Kenji-Endo: a BabyLM @EVALITA

Calogero Jerik Scozzaro[1], Mattero Rinaldi[1], Gianluca Mittone[1] and Marco Antonio Stranisci[1,2]

[1]University of Torino, Computer Science Department, Corso Svizzera 185, 10149 Torino, Italy
[2]aequa-tech, Turin, Italy

Abstract

We present Kenji-Endo, a BabyLM pretrained on a dedicated Italian dataset, which participated in four tasks at the 9th edition of EVALITA: DeSegMa, MultiPRIDE, IMPOLS, and FadeIT. Kenji-Endo achieved competitive performance across all tasks, demonstrating that language modeling with limited data and compact model sizes can represent a viable alternative to Large Language Models.

Keywords

NLP, Evaluation, Italian, Language Models, BabyLM

Katsumata: Senti, diventeresti mio amico?
Kenji-Endo: Mi va anche bene, però diventare amici non è una cosa che si decide, sai?

Naoki Urasawa

1. Introduction

Large Language Models (LLMs) have become a dominant paradigm in Natural Language Processing and their adoption has gone beyond the academic field, supporting a high number of real-life applications. For a long time, the development of these models followed the scaling law principle [1], according to which bigger models and datasets would have led to better performance. LLMs dramatically increased in the number of parameters: BERT [2] and GPT-2 [3], both released in 2019, counted 110M and 124M parameters; Llama4 Behemoth, released in 2025, 2T parameters [4]. This growth has also been facilitated by architectural innovations such as the Mixture-of-Experts (MoE) paradigm [5, 6]. In MoE models, the standard feed-forward layers are replaced by multiple parallel "expert" networks, and a routing mechanism dynamically selects a small subset of them for each input token. As a result, only a fraction of the total parameters is activated at each forward pass, allowing the model to scale to very large parameter counts while keeping computational costs relatively limited. The same happened for datasets' size, which scaled from billions [7] to trillions of tokens [8].

Alongside this significant growth, new approaches are emerging as critical alternatives focused on environmental impact [9] and more cognitively inspired architectures [10]. Among these approaches, the BabyLM challenge [11, 12, 13] is emerging as a novel line of research in sample-efficient language modeling. The initiative calls for small architectures that aim to simulate developmental learning processes by limiting the amount of pretraining data to developmentally plausible quantities, typically enforcing budgets of 10M and 100M words of text-only data. In addition to data constraints, the challenge also encourages limited training durations, further emphasizing efficiency in learning. Initially conceived as an English-based project, BabyLM models have been proposed for Italian [14], bilingual [15], and multilingual [16] settings.

In this paper we present Kenji-Endo: a BabyLM system presented at the 9th edition of EVALITA [17]. Whilst inspired by BabyLM, the focus on Kenji-Endo is not on developmental learning but on testing

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT
calogerojerik.scozzaro@unito.it (C. J. Scozzaro); matteo.rinaldi@unito.it (M. Rinaldi); gianluca.mittone@unito.it (G. Mittone); marcoantonio.stranisci@unito.it (M. A. Stranisci)
0009-0003-8311-7801 (C. J. Scozzaro); 0000-0001-9457-6243 (M. Rinaldi); 0000-0001-9457-6243 (G. Mittone); 0000-0001-9457-6243 (M. A. Stranisci)

Group	Tokens
Books	18.1M
Conversations	3.4M
Forums	13.2M
Laws	5.5M
News	18.4M
Social Media	273K
Subtitles	4.3M
Wikipedia	13.8M

Table 1
Composition of the Kenji-Endo corpus by text group, source, and total number of characters.

the ability of a small architecture against larger LLMs. To this extent, we developed the Kenji-Endo dataset, a corpus of approximately 135 million tokens collected from different sources (e.g., legal texts, subtitles) and filtered using a text readability criterion. This dataset was used to pretrain Kenji-Endo from scratch. We adopted three pretraining settings: two standard dense transformer architecture and a Mixture-of-Experts (MoE) variant.

Kenji Endo has been tested on the following EVALITA tasks: DeSegMa-IT [18], MultiPRIDE [19], IMPOLS, and FadeIT [20]. Results obtained by our system show that Kenji-Endo achieves competitive performance across multiple EVALITA tasks, including the second-best accuracy on DeSegMa-IT Subtask A and the best score on IMPOLS Subtask 1, despite its small size and limited pretraining computation. However, performance degrades in settings that require the exploitation of noisy or highly abstract contextual information, highlighting limitations in handling complex pragmatic phenomena. Despite these limitations, Kenji-Endo demonstrates that small models can remain effective and competitive when carefully pretrained and adapted to downstream tasks.[1]

2. System Description

In this section, we describe the architecture and training pipeline of the proposed system. We first introduce the Kenji-Endo corpus developed for pretraining, detailing its composition and data selection criteria. We then describe the pretraining setup and the resulting pretrained models, followed by the fine-tuning and prompt-based strategies employed to adapt the models to the downstream classification tasks.

2.1. The Kenji-Endo Corpus

In order to pretrain our system, we developed the Kenji-Endo corpus, a dedicated dataset composed of documents selected from different sources and belonging to different text types. Particular attention was paid to choosing datasets that may be small in terms of token count but capable of providing a diverse and peculiar data mixture. Given that the object of the work is a BabyLM, CHILDEES, a corpus focused on children's language, was included.[2] We also included Kiparla [21] and VoLip [22] as examples of spoken text transcriptions, as well as [23] and [24], two corpora derived from tweets. Parts of Eurlex and Gazzetta Ufficiale [25] were also incorporated. We included a dump of articles from DueParole,[3] given its focus on easy-to-read Italian. News, blog posts and discussion board posts were obtained from TestiMole [26], alongside public domain books sourced mainly from LiberLiber[4] and the clean Wikipedia dump contained in TestiMole. A synthetic description of the Kenji-Endo corpus is reported in Table 1.

[1]Code and data of our system are available at https://github.com/neurosymbolic-unito/kenji-endo-at-evalita
[2]The corpus was obtained from the Italian language section of https://talkbank.org/childes/access/Romance/
[3]Articles accessible at http://dueparole.it
[4]https://liberliber.it/, included in TestiMole

An additional criterion adopted in the construction of the Kenji-Endo dataset was a filtering process based on text readability. To this extent, we exploited *Il Nuovo Vocabolario di Base della Lingua Italiana*[5], a dictionary of high-frequency Italian words developed by Tullio De Mauro. We used the vocabulary to classify documents according to their readability by computing the relative frequency of high-occurring terms within each document. For each document type, we computed the third-quartile threshold of this distribution and retained only documents whose values exceeded this threshold, corresponding to texts with a higher presence of high-frequency terms. The final corpus is composed of 184,000 documents and 83.80 millions of tokens.

2.2. Pre-Training Setup

The BabyLM pre-training is run on a single NVIDIA Grace Hopper Superchip (1x Grace A02 CPU 72 cores + 1x NVIDIA H100 GPU) on the HPC4AI research cluster hosted by the University of Turin, Italy [27]. The training code is based on PyTorch exploiting the xFFL framework, which allows high-performance, effortless deployment of AI training on distributed infrastructures [28]. From a technical perspective, the whole computation exploits the half-precision bfloat16 data format to reduce GPU RAM impact, halving the memory required to store the model's parameters, activations, and gradients, while improving the overall computational performance, since the GH100 can process such low-precision data more than one order of magnitude faster than standard FP32 [29]. Activation checkpointing is also adopted to save additional GPU RAM and thus allowing even larger batch sizes, which empirically provide better learning convergence; the final training batch size used is 64 samples. AdamW is used as optimizer, as common in contemporary practice, with a starting learning rate of $2e^{-4}$, weight decay of 0.1, and betas of $(0.9, 0.95)$. A warmup-cosine-decay learning rate scheduler is adopted to obtain improved and more stable learning convergence, using 1% of the available batches for the warmup phase, and then decreasing the learning rate to 1% of its initial value over a single training epoch (1,272 batches in total). During pre-training, we used a maximum context window of 1024 tokens and trained the models for a total of 10 epochs. Gradient clipping is set to 1.0, and no further gradient accumulation is used.

This pre-training procedure resulted in two pretrained language models, differing in their architectural design:

- Kenji-Endo vanilla[6]: a dense transformer-based causal language model with 12 layers and a hidden size of 768, trained using full causal self-attention at every layer and based on the `Qwen3ForCausalLM` architecture. This model follows a standard decoder-only architecture and has approximately 130M parameters.
- Kenji-Endo MoE[7]: a Mixture-of-Experts (MoE) variant with the same number of layers and hidden dimensionality as the vanilla model, in which the standard feed-forward blocks are replaced by sparse expert routing, based on the `Qwen2MoeForCausalLM` architecture. At each token, a single expert is selected among 7 available experts, together with one shared expert that is always active, enabling increased model capacity while maintaining a comparable computational cost, for a total of approximately 240M parameters.

Additionally, we pretrained a further model variant employing more recent architectural features: Kenji-Endo AliBi.[8] This model was trained using the Matformer[9] framework instead of the Hugging Face `Trainer`. Matformer is a new in-house developed library that focuses on better modularity and code readability, while providing readily available support for different configurations and optimization strategies. Kenji-Endo AliBi was trained for 8 hours on a single Nvidia 3090 GPU connected to a Thinkpad R61 laptop, an extremely economical setting that highlights the low access barrier needed for

[5] https://github.com/marcostranisci/babyLM/blob/main/resources/nvdb.csv
[6] https://huggingface.co/CCC-Unito/kenji-endo-1.0-vanilla
[7] https://huggingface.co/CCC-Unito/kenji-endo-1.0-moe
[8] https://huggingface.co/CCC-Unito/kenji-endo-1.0
[9] https://github.com/mrinaldi97/matformer

developing BabyLMs. It consists of 12 layers, a hidden size of 768, a SwiGLU feed-forward network with an intermediate size set to three times the hidden size, and uses RMSNorm for layer normalization. The model was trained for 10 epochs with a batch size of 16 and a gradient accumulation of 12, adopting a warmup-hold-cosine decay lr scheduling that peaked at $1e-3$ and decayed until $8e-5$. As a positional embedding strategy, AliBi [30] was used instead of RoPe [31] to enable generalization and scaling beyond the 1024 token sequence length employed during training. Flash attention 2, Liger Kernels and Nvidia's transformer-engine are natively supported by the Matformer library and were used to speed up training. Finally, Muon [32] was used instead of AdamW as optimizer, with a Nesterov-style momentum of 0.95 and 5 Newton–Schulz iteration steps. Compared to AdamW, Muon allows for faster convergence, an aspect that may be crucial in the development of Baby-LMs. The final model achieved performance comparable to that of Kenji-Endo vanilla; however we reserve for the future to conduct more tests especially to compare the performances in long context scenario.

2.3. Fine-Tuning and Prompt-Tuning Setup

The pretrained models are trained with next-token prediction objective. However, the tasks addressed in this work are framed as classification problems, which require an additional adaptation step in order to map the model outputs to discrete class labels. To this end, we explored two classification strategies on the Kenji-Endo vanilla and Kenji-Endo MoE models: a discriminative fine-tuning approach and a generative, prompt-based approach. In the discriminative setting, we fine-tuned the models using a batch size of 8 and the `AutoModelForSequenceClassification`[10] framework, which connects the pretrained model to a classification head. The classification head consists of a linear layer applied to the hidden representation of the last token of the input sentence, producing class logits. The model is trained by directly optimizing a cross-entropy loss over the class labels. All model parameters were updated during fine-tuning. This configuration was applied only to the vanilla model, as the MoE architecture employed does not support this class. Training was carried out for 3 epochs on a single NVIDIA T4 GPU. In the generative setting, no fine-tuning was performed. Instead, we adopted a prompt-based approach applicable to both vanilla and MoE models, casting the classification task as text generation. We used a batch size of 4 and a simple prompt that includes the input text followed by a class label request (e.g., *"Testo: {text} Classe:"*).

For all experiments, we fixed the maximum sequence length to 1024 tokens, including tasks in which a portion of the instances exceeded this length. For each task, we split the original training set into 80% training and 20% validation data. This split was used to fine-tune and select the best checkpoint in the discriminative setting. In the generative setting, no parameter updates were performed, as the models were evaluated in a zero-shot fashion. The validation split was used solely for configuration comparison and model selection. During fine-tuning, model checkpoints were saved at each epoch, and the checkpoint achieving the best performance on the validation set was selected. Unless otherwise specified, we relied on the default hyperparameter settings provided by the Hugging Face `Trainer`.[11] Fine-tuning was carried out on NVIDIA A40 GPUs available on the HPC4AI infrastructure. The compute time of each model has been 1110.70 seconds on average.

3. Results Overview

In this section, we present an overview of the tasks in which our team participated and report the corresponding experimental results. For each task, we provide a brief description and compare our performance against other participating systems using the official evaluation metrics. Table 2 summarizes the final configurations adopted for each subtask.

[10] https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#automodelforsequenceclassification
[11] https://huggingface.co/docs/transformers/main_classes/trainer

Task	Subtask	Strategy	Model
DeSegMa	A	Discriminative	Kenji-Endo vanilla
MultiPRIDE	A	Discriminative	Kenji-Endo vanilla
MultiPRIDE	B	Discriminative	Kenji-Endo vanilla
IMPOLS	1	Discriminative	Kenji-Endo vanilla
IMPOLS	2	Generative	Kenji-Endo vanilla
FadeIT	A	Discriminative	Kenji-Endo AliBi

Table 2
Overview of the tasks and subtasks in which Kenji-Endo participated, including the strategy and model variant used for the final submission.

Position	Team	Accuracy
1	Gradient Descenders	0.9458
2	**Kenji Endo**	**0.9426**
3	UniTor	0.9288
4	Nicla	0.9243
5	Stochastic Gradient Descenders	0.9216

Table 3
Results of DeSegMa-IT Subtask A ranked by accuracy.

3.1. DeSegMa

DeSegMa-IT (Detection and Segmentation of Machine Generated Text in Italian) [18] focuses on the detection of machine-generated text in Italian. It is composed of two subtasks: Subtask A, where given an input text, systems are required to perform binary classification, distinguishing between human-written and machine-generated content; and Subtask B, where participants are required to detect the boundary between human-written text and its machine-generated continuation by identifying the character index marking the beginning of the machine-generated content. Our participation was limited to Subtask A.

Due to the large size of the DeSegMa-IT training set, under the discriminative strategy we explored two training configurations: *i)* training for a single epoch on the full dataset, and *ii)* training for three epochs on a reduced subset of 3,000 instances. Overall, we evaluated four experimental configurations: *i)* vanilla discriminative training on the full dataset, *ii)* vanilla discriminative training for three epochs on the reduced subset, *iii)* vanilla generative inference, and *iv)* MoE generative inference. Based on validation performance, the vanilla discriminative model trained for a single epoch on the full dataset achieved the best results (Accuracy: 0.9952, F1-score: 0.9952) and was therefore selected for the final submission.

Table 3 reports the official results on the DeSegMa-IT Subtask A test set. Our system achieved the second-best performance among the participating teams, with an accuracy of 0.9426 and, ranking closely behind the top-performing approach. A closer inspection of the classification results shows a generally balanced performance across the two classes. Overall precision, recall, and F1-score are comparable (Precision: 0.9455, Recall: 0.9426, F1-score: 0.9426), with only minor differences in class-wise metrics. For the human-written class, recall is higher (0.9824) than precision (0.91), while for the machine-generated class, precision (0.98) is higher than recall (0.9031). This behavior shows a cautious tendency in assigning the machine-generated label, characterized by higher precision than recall. For completeness, we conducted a full analysis by evaluating the performance of all four experimental configurations. The submitted configuration achieved the best overall performance (F1-score: 0.9426). The discriminative configuration trained for three epochs on the reduced subset showed slightly lower results (F1-score: 0.9207), suggesting that exposure to the full training data was more beneficial than additional epochs on a smaller subset. The generative approaches, using the vanilla and MoE models, achieved competitive but lower scores (F1-scores of 0.9207 and 0.9225, respectively), indicating that the prompt-based formulation did not match the performance of full discriminative fine-tuning on this task.

3.2. MultiPRIDE

MultiPRIDE is a binary classification task in which systems are required to determine whether a term related to an LGBTQ+ context in a sentence is used with a reclamatory intent. The task is structured into two subtasks: Subtask A, where only the textual content of the tweet is provided; and Subtask B, where, in addition to the tweet content, participants have access to optional contextual information related to the author's profile, such as their biography, when available. The task is proposed in a multilingual setting, encompassing data in Italian, Spanish, and English. Our team participated in Subtasks A and B for the Italian language.

For each subtask, we evaluated three experimental configurations, corresponding to the vanilla discriminative setting and the two generative inference setups described in Section 2.3. In both cases, the discriminative Kenji-Endo vanilla model achieved the best performance on the development set and was therefore selected for the final submission. Specifically, for Subtask A, the selected model obtained a development set Accuracy of 0.9174 and an F1-score of 0.7429, while for Subtask B it achieved a development set Accuracy of 0.8394 and an F1-score of 0.6316.

Table 4 reports the official rankings for Subtasks A and B on the Italian language, evaluated in terms of macro F1-score. Our system achieved a macro F1-score of 0.8360 in Subtask A and 0.7489 in Subtask B, placing in the mid-to-lower range of participating systems.

Analyzing results, it is possible to observe that the model's performance is skewed towards precision in the Subtask A while the opposite happens for Subtask B. Besides being detrimental in the overall performance, the additional context appears to produce a greater number of false positive (+0.13) than false negatives (+0.2). To further investigate this behavior, we analyzed the tokenization characteristics of the inputs. We observed that the biography field contains a substantially higher proportion of byte-fallback tokens (e.g., emojis and flag symbols) than the tweet text. Specifically, the mean byte-fallback ratio is 0.1149 for biographies compared to 0.0134 for tweet text, and 34.3% of biography entries contain at least one such token (versus 9.9% for tweet text). This suggests that the additional contextual information in Subtask B may introduce noisy and semantically fragmented token sequences, which can dilute the useful signal and provide a plausible explanation for the lower F1-score despite the richer input.

3.3. IMPOLS

IMPOLS focuses on the automatic recognition of implicit content in Italian political speech. Given an utterance within its context, systems are required to detect and classify implicit, non-bona fide contents, i.e., questionable information that is not explicitly stated but is implicitly conveyed and understood as true. Such content is commonly used in political communication to convey potentially manipulative messages implicitly. The task is structured into three subtasks: 1) a binary detection task aimed at identifying the presence of questionable implicit content; 2) a binary classification task distinguishing between implicatures and presuppositions; and 3) a multiclass classification task focusing on the categorization of implicatures into particularized conversational, generalized conversational, or conventional. Our team participated in Subtasks 1 and 2. For each subtask, we evaluated multiple experimental configurations, similarly to the previous tasks. For Subtask 1, the discriminative Kenji-Endo vanilla model achieved the best performance on the development set and was therefore selected for the final submission. Specifically, the selected model obtained a development set accuracy of 0.9507 and an F1-score of 0.9521. For Subtask 2, the generative Kenji-Endo vanilla model achieved a development set accuracy of 0.5027 and an F1-score of 0.4417.

Table 5 reports the official results obtained on the IMPOLS test set. Our system achieved the best performance in Subtask 1, obtaining an F1-score of 0.9496 and ranking first among the participating teams. From a closer look at model's performance two main aspects emerge: *i.* Most of the errors are false negatives (63%), showing that the model lacks of recall; *ii.* The model struggles with discourses from a specific speaker - Pierluigi Bersani - whose speech excerpts represent 29.2% of all the wrong prediction. The second speaker with the highest number of wrong predictions is Mario Draghi (4, 9.7%).

Pos.	Team (Run)	F1
1	Ghavidel-Rajabi (1)	0.8981
2	MilaNLP (1)	0.8959
3	Ghavidel-Rajabi (2)	0.8909
4	SaFe Tweets (1)	0.8895
5	LlaNa (1)	0.8835
6	GRUPPETTOZZO (1)	0.8834
7	Challenger (1)	0.8816
8	HateItOff (1)	0.8809
9	GRUPPETTOZZO (2)	0.8735
10	baseline (1)	0.8731
11	UniBO-FICLIT (1)	0.8707
12	NamDang (2)	0.8589
13	HateItOff (2)	0.8584
14	NetGuardAI (1)	0.8537
15	The Hate Busters (2)	0.8503
16	I2C (2)	0.8435
17	NamDang (1)	0.8407
18	**Kenji-Endo (1)**	**0.8360**
19	The Hate Busters (1)	0.8345
20	MilaNLP (2)	0.8306
21	KIT-TIP-NLP (2)	0.8103
22	Avahi (1)	0.7904
23	KIT-TIP-NLP (1)	0.7659
24	I2C (1)	0.6202

(a) Subtask A

Pos.	Team (Run)	F1
1	LlaNa (1)	0.9021
2	baseline (1)	0.8981
3	GRUPPETTOZZO (1)	0.8979
4	The Hate Busters (2)	0.8827
5	MilaNLP (1)	0.8827
6	GRUPPETTOZZO (2)	0.8681
7	UniBO-FICLIT (1)	0.8644
8	MilaNLP (2)	0.8641
9	AIWizards (1)	0.8564
10	AIWizards (2)	0.8549
11	HateItOff (2)	0.8462
12	The Hate Busters (1)	0.8324
13	HateItOff (1)	0.8319
14	SaFe Tweets (1)	0.8164
15	Avahi (1)	0.8031
16	DataSummit (1)	0.7734
17	**Kenji-Endo (1)**	**0.7489**
18	NetGuardAI (1)	0.7405

(b) Subtask B

Table 4
Official rankings for MultiPRIDE (Italian) in terms of macro F1-score for Subtasks A (left) and B (right). The number in parentheses indicates the submitted run identifier.

Team	Subtask 1	Subtask 2	Subtask 3
kenji-endo	**0.9496**	**0.2226**	–
RES2	0.9089	0.8762	0.6369
Baseline	0.4602	0.3352	0.2942

Table 5
Results on the IMPOLS task in terms of F1-score across the three subtasks.

Performance on Subtask 2 was substantially lower (F1-score of 0.2226). An inspection of the model predictions revealed a degenerate behavior, with the model predicting a single class, which negatively impacted the overall performance on this subtask. The drop of performance might suggest the lack of model's ability to distinguish between more abstract concepts related to pragmatics, due to the limited amount of pretraining data that does not include discipline-specific knowledge.

3.4. FadeIT

FadeIT is a task focused on the detection of fallacies expressed in Italian social media texts and is based on the FAINA dataset [33]. AINA consists of Italian social media posts manually annotated by expert annotators for the presence of argumentative fallacies, with over 11,000 span-level annotations spanning 20 different fallacy types (e.g., ad hominem, strawman). The criterion for selecting the texts was their belonging to three specific categories: migration, climate change and public health. Each text can be classified as expressing zero, one, or multiple fallacy types. The annotations also account for human label variation, highlighting the intrinsic subjectivity of fallacy identification. The task is structured into two subtasks: Subtask A, "post-level fallacy detection", is framed as a multi-label classification tasks in which the model should correctly detect the fallacies expressed in a given text.

Position	Team (Run)	P	R	F1
1	TiGRO (3)	53.24	59.99	56.39
2	MALTO (2)	55.75	53.60	54.63
3	TiGRO (2)	53.43	51.48	52.41
4	UNICA (1)	55.22	45.23	49.71
5	TiGRO (1)	62.52	39.85	48.65
6	UNICA (3)	51.19	44.26	47.45
7	**Kenji-Endo (1)**	**49.38**	**45.55**	**47.37**
8	UNICA (2)	58.70	38.44	46.44
9	MALTO (1)	46.56	43.65	45.05
10	RBG-AI (2)	33.09	57.78	42.07
11	RBG-AI (3)	30.65	57.62	40.00
12	Label (3)	27.60	68.08	39.26
13	Label (1)	52.82	30.94	38.96
14	RBG-AI (1)	36.35	41.11	38.57
15	Label (2)	52.76	30.11	38.32
16	Baseline	38.53	14.28	20.84

Table 6
Test set results for FadeIT Subtask A. Systems are ranked by decreasing micro F1-score. The number in parentheses indicates the submitted run identifier.

Subtask B, "fine-grained fallacy detection", instead, is framed as a multilabel span classification task in which the goal is do detect the specific text segments in which the fallacy is expressed. Our team participated in subtask A. For this purpose, the decoder head of the autoregressive model was replaced with a linear classification head, taking as input the output of a mean pooling applied to the models' last layer. To stabilize training and avoid overfitting, we set dropout at 0.5.

Results of Table 6 show that our system ranked 7 out of 15 runs (16 including the baseline) with a micro F1-score of 47.37, with a precision of 49.38 and a recall of 45.55. Compared to the baseline (F1: 20.84), this corresponds to a substantial improvement of +26.53 points. The finer-grained analysis of results shows important limitations in our system, though. Kenji-Endo was not able to identify 14 fallacy-types out of 19 confirming the issues encountered in other tasks where models have to deal with more complex phenomena.

4. Conclusion and Future Work

In this paper, we presented Kenji-Endo, a BabyLM system that participated in four EVALITA tasks: DeSegMa, MultiPRIDE, IMPOLS, and FadeIT. The system was pretrained from scratch on a dedicated corpus using two architectures (simple auto-regressive and Mixture of Experts). Across tasks, Kenji-Endo achieved competitive results, ranking second in DeSegMa Subtask A and first in IMPOLS Subtask 1, while obtaining solid mid-range performance in MultiPRIDE and FadeIT. However, it systematically struggles with context-dependent tasks, as it fails to distinguish pragmatics concepts (implicatures *versus* presupposition) and to handle noisy autobiographical signals such as emojis. Despite the existing limitation, Kenji-Endo demonstrates that small models can remain effective and competitive when carefully pretrained and adapted to downstream tasks. Future work will proceed along two main directions. We will investigate methods to improve performance on more complex and abstract linguistic phenomena, such as the detection of cultural and intertextual references, which remain challenging for models at this scale. We will systematically study the interaction between data recipes and model learning by conducting extensive ablation experiments, with the goal of disentangling the effects of training procedures from those of data composition on downstream task performance.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (OpenAI) and Grammarly in order to: Improve writing style and Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, arXiv preprint arXiv:2001.08361 (2020).

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.

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

[4] A. Meta, The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, https://ai. meta. com/blog/llama-4-multimodal-intelligence/, checked on 4 (2025) 2025.

[5] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, J. Dean, Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, in: International Conference on Learning Representations, 2017.

[6] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, et al., Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, arXiv preprint arXiv:2401.06066 (2024).

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.

[8] L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, et al., Dolma: An open corpus of three trillion tokens for language model pretraining research, in: Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: long papers), 2024, pp. 15725–15788.

[9] M. C. Rillig, M. Ågerstrand, M. Bi, K. A. Gould, U. Sauerland, Risks and benefits of large language models for the environment, Environmental science & technology 57 (2023) 3464–3466.

[10] I. Lee, T. Berg-Kirkpatrick, Readability≠ learnability: Rethinking the role of simplicity in training small language models, in: Second Conference on Language Modeling, ????

[11] A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, et al., Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora, in: Proceedings of the BabyLM challenge at the 27th conference on computational natural language learning, 2023, pp. 1–34.

[12] M. Y. Hu, A. Mueller, C. Ross, A. Williams, T. Linzen, C. Zhuang, R. Cotterell, L. Choshen, A. Warstadt, E. G. Wilcox, Findings of the second babylm challenge: Sample-efficient pretraining on developmentally plausible corpora, in: The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, 2024, p. 1.

[13] L. G. G. Charpentier, L. Choshen, R. Cotterell, M. O. Gul, M. Y. Hu, J. Liu, J. Jumelet, T. Linzen, A. Mueller, C. Ross, et al., Proceedings of the first babylm workshop, in: Proceedings of the First BabyLM Workshop, 2025.

[14] L. Capone, A. Suozzi, G. E. Lebani, A. Lenci, Bambi goes to school: Evaluating italian babylms with invalsi-ita (2025).

[15] Z. Shen, A. Joshi, R.-C. Chen, Bambino-lm:(bilingual-) human-inspired continual pre-training of

babylm, in: Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, 2024, pp. 1–7.
[16] J. Jumelet, A. Fourtassi, A. Haga, B. Bunzeck, B. Shandilya, D. Galvan-Sosa, F. G. Haznitrama, F. Padovani, F. Meyer, H. Hu, et al., Babybabellm: A multilingual benchmark of developmentally plausible training data, arXiv preprint arXiv:2510.10159 (2025).
[17] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
[18] G. Puccetti, A. Pedrotti, A. Esuli, Desegma-it at evalita 2026: Overview of the detection and segmentation of machine generated text in italian task, 2026.
[19] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
[20] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
[21] E. Goria, C. Mauri, Il corpus KIParla: una nuova risorsa per lo studio dell'italiano parlato, volume 2, CLUB- Circolo linguistico dell'Università di Bologna, Bologna, 2018, pp. 96–116. URL: https://hdl.handle.net/2318/1689340.
[22] M. Voghera, C. Iacobini, F. Cutugno, R. Savy, I. Alfano, A. De Rosa, Il VoLIP: una risorsa per lo studio della variazione nel parlato della lingua italiana, in: Actes du XXVIIe Congrès international de linguistique et de philologie romanes, ELiPhi, Strasbourg, 2016, pp. 1651–1663.
[23] M. A. Stranisci, S. Frenda, M. Lai, O. Araque, A. T. Cignarella, V. Basile, C. Bosco, V. Patti, O-dang! the ontology of dangerous speech messages, in: I. Kernerman, S. Carvalho, C. A. Iglesias, R. Sprugnoli (Eds.), Proceedings of the 2nd Workshop on Sentiment Analysis and Linguistic Linked Data, European Language Resources Association, Marseille, France, 2022, pp. 2–8. URL: https://aclanthology.org/2022.salld-1.2/.
[24] A. Ramponi, C. Casula, DiatopIt: A corpus of social media posts for the study of diatopic language variation in Italy, in: Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023), Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 187–199. URL: https://aclanthology.org/2023.vardial-1.19.
[25] E. Federici, M. Ferraretto, N. Landro, Gazzetta Ufficiale: A dataset of legislative texts, public and private acts, 2024. URL: https://huggingface.co/datasets/mii-llm/gazzetta-ufficiale.
[26] M. Rinaldi, R. Varvara, V. Patti, Testimole-conversational: A 30-billion-word italian discussion board corpus (1996-2024) for language modeling and sociolinguistic research, 2026. URL: https://arxiv.org/abs/2602.14819. arXiv:2602.14819.
[27] M. Aldinucci, S. Rabellino, M. Pironti, F. Spiga, P. Viviani, M. Drocco, M. Guerzoni, G. Boella, M. Mellia, P. Margara, I. Drago, R. Marturano, G. Marchetto, E. Piccolo, S. Bagnasco, S. Lusso, S. Vallero, G. Attardi, A. Barchiesi, A. Colla, F. Galeazzi, HPC4AI: an ai-on-demand federated platform endeavour, in: D. R. Kaeli, M. Pericàs (Eds.), Proceedings of the 15th ACM International Conference on Computing Frontiers, CF 2018, Ischia, Italy, May 08-10, 2018, ACM, 2018, pp. 279–286. URL: https://doi.org/10.1145/3203217.3205340. doi:10.1145/3203217.3205340.
[28] I. Colonnelli, R. Birke, G. Malenza, G. Mittone, A. Mulone, J. Galjaard, L. Y. Chen, S. Bassini, G. Scipione, J. Martinovič, V. Vondrák, M. Aldinucci, Cross-Facility Federated Learning, Procedia Computer Science 240 (2024) 3–12. doi:10.1016/j.procs.2024.07.003.
[29] NVIDIA Developer, NVIDIA Hopper Architecture In Depth, 2023. URL: https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/, accessed: 2026-01-07.
[30] O. Press, N. A. Smith, M. Lewis, Train short, test long: Attention with linear biases enables input

length extrapolation, arXiv preprint arXiv:2108.12409 (2022). URL: https://arxiv.org/abs/2108.12409. arXiv:2108.12409.

[31] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, Y. Liu, Roformer: Enhanced transformer with rotary position embedding, Neurocomputing 568 (2024) 127063.

[32] K. Jordan, Y. Jin, V. Boza, Y. Jiacheng, F. Cesista, L. Newhouse, J. Bernstein, Muon: An optimizer for hidden layers in neural networks, 2024. URL: https://kellerjordan.github.io/posts/muon/.

[33] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), 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), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: https://aclanthology.org/2025.naacl-long.34/. doi:10.18653/v1/2025.naacl-long.34.