraining languages.

2.2 Pretraining Procedure

Both PIXEL-M4 and PIXEL-BIGRAMS follow the PIXEL pretraining recipe from Rust et al. (2023), including hyperparameter values. Based on the Masked Autoencoding Vision Transformer (He et al., 2022), the models render each input sequence to a 529-patch image using the PangoCairo rendering library,[3] where each image patch is 16 × 16 pixels. We use the Google Noto Sans fonts collection to ensure that the majority of Unicode codepoints can be accurately rendered.[4] PIXEL-M4 is trained by mixing the four languages within each batch;

[1] The downstream task languages also cut across different language families, e.g. Indo-European, Sino-Tibetan, and Turkic. However, we focus on script transfer, given the visual nature of the data processed by PIXEL-M4.

[2] Code and models: ilkerkesen/pixel-m4

[3] https://docs.gtk.org/PangoCairo

[4] https://fonts.google.com/noto

however, each individual sample consists of only one language. The image patches are first embedded through a linear projection, 25% of them are masked (in spans of up to 6 consecutive patches), and only the unmasked patches plus a CLS token are passed to the encoder. After the encoder, a lightweight decoder reconstructs the pixel values of only the masked patches. For downstream tasks we remove the decoder and instead attach a task-specific head, and disable patch masking in inputs.

3 Experimental Setup

3.1 Tasks & Benchmarks

Text Classification. We first test the models on the sentence-level semantic task of topic classification using the SIB-200 benchmark (Adelani et al., 2024). Each example in SIB-200 is semantically aligned across languages. This aspect of SIB-200 allows us to make a controlled comparison across different languages and scripts. Our first set of evaluations cover the four pretraining languages of PIXEL-M4: Latin (English ENG), Han (Chinese ZHO), Cyrillic (Ukrainian UKR), and Devanagari (Hindi HIN). For the same-script transfer setting, we experiment with Latin script languages (German DEU, Finnish FIN, French FRA, Turkish TUR, Uzbek UZN) and Cyrillic script languages (Kyrgyz KIR, Russian RUS). For the related-script transfer setting, we perform experiments in Japanese (JPN) and Brahmic script languages (Bengali BEN, Standard Tibetan BOD, Tamil TAM, Telugu TEL). Lastly, we cover Armenian (HYE), Greek (ELL), Hebrew (HEB), Korean (KOR) and Arabic script languages (Egyptian Arabic ARZ, Uyghur UIG, Urdu URD) to test transfer to unrelated novel scripts. We report macro-averaged F1 score as the metric.

Dependency Parsing. We evaluate on the token-level syntactic parsing task of dependency parsing using the Universal Dependencies (UD) benchmark (Nivre et al., 2020; Zeman et al., 2022). We also compare the models using the same three transfer learning settings again: (i) same-script languages seen during pretraining: Latin (English ENG, Vietnamese VIE), Devanagari (Hindi HIN), Han (Chinese ZHO), and Cyrillic (Ukrainian UKR, Russian RUS, Bulgarian BUL); (ii) languages in scripts related to at least one pretraining script: Coptic (COP), Japanese (JPN) and Brahmic script languages (Tamil TAM, Telugu TEL); (iii) languages in scripts unrelated to the pretraining scripts: Arabic abjad (Arabic ARA, Urdu URD) and Korean (KOR). We report Labeled Attachment Score (LAS) as the evaluation metric.

Named Entity Recognition. Lastly, we perform experiments on the token-level semantic task of Named Entity Recognition (NER) using three benchmarks: the multilingual Universal NER (Mayhew et al., 2024, UNER) and Naamapadam (Mhaske et al., 2023) benchmarks, as well as the NER portion of the Korean Language Understanding Evaluation (Park et al., 2021, KLUE). Once again, we cover same-script, related-script and unrelated-script transfer scenarios. Here, three of the four scripts seen during pretraining – Latin (English ENG, Serbian SRP), Han (Chinese ZHO), and Devanagari (Hindi HIN) – are additionally evaluated on Korean KOR, as well as three Brahmic scripts (Bengali BEN, Tamil TAM, Telugu TEL). We report macro-averaged F1 scores.

3.2 Baselines

We mainly compare PIXEL-M4 against the monolingual PIXEL-BIGRAMS model, which is trained exclusively on English text rendered at the bigram level. PIXEL-M4 implements the identical architecture, text rendering strategy and pretraining procedure with the same set of hyperparameters, but PIXEL-M4 is multilingually pretrained on equal amounts of English, Hindi, Ukrainian and Simplified Chinese. This comparison allows us to observe the effects of multilingual pretraining for pixel language models in different transfer learning settings.

We also compare PIXEL-M4 against four monolingual BERT variants: The original English BERT (Devlin et al., 2019) primarily for the Latin languages, Chinese BERT (Devlin et al., 2019) for Han and Japanese scripts, a Hindi BERT (Samuel et al., 2023) for the Brahmic script languages, and a Ukrainian BERT (Samuel et al., 2023) for the Cyrillic languages. English BERT is also used as a fallback option to evaluate languages that do not belong to any of the pretraining scripts, such as Arabic or Hangul. This allows us to test whether multilingually-pretrained pixel models can match or exceed the cross-lingual transfer capabilities of the tokenizer-based models, not only for Latin scripts but also for others.

Lastly, we include CANINE-S (Clark et al., 2022) as a character-level multilingual baseline, which is comparable in size to PIXEL-M4. However, CANINE-S is pretrained with a subword loss objective on 104 languages using the same multilingual

	Arabic			Brahmic					Cyrillic		
	ARZ	UIG	URD	BEN	BOD	HIN	TAM	TEL	KIR	RUS	UKR
BERT-MONO	29.1	43.9	31.1	38.4	40.7	**87.2**	48.6	29.5	**73.5**	**83.8**	**86.5**
CANINE-S	**65.1**	53.5	**55.5**	**54.5**	39.3	53.4	57.5	44.5	65.5	75.9	74.7
PIXEL-BIGRAMS	38.3	48.6	36.5	31.7	36.9	32.6	39.7	39.9	47.1	37.7	44.4
PIXEL-M4	37.5	**53.7**	41.6	46.2	**46.3**	78.6	**64.5**	**46.6**	62.9	74.7	80.5

	Latin						CJK			Others			Avg.
	DEU	ENG	FIN	FRA	TUR	UZN	ZHO	JPN	KOR	ELL	HEB	HYE	
BERT-MONO	63.8	**88.1**	43.5	76.1	62.7	59.4	**89.5**	**78.9**	15.4	32.6	32.7	36.5	55.3
CANINE-S	**77.7**	78.4	**61.6**	**77.4**	**68.2**	**63.6**	77.9	75.3	64.5	**65.3**	**54.4**	**60.8**	**63.7**
PIXEL-BIGRAMS	63.8	84.3	59.7	73.2	60.7	56.7	48.5	41.0	37.8	34.3	26.7	37.3	46.0
PIXEL-M4	67.3	83.9	60.6	70.7	59.9	56.2	75.5	65.0	**64.7**	36.9	31.3	44.8	58.7

Table 1: Text classification results on a selected language subset of the SIB-200 benchmark using macro F1-score. BERT-MONO indicates that the monolingual BERT model varies by language (see §3.2 for details). Best performances are bolded. PIXEL-M4 significantly outperforms its English-only-pretrained equivalent PIXEL-BIGRAMS in almost all non-Latin languages, and PIXEL-M4 performs better than monolingual BERT models on novel writing systems. Even though CANINE-S is pretrained on 104 languages, PIXEL-M4 outperforms it on 8 of 23 languages.

Wikipedia corpus as mBERT (Devlin et al., 2019) whereas PIXEL-M4 has only seen four languages in pretraining. Consequently, this difference in pretraining data and languages prevents a fully fair comparison of these models.

4 Results and Discussion

Text Classification. Table 1 presents the results on SIB-200 for text classification. PIXEL-M4 outperforms PIXEL-BIGRAMS by large margins in its pretraining languages (HIN: +46.0, UKR: +36.1, ZHO: +27.0), which are unseen by PIXEL-BIGRAMS during the pretraining. We also observe substantial gains in Cyrillic languages (KIR: +15.8, RUS: +37.0), showing that pretraining pixel models on a particular script enhances transfer learning within the same-script languages. In English and other Latin languages, both models achieve similar performances. The significant performance gains in Japanese (JPN: +24) and the Brahmic languages (BEN: +14.5, BOD: +9.4, TAM: +24.8, TEL: 6.7) showcase PIXEL-M4's cross-lingual transfer learning ability to novel scripts orthographically related to one pretraining script. Lastly, we compare both PIXEL-M4 and PIXEL-BIGRAMS in languages with writing systems visually distant to the pretraining scripts. Once again, PIXEL-M4 performs better than PIXEL-BIGRAMS in these languages, where we can observe improvements for Armenian (HYE: +7.5), Greek (ELL: +4.3), Korean (KOR: +26.9) and the languages in right-to-left abjad writing systems (HEB: 4.6, UIG: +5.1, URD: +5.1). These results illustrate that multilingual pretraining with a diverse set of scripts accelerates cross-lingual generalization even for novel and distant writing systems. Overall, these results highlight that visually and linguistically diverse multilingual pretraining for pixel models leads to substantial gains in all types of transfer learning scenarios investigated in this work.

Compared to the monolingual BERT variants, PIXEL-M4 performs consistently better, especially in the transfer learning setting involving unseen scripts. Conversely, BERT-MONO models surpass PIXEL-M4 in transfer learning within the same-script, yet BERT-MONO pretrained in English falls behind PIXEL-M4 in German (DEU: +3.5) and Finnish (FIN: +17.1).

We further compare PIXEL-M4 against character-level multilingually-pretrained CANINE-S. PIXEL-M4 outperforms CANINE-S on 3 out of 4 pretraining languages and 4 out of 5 Brahmic languages, as well as slight gains in KOR and UIG. These gains are notable given that CANINE-S was pretrained on all of these languages except ARZ and BOD.

Dependency Parsing. Table 2 presents the results on the UDP benchmark. In the pretraining languages, PIXEL-M4 significantly improves upon

	Arabic		Brahmic			Cyrillic			Latin		CJK			Other	Avg.
	ARA	URD	HIN	TAM	TEL	BUL	RUS	UKR	ENG	VIE	ZHO	JPN	KOR	COP	
BERT-MONO	**77.7**	71.9	**92.8**	43.4	75.6	**89.8**	**87.5**	**92.0**	**90.6**	**49.4**	**85.5**	87.9	30.2	13.0	70.5
CANINE-S	75.5	73.0	86.2	48.2	77.7	82.1	80.3	74.9	82.4	41.8	75.1	**91.2**	77.7	68.8	73.9
PIXEL-BIGRAMS	**77.7**	75.3	88.6	49.8	79.0	86.3	79.1	74.4	89.6	**49.4**	73.9	90.8	78.1	81.4	76.7
PIXEL-M4	74.2	**75.9**	91.6	**60.5**	**79.7**	88.8	83.0	84.3	87.6	**49.4**	79.9	**91.2**	**82.3**	**81.6**	**79.3**

Table 2: Dependency parsing results for the selected set of languages in the UDP benchmark with LAS. BERT-MONO indicates that the monolingual BERT model varies by language. PIXEL-M4 outperforms PIXEL-BIGRAMS in non-Latin script languages, and it again achieves a better performance than BERT-MONO on novel scripts, while showing substantial gains over CANINE-S on every language but one.

	Latin		Brahmic				CJK		Avg.
	ENG	SRP	HIN	BEN	TAM	TEL	KOR	ZHO	
BERT-MONO	**79.3**	**85.8**	**82.5**	75.4	67.3	78.3	30.6	**85.4**	73.1
CANINE-S	67.1	83.0	79.5	**78.5**	**69.7**	**80.5**	80.6	75.8	**76.8**
PIXEL-BIGRAMS	63.4	81.6	79.0	78.0	67.9	79.6	80.4	61.4	73.9
PIXEL-M4	67.3	82.1	80.9	**78.5**	68.0	79.6	**81.6**	74.9	76.6

Table 3: NER results by macro-averaged F1-scores. BERT-MONO is the monolingual BERT model, which varies by language depending on script. Overall, PIXEL-M4 performs on par with CANINE-S, and outperforms PIXEL-BIGRAMS and BERT-MONO with an average score of 76.6.

PIXEL (HIN: +3.0, UKR: +9.9, ZHO: +6.0) except in English (ENG: -2.0), which both models have seen in their pretraining. PIXEL-M4 outperforms PIXEL on the languages written in Cyrillic (BUL: +2.5, RUS: +3.9), which demonstrates improved cross-lingual transfer learning within the same-script languages once again. For the unseen Brahmic languages, PIXEL-M4 achieves a slight gain in Telugu (TEL: +0.7) and a much larger performance boost in Tamil (TAM: +10.7). For the orthographically distant Korean language, PIXEL-M4 outperforms PIXEL-BIGRAMS (KOR: +4.2). For the Arabic-script languages, we observe mixed results: In Arabic, the performance drops (ARA: -3.5), while we observe a modest gain in Urdu (URD: +0.6). Altogether, multilingually-pretrained PIXEL-M4 improves on PIXEL-BIGRAMS on the dependency parsing task for the unseen languages considering various cross-lingual transfer learning settings. Our findings on this task is similar to the SIB-200 findings for comparing PIXEL-M4 against monolingual BERT models: (i) PIXEL-M4 achieves a better overall performance than BERT-MONO in cross-lingual transfer involving writing systems unknown to both; (ii) BERT-MONO performs better than PIXEL-M4 for the pretraining scripts and cross-lingual transfer within the same-script. Lastly, when compared with CANINE-S, PIXEL-M4 achieves consistently better overall performance, despite the fact that CANINE-S has been trained on all languages except COP.

Named Entity Recognition. Table 3 reports macro-averaged F1 for NER across eight languages. As expected, multilingual pixel pretraining (PIXEL-M4) outperforms the English-only PIXEL-BIGRAMS model on every language, raising the average F1 from 73.9 to 75.9. The largest boost is seen in Chinese (ZHO: +13.5), reflecting that exposure to Chinese during PIXEL-M4's pretraining. Other pretraining languages also benefit from multilingual pretraining (ENG: +3.9, HIN: +1.9). Differently from the other tasks, both PIXEL-M4 and PIXEL-BIGRAMS perform on par in the Brahmic scripts (HIN: +1.9, BEN: +0.5, TAM: +0.1, TEL: 0.0): This might be due the larger training sets available in the Naamapadam benchmark. Later, in §5, we show that PIXEL-M4 outperforms PIXEL-BIGRAMS with large margins in low-resource settings. Lastly, +1.2 gain in Korean suggests that PIXEL-M4 can transfer visual substructure from unrelated scripts for better entity processing.

The monolingual BERT models achieve a better performance than PIXEL-M4 for the languages with writing systems known by both models, underscoring that world-knowledge and semantic co-occurrence patterns encoded into specific token entities remain crucial for this semantic task. This is especially the case for English, as both BERT and PIXEL-BIGRAMS are pretrained using exactly the same data. Nonetheless, our findings for the languages in unseen scripts is inline with previous experiments where PIXEL-M4 performs better than BERT-MONO: (BEN: +3.1, TAM: +0.7, TEL: 1.6). These improvements highlight how pixel

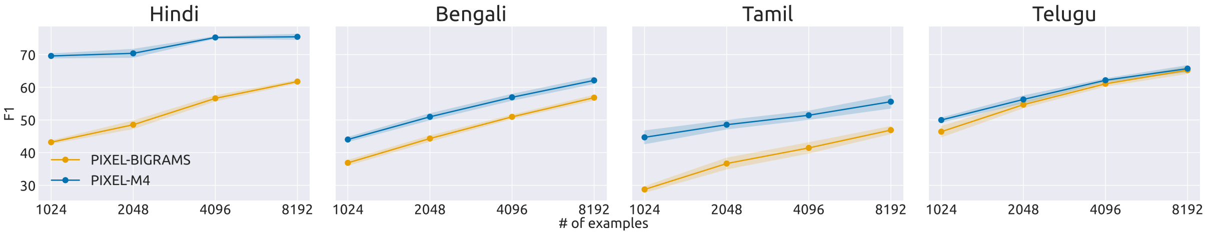

Figure 2: Data-efficient learning experiments on the Naamapadam NER benchmark showing the mean test F_1 score as a function of training set size in log scale for four Brahmic languages. In each experiment, PIXEL-M4 consistently outperforms PIXEL-BIGRAMS, with the largest relative gains under the smallest data regimes.

models can process languages in related scripts directly, avoiding the tokenization failure modes of subword-based models.

Finally, PIXEL-M4 performs on par with CANINE-S overall, although CANINE-S was exposed to all downstream task languages during pretraining.

5 Analysis

5.1 Data-Efficiency Analysis

To investigate the capabilities of PIXEL-M4 further, we perform a data-efficiency analysis on Naamapadam – the Indic languages benchmark. Using the original training splits, we create subsets of size 1024, 2048, 4096 and 8192 examples. We repeat this process 8 times using different random seeds, resulting 32 different subsets. Next, we train both PIXEL-BIGRAMS and PIXEL-M4 on these subsets and compare them in terms of data-efficiency. Figure 2 illustrates this comparison, where each subplot represents the results for the specified language. For Hindi, Bengali and Tamil, PIXEL-M4 performs significantly better than PIXEL in all settings. The results in Bengali and Tamil also highlight the cross-lingual transfer learning capacity of the PIXEL-M4 in low-resource settings. As we decrease the number of examples, we observe more substantial gains in all languages including Telugu, where PIXEL-M4 performs slightly better than PIXEL-BIGRAMS on the entire set of tasks. Overall, multilingual pretraining of pixel language models substantially enhances transfer learning in low-resource settings.

5.2 Word-Level Probing

We also performed a probing analysis similar to Tatariya et al. (2024). Here, we use LINSPECTOR (Şahin et al., 2020), a multilingual word-level probing benchmark, to investigate the transferability of multilingual representations encoded by PIXEL-M4. We investigate hidden representations encoded by both PIXEL-M4 and PIXEL-BIGRAMS after each layer, and compare them against each other. We perform this analysis on four different tasks (Case Marking, POS, SameFeat, TagCount) using five different languages (Arabic, Armenian, Greek, Russian, Macedonian).[5] *Case Marking* requires assessing the grammatical case (e.g. nominative, accusative) of a given input word. *POS* involves predicting the POS tag for the given word. The *SameFeat* task measures the ability to detect the mutual morphological feature of two given words in their surface forms. Lastly, *TagCount* requires correctly predicting the number of morphological tags for the given input word. *SameFeat* and *TagCount* are more difficult than the other tasks, as both require predicting the entire set of morphological features for the given word(s).

We show the results of our probing analyses in Figure 3. In this grid of subplots, each row investigates a different task, and each column investigates a different language. In Macedonian and Russian, PIXEL-M4 learns significantly better representations compared to PIXEL-BIGRAMS, which is expected because PIXEL-M4 has seen a similar language in the same script during pretraining. The gap between two models in earlier layers (1-3) is smaller on *SameFeat* and *TagCount*, as they require more complex linguistic assessment. This also applies for the other tested languages, and it is in line with the observations of Tatariya et al. (2024), where earlier layers focus more on visual rather than semantic processing. In Arabic, Armenian, and Greek, PIXEL-M4 still performs slightly better than PIXEL-BIGRAMS on the majority of tasks, which showcases its improved visual processing and transfer learning to unseen languages. For these unseen languages, the performance of PIXEL-M4 starts to plateau starting from the 7th or

[5] See Appendix for a larger set of tasks and languages.

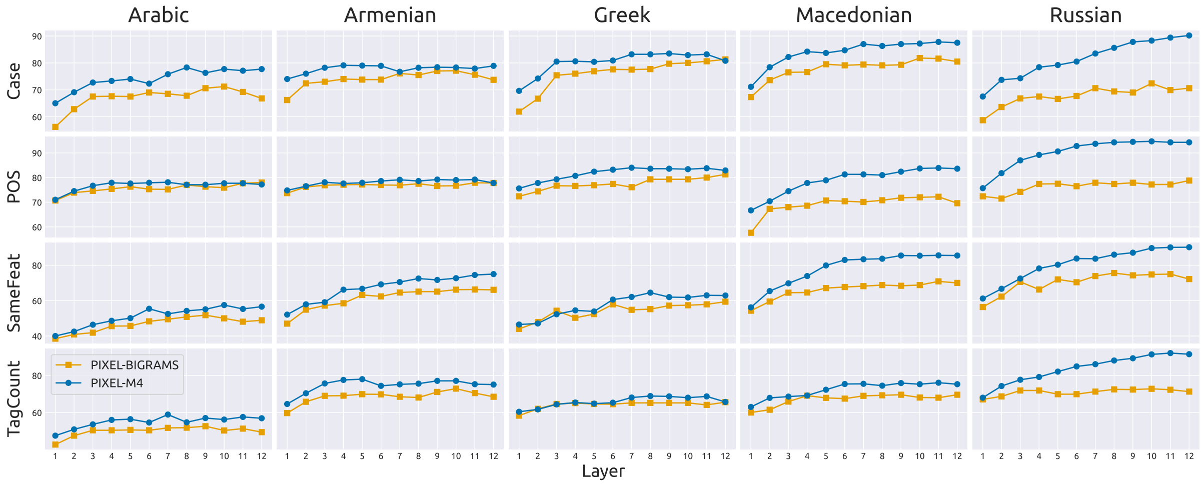

Figure 3: Word-level probing analysis on LINSPECTOR, where each row investigates a different task, and each column investigates a different language. In each subplot, y-axis represents the model accuracies and x-axis represents the corresponding layer number for the used hidden representations. Multilingually-pretrained PIXEL-M4 has learned better linguistic representations even for the languages with orthographically distant writing systems.

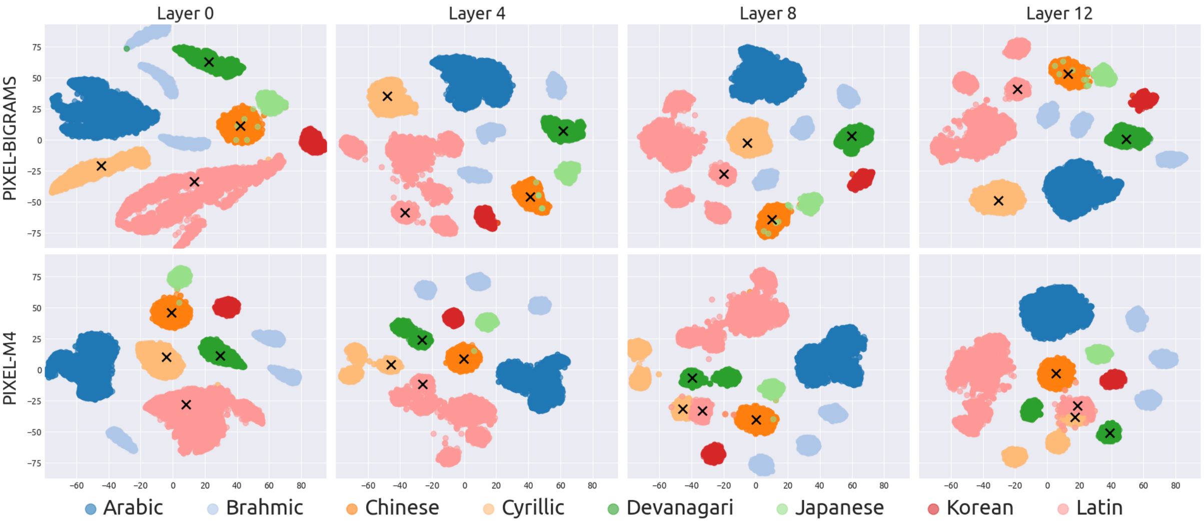

Figure 4: t-SNE visualization of the outputs for the specified layers. Each row contains visualizations for a particular model, and each column focuses on a particular layer. Each '×' marker appear at the centroid of a different pretraining language seen by PIXEL-M4. Both models cluster languages based on their scripts, yet PIXEL-M4 clusters some pretraining languages in the later layers.

8th layer. Overall, these results demonstrate that the multilingual pretraining produces a better set of hidden representations throughout the entire model, even for the unseen scripts.

5.3 Analyzing Hidden Representations

Similar to Salesky et al. (2023), we visualize the hidden representations learned by both PIXEL-BIGRAMS and PIXEL-M4 using t-SNE (Van der Maaten and Hinton, 2008). To perform this analysis, we use a subset of SIB-200 (Adelani et al., 2024) including the training splits of 26 languages. We perform t-SNE visualization throughout the model, starting from the convolved input representations (Layer 0) to the output of the last transformer layer (Layer 12). Figure 4 shows t-SNE plots: rows correspond to models, columns to layers, and '×' marks the PIXEL-M4 pretraining-language centroids. We observe the same phenomenon for the convolved features as demonstrated in Salesky et al. (2023): Languages which use the same or a related writing script are grouped together. This can be observed for both models, where we can see large clusters for Arabic, Cyrillic and Latin, and Chinese-Japanese language clusters appear next to each other. As we move through in

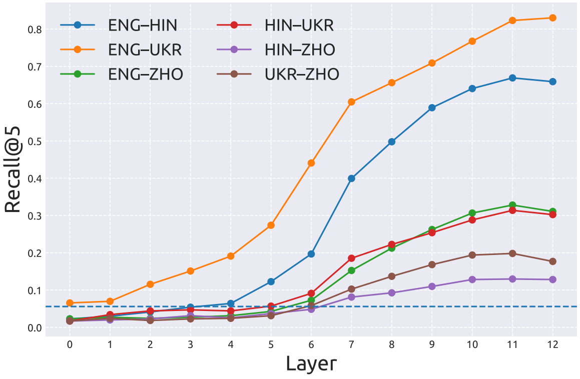

Figure 5: Cross-lingual similarity analysis on SIB-200 using the mean pooled hidden representations of PIXEL-M4. The x-axis indicates the layer number; the y-axis reports the performance using recall@5. Each line focuses on a different language-pair combination. The dashed line shows the maximum recall@5 value obtained by PIXEL-BIGRAMS for these language pairs. This analysis reveals that PIXEL-M4 has learned a mutual semantic representation for some pretraining language pairs.

the model layers, we start to see some languages form their own separate clusters by moving away from their script clusters (e.g. Layer 4 and 8). More importantly, in the later layers of PIXEL-M4, we observe that the pretraining languages move away from the rest of the languages that share the same script, and they start to cluster together. This observation demonstrates that PIXEL-M4 shifts its focus from visual processing more to the semantics in the later layers. This raises the question of whether PIXEL-M4 has learned a semantic representation space shared between different pretraining languages.

To determine whether PIXEL-M4 has learned a representation space shared between different pretraining languages, we perform a cross-lingual retrieval experiment on the multilingually aligned SIB-200 benchmark. To obtain sentence embeddings, we apply L2 normalization to the mean pooled hidden representations after each layer. At each layer, we treat each sentence embedding in one language as a query and compute its cosine similarity against every sentence embedding in the other language. We report recall@5, i.e., the percentage of the examples where the true translation is ranked in the top 5. Since each sentence has exactly one correct translation, retrieval performance per example is binary, taking values of either 0 or 1. Figure 5 shows the results for each language pair. We see that the semantic alignment between each language pair increases as we move through in the layers. Particularly, the semantic alignment between English and Ukrainian is very high, as they are also tightly clustered in the t-SNE feature space. We can also observe a high semantic alignment between English and Hindi, yet the remaining pairs do not share a highly aligned semantic representation space.

6 Related Work

Salesky et al. (2021) proposed an encoder-decoder-based machine translation model that replaces the tokenizer in the encoder by processing source text as rendered images. Rust et al. (2023) proposed PIXEL, the first model that relies on purely processing visually rendered text. Later, Lotz et al. (2023) investigated different strategies for text rendering with the aim of removing redundant patches. Fei et al. (2024) experimented with replacing BERT's tokenizer with pixel-based processing. Gao et al. (2024) extended PIXEL with a mixed modality pretraining objective, which produced substantial improvements. Tai et al. (2024) pretrained PIXAR, which is the first autoregressive pixel language model that purely relies on processing rendered text. Gao et al. (2024); Chai et al. (2024) also proposed pixel language models with text generation abilities, yet they achieved this by still depending on subword tokenizers. Recently, Lotz et al. (2025) embedded pixel language models into the English-centric language models as a fallback mechanism to better adapt these models to novel languages and scripts. Most notably, Salesky et al. (2023) is closely related to our work as it employs a multilingual pretraining. However, their experiments focus on learning a shared encoder for machine translation, while we pretrained a multilingual pixel language model for general-representation learning without relying on any tokenizer.

7 Conclusion

In this work, we explored multilingual pretraining for pixel language models. We pretrained PIXEL-M4, a multilingual pixel-based language model on four visually and linguistically diverse languages, namely English, Hindi, Ukrainian and Simplified Chinese. We performed downstream task experiments on three different tasks: sentence classification, dependency parsing, and named entity recognition. In these experiments, we covered a diverse set of languages and scripts, where we evaluated on 27 languages and 15 scripts. Our experiments revealed that PIXEL-M4 achieves superior performance in low-resource settings compared to its monolingually-pretrained predecessor PIXEL-BIGRAMS, outperforming it in almost all non-Latin languages by a large margin. In order to better understand the representations learned by PIXEL-M4, we conducted word-level and sentence-level analyses. Our word-level probing analysis illustrated that PIXEL-M4 has learned better hidden representations than PIXEL-BIGRAMS throughout the network for the unseen scripts, highlighting its cross-lingual transfer capabilities. Additionally, an analysis on the hidden layer representations revealed that PIXEL-M4 has learned a semantic representation space shared by a subset of pretraining languages in the later layers. In future work, we aim to scale up multilingual pretraining for pixel models with larger model capacity and more languages included in pretraining.

Limitations

PIXEL-M4 inherits many of the limitations of its predecessors. First, rendering text using the bigrams strategy leads to increased sequence lengths when a bigram does not fit into single patch. Like Rust et al. (2023) and Lotz et al. (2023), PIXEL-M4 cannot generate text. The improvements over PIXEL-BIGRAMS are also limited for Latin-script languages and also for high-resource settings. Due to our limited compute budget, we pretrained a single PIXEL-M4 model on only four languages-each in a different script. Consequently, we have not explored larger or different combinations of languages and scripts, such as additional Latin-script languages (e.g. French, Estonian, Turkish) or right-to-left scripts (e.g. Hebrew, Arabic). Finally, the comparison between PIXEL-M4 and PIXEL-BIGRAMS is not entirely fair, as PIXEL-M4 was exposed to more data. Nonetheless, pretraining another monolingual model on an equivalent amount of data for such a comparison was not feasible within our computational budget. We leave these comparisons to future work.

Acknowledgements

IK, IZ and DE were supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 101135671 (TrustLLM). JFL is funded by the ROCKWOOL Foundation (grant 1242). DE was supported by a research grant (VIL53122) from VILLUM FONDEN. PR is funded by the Novo Nordisk Foundation (grant NNF 20SA0066568).

IK and IZ acknowledge the EuroHPC Joint Undertaking for awarding access to MareNostrum5, hosted at Barcelona Supercomputing Center (BSC), Spain, under proposals No. EHPC-DEV-2024D11-047 and EHPC-DEV-2024D12-031.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, and 3 others. 2016. Tensorflow: A system for large-scale machine learning. *Preprint*, arXiv:1605.08695.

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2024. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta. Association for Computational Linguistics.

Yekun Chai, Qingyi Liu, Jingwu Xiao, Shuohuan Wang, Yu Sun, and Hua Wu. 2024. Autoregressive pretraining on pixels and texts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3106–3125, Miami, Florida, USA. Association for Computational Linguistics.

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.

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.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454, Singapore. Association for Computational Linguistics.

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.

Wenlong Fei, Xiaohua Wang, Min Hu, Qingyu Zhang, and Hongbo Li. 2024. MTLS: Making texts into linguistic symbols. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3521–3535, Miami, Florida, USA. Association for Computational Linguistics.

Tianyu Gao, Zirui Wang, Adithya Bhaskar, and Danqi Chen. 2024. Improving language understanding from screenshots. *Preprint*, arXiv:2402.14073.

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.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 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. *Preprint*, arXiv:2504.02122.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Suppa, Hila Gonen, Joseph Marvin Imperial, Börje Karlsson, Peiqin Lin, Nikola Ljubešić, Lester James Miranda, Barbara Plank, Arij Riabi, and Yuval Pinter. 2024. Universal NER: A gold-standard multilingual named entity recognition benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4322–4337, Mexico City, Mexico. Association for Computational Linguistics.

Arnav Mhaske, Harshit Kedia, Sumanth Doddapaneni, Mitesh M. Khapra, Pratyush Kumar, Rudra Murthy, and Anoop Kunchukuttan. 2023. Naamapadam: A large-scale named entity annotated data for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10441–10456, Toronto, Canada. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Verena Blaschke, and Barbara Plank. 2025. Evaluating pixel language models on non-standardized languages. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6412–6419, Abu Dhabi, UAE. Association for Computational Linguistics.

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.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint*.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, and 14 others. 2021. Klue: Korean language understanding evaluation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 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*.

Gözde Gül Şahin, Clara Vania, Ilia Kuznetsov, and Iryna Gurevych. 2020. Linspector: Multilingual probing tasks for word representations. *Computational Linguistics*, 46(2):335–385.

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. 2023. 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.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

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.

Kushal Tatariya, Vladimir Araujo, Thomas Bauwens, and Miryam de Lhoneux. 2024. 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.

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

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.

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.

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)*.

A Appendix

This appendix section contains a summary of data statistics, implementation details of the downstream task experiments and the rest of the LINSPECTOR word-level probing analyses.

A.1 Data Statistics

We summarize data statistics of the benchmark used in this work in this section. Table 4 contains statistics for SIB-200 (Adelani et al., 2024; Goyal et al., 2022; NLLB Team et al., 2022) and LINSPECTOR (Şahin et al., 2020), where each language split contains same number examples for training, validation and testing purposes. Table 5 reports the statistics of dependency parsing treebanks used in this work. Lastly, we share the NER benchmarks statistics in Table 6.

A.2 Implementation Details

PIXEL-M4. Table 7 lists the hyperparameter configurations used for pixel language models, PIXEL-M4 and PIXEL-BIGRAMS, across downstream tasks. Overall, we use the same set of hyperparameters with the previous work (Lotz et al., 2023). We repeat the same experiment using different random seeds. For reporting test results, we average the test scores of the five runs with the highest validation split performance.

CANINE-S. We use the same experimental setup and hyperparameters for CANINE-S with PIXEL-M4 and PIXEL-BIGRAMS.

Monolingual BERT Models. All models were fine-tuned in 16-bit BrainFloat (Abadi et al., 2016) using AdamW (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with a maximum learning rate of $5e{-}5$ that is warmed up over the first 100 steps and subsequently linearly decayed toward 0. Across all tasks, we fine-tune for at maximum 15,000 steps, while evaluating every 500 steps for dependency parsing and NER, whereas topic classification is evaluated every epoch. Early stopping of 5 evaluation cycles (DP and NER) or 20 epochs with a threshold of 0.0 is implemented. For all tasks and

Benchmark	License	Train	Validation	Test
SIB-200	CC BY-SA 4.0	701	99	204
LINSPECTOR	Apache 2.0	7000	2000	1000

Table 4: Data statistics for the equally-sized SIB-200 and LINSPECTOR language splits.

languages, when a separate evaluation split is available, we selected the checkpoint performing best on it and evaluated on the test split. If no separate evaluation split was available, we selected and reported the best performance on the evaluation split. Inputs were truncated or padded to a maximum length of 256 tokens for parsing and classification, and 196 tokens for NER. For parsing and NER, a batch size of 64 is used, while topic classification is trained with batch size 32. We followed Rust et al. (2023) and evaluated dependency parsing using a biaffine parsing head (Dozat and Manning, 2017; Glavaš and Vulić, 2021).

A.3 LINSPECTOR Results

In this appendix section, we share the results for the rest of the word-level probing analyses on LINSPECTOR (Şahin et al., 2020). We analyze our model on fifteen languages—Arabic, Armenian, Bulgarian, Dutch, Estonian, Finnish, French, German, Greek, Hungarian, Macedonian, Polish, Russian, Swedish, and Turkish—across fourteen linguistic probing tasks: *Case Marking* (Fig. 6), *Gender* (Fig. 17), *Mood* (Fig. 7), *Number* (Fig. 8), *OddFeat* (Fig. 9), *Person* (Fig. 10), *Polarity* (Fig. 18), *POS* (Fig. 11), *Possession* (Fig. 19), *Pseudo* (Fig. 12), *SameFeat* (Fig. 13), *TagCount* (Fig. 14), *Tense* (Fig. 15), and *Voice* (Fig. 16).

These analyses provide further support for the findings reported in §5. Throughout the entire network, PIXEL-M4 captures more robust linguistic features than PIXEL-BIGRAMS on all tasks for the Cyrillic script languages, Bulgarian, Macedonian and Russian. This is again expected since PIXEL-M4 has seen a similar language, e.g. Ukrainian, during pretraining. Similarly, our observations are the same for the languages in unseen scripts, Arabic, Armenian and Greek, showcasing the improved cross-lingual transfer learning capabilities of PIXEL-M4. Furthermore, on Latin script languages, both models achieve similar overall performances across the layers. Nonetheless, on some tasks, PIXEL-M4 captures better linguistic features for Latin languages with diacritics (e.g. Turkish, Swedish). Additionally, on more complex tasks such as *OddFeat* and *SameFeat*, PIXEL-M4 outperforms PIXEL-BIGRAMS on Latin script languages like German and Hungarian, where the two models perform similarly on the other tasks.

A.4 Use of AI Assistants

Within this work, we used AI assistants only to generate code for producing the plots.

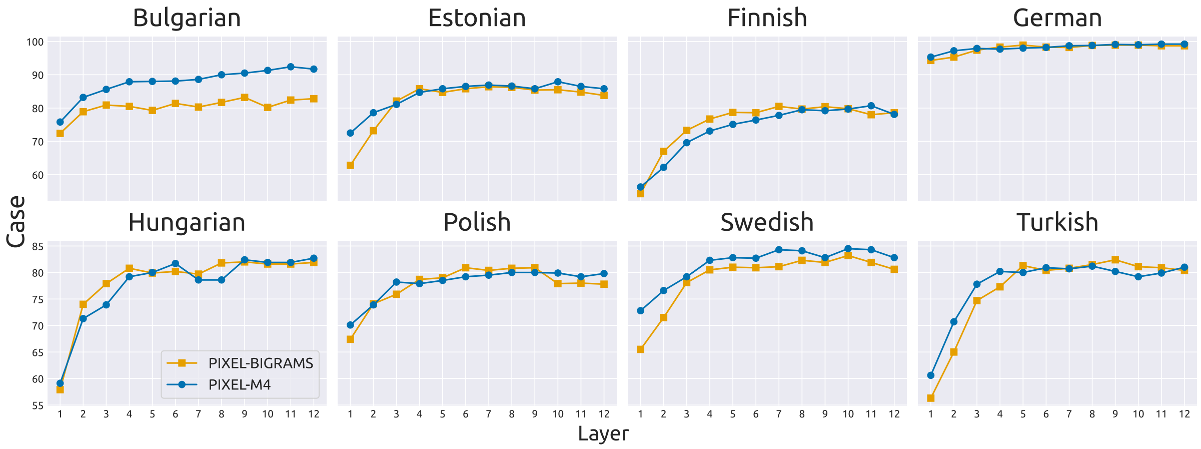

Figure 6: Word-level probing analysis on LINSPECTOR for the Case task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

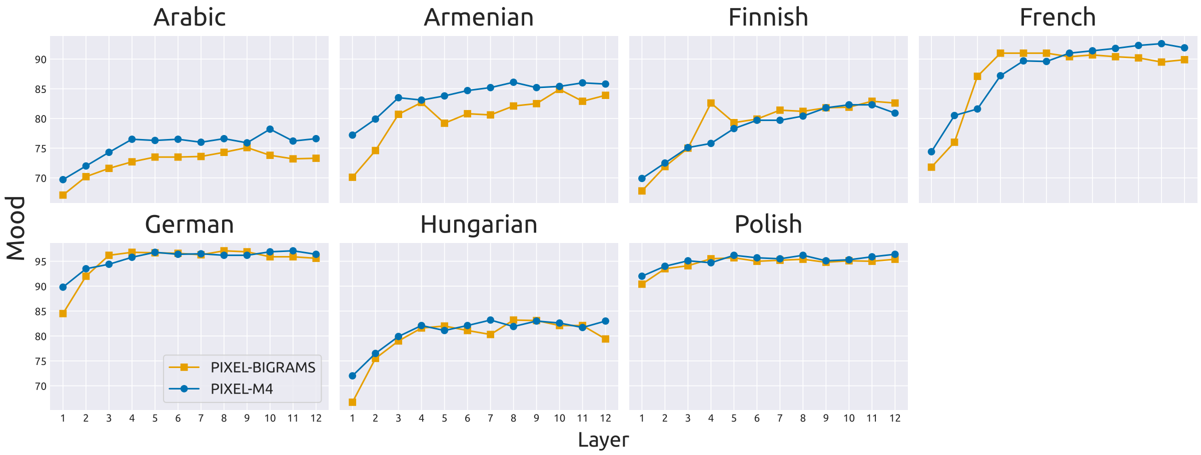

Figure 7: Word-level probing analysis on LINSPECTOR for the Mood task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

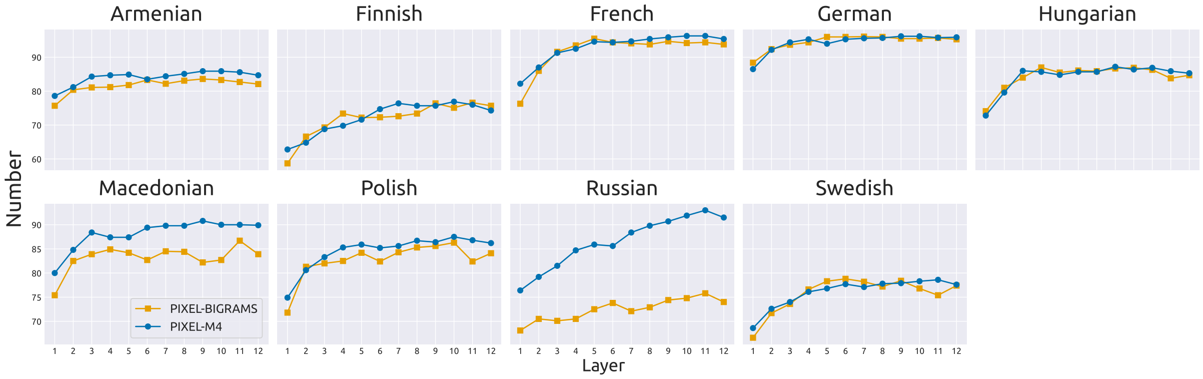

Figure 8: Word-level probing analysis on LINSPECTOR for the Number task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

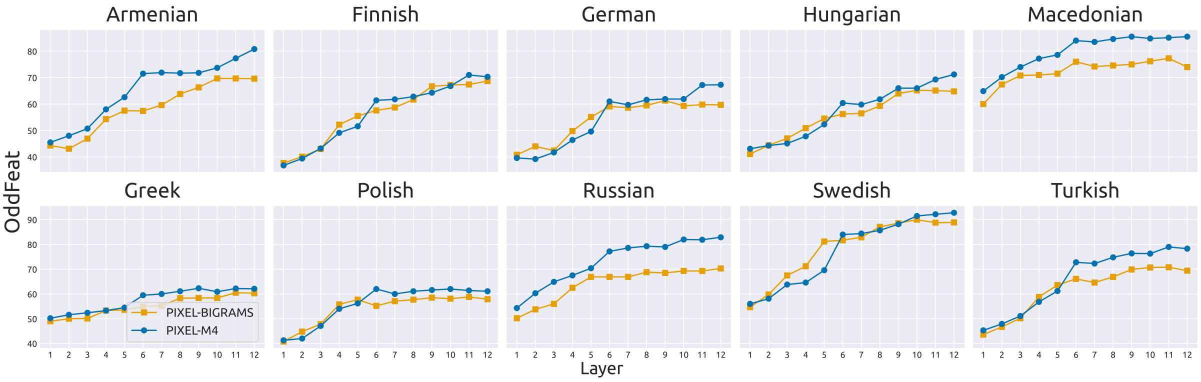

Figure 9: Word-level probing analysis on LINSPECTOR for the OddFeat task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

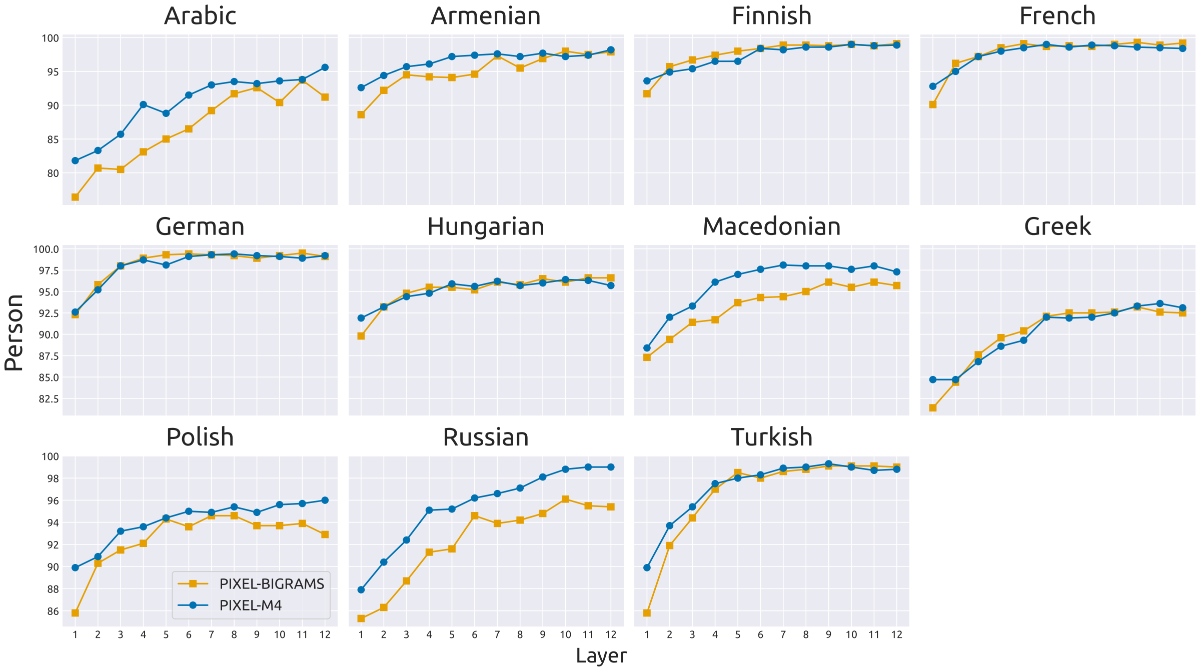

Figure 10: Word-level probing analysis on LINSPECTOR for the Person task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

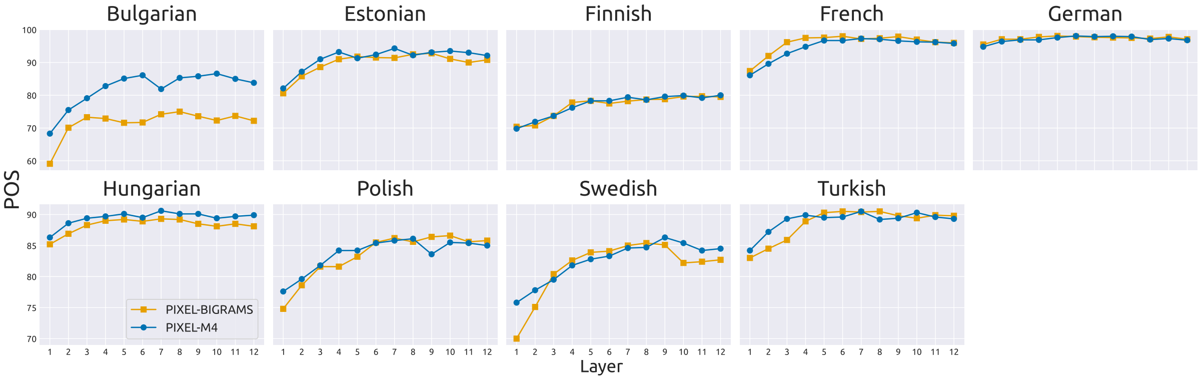

Figure 11: Word-level probing analysis on LINSPECTOR for the Pos task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

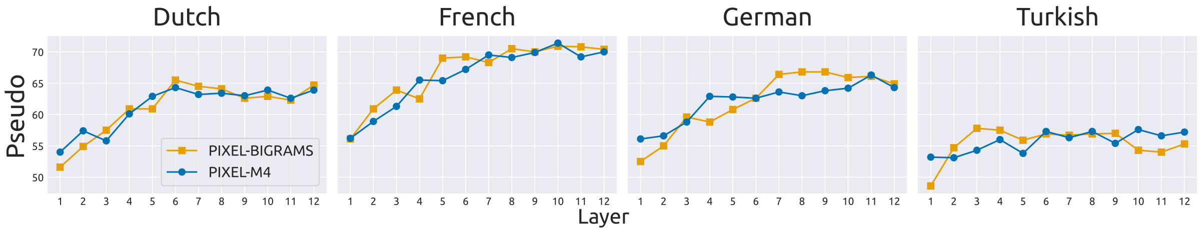

Figure 12: Word-level probing analysis on LINSPECTOR for the Pseudo task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

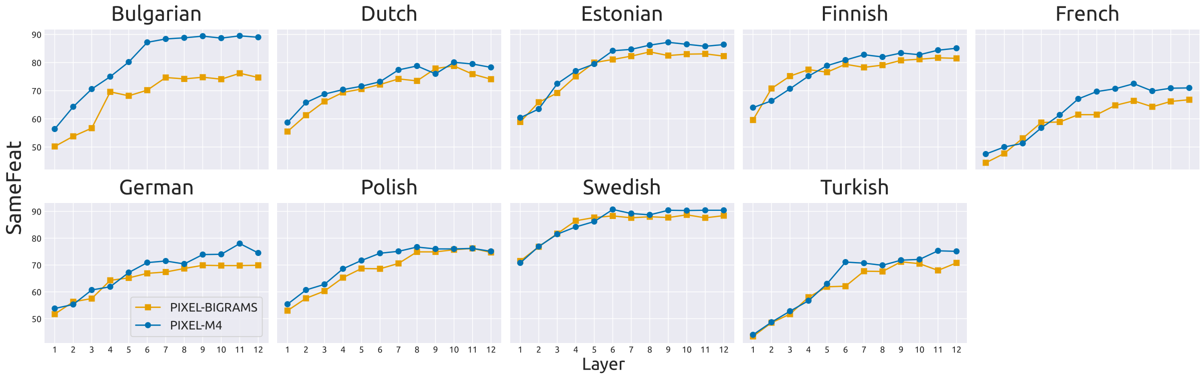

Figure 13: Word-level probing analysis on LINSPECTOR for the SameFeat task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

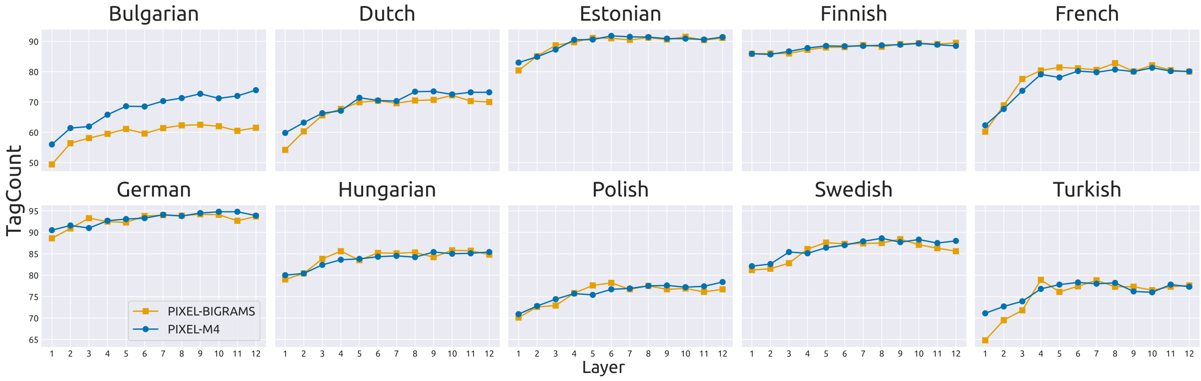

Figure 14: Word-level probing analysis on LINSPECTOR for the TagCount task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

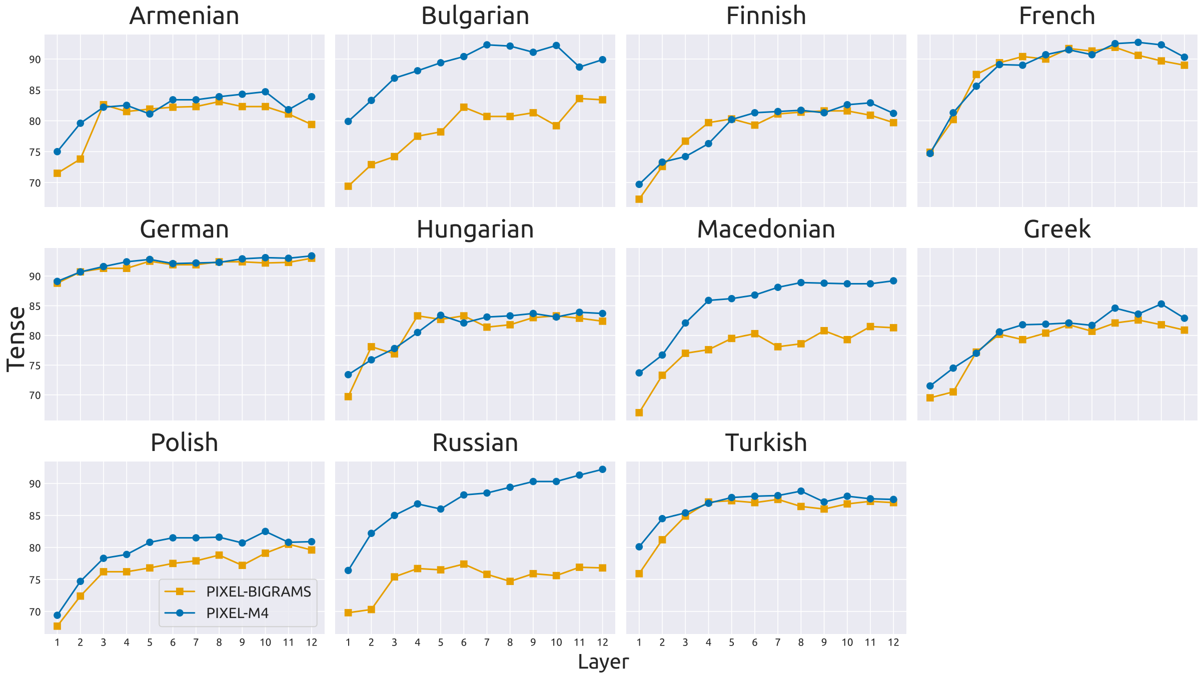

Figure 15: Word-level probing analysis on LINSPECTOR for the Tense task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

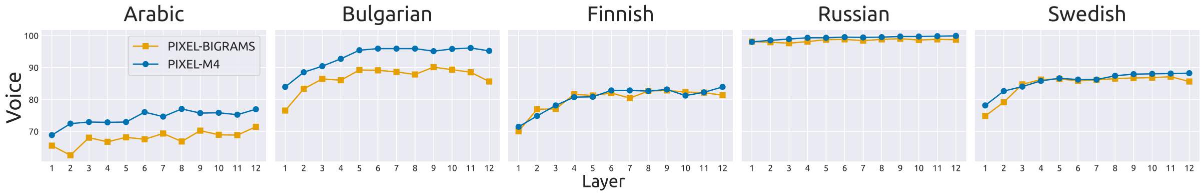

Figure 16: Word-level probing analysis on LINSPECTOR for the Voice task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

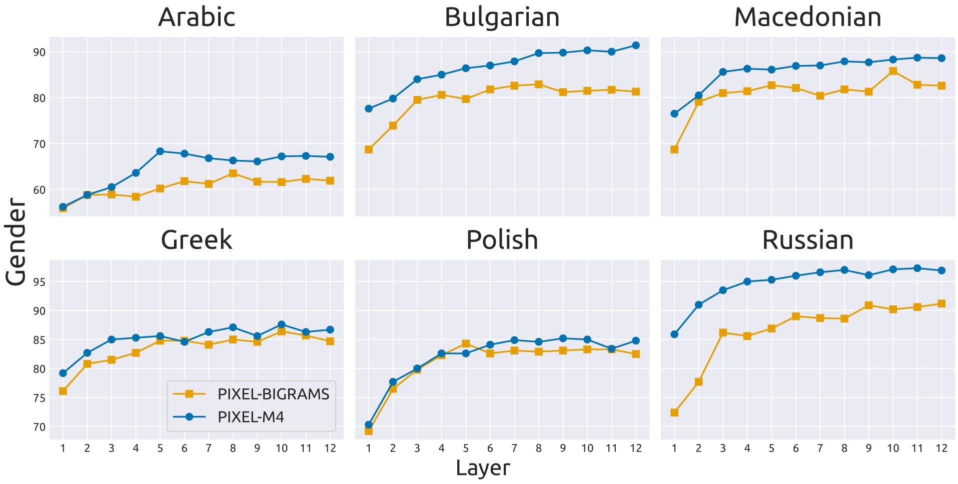

Figure 17: Word-level probing analysis on LINSPECTOR for the Gender task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

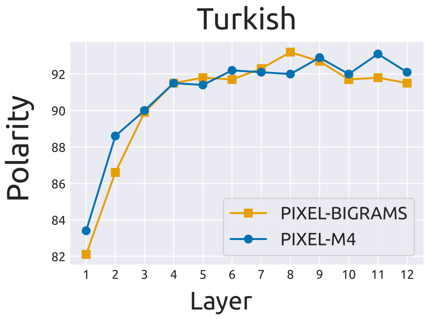

Figure 18: Word-level probing analysis on LINSPECTOR for the Polarity task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

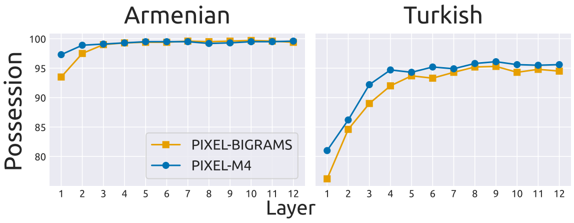

Figure 19: Word-level probing analysis on LINSPECTOR for the Possession task. Each subplot shows a different language; in each, the y-axis represents model accuracies and the x-axis represents layer number of the hidden representations.

Language	Treebank	#Sentences	License
ENG	English-EWT	16621	CC BY-SA 4.0
ARA	Arabic-PADT	7664	CC BY-NC-SA 3.0
BUL	Bulgarian-BTB	11138	CC BY-NC-SA 3.0
COP	Coptic-Scriptorium	2011	CC BY 4.0
HIN	Hindi-HDTB	16647	CC BY-NC-SA 4.0
JPN	Japanese-GSD	8100	CC BY-SA 4.0
KOR	Korean-GSD	6339	CC BY-SA 4.0
RUS	Russian-GSD	5030	CC BY-SA 4.0
TAM	Tamil-TTB	600	CC BY-NC-SA 3.0
TEL	Telugu-MTG	5130	CC BY-SA 4.0
UKR	Ukrainian-IU	5030	CC BY-NC-SA 4.0
URD	Urdu-UDTB	5130	CC BY-NC-SA 4.0
VIE	Vietnamese-VTB	3000	CC BY-SA 4.0
ZHO	Chinese-GSD	4997	CC BY-SA 4.0

Table 5: Total number of sentences of Universal Dependencies v2.10 (Zeman et al., 2022; Nivre et al., 2020) treebanks used for dependency parsing task evaluations, including dataset licenses. Adapted from Rust et al. (2023).

Language	Source	#Sentences	License
ENG	English-EWT	16621	CC BY-SA 4.0
SRP	Serbian-SET	4384	CC BY-SA 4.0
HIN	Naamapadam	1M	CC0
BEN	Naamapadam	967k	CC0
TAM	Naamapadam	501k	CC0
TEL	Naamapadam	511k	CC0
KOR	KLUE	26k	CC BY-SA 4.0
ZHO	Chinese-GSD	4997	CC BY-SA 4.0

Table 6: Overview of NER datasets (Mayhew et al., 2024; Mhaske et al., 2023; Park et al., 2021).

Parameter	SIB-200	UDP	NER
Classification head pooling	Mean	—	—
Optimizer	AdamW		
Adam β	0.9, 0.999		
Adam ε	$1e{-8}$		
Weight decay	0		
Learning rate	$\{1e{-5}, 3e{-5}, 5e{-5}, 7e{-5}, 9e{-5}\}$		
Learning rate schedule	Linear decay		
Warmup steps	100		
Max sequence length	256	256	196
Stride	—	—	—
Batch size	32	64	64
Max steps	15 000	15 000	15 000
Eval strategy	epochs	steps	steps
Eval steps	—	500	500
Early stopping	✓		
Early stopping patience	20	5	5
Dropout probability	0.1		

Table 7: Hyperparameters used for fine-tuning and evaluating models on the SIB-200, UDP parsing, and NER tasks.