GENERATIVE MODEL FOR LESS-RESOURCED LANGUAGE WITH 1 BILLION PARAMETERS

Domen VREŠ,[1] Martin BOŽIČ,[1] Aljaž POTOČNIK,[2] Tomaž MARTINČIČ,[2] Marko ROBNIK-ŠIKONJA[1]

[1]University of Ljubljana, Faculty of Computer and Information Science

[2]XLAB d.o.o.

Large language models (LLMs) are a basic infrastructure for modern natural language processing. Many commercial and open-source LLMs exist for English, e.g., ChatGPT, Llama, Falcon, and Mistral. As these models are trained on mostly English texts, their fluency and knowledge of low-resource languages and societies are superficial. We present the development of large generative language models for a less-resourced language. GaMS 1B - Generative Model for Slovene with 1 billion parameters was created by continuing pretraining of the existing English OPT model. We developed a new tokenizer adapted to Slovene, Croatian, and English languages and used embedding initialization methods FOCUS and WECHSEL to transfer the embeddings from the English OPT model. We evaluate our models on several classification datasets from the Slovene suite of benchmarks and generative sentence simplification task SENTA. We only used a few-shot in-context learning of our models, which are not yet instruction-tuned. For classification tasks, in this mode, the generative models lag behind the existing Slovene BERT-type models fine-tuned for specific tasks. On a sentence simplification task, the GaMS models achieve comparable or better performance than the GPT-3.5-Turbo model.

Keywords: large language models, generative models, knowledge transfer, OPT model, GaMS model, language adaptation

1 INTRODUCTION

Large language models (LLMs), in particular generative LLMs like GPT models (Brown et al., 2020; OpenAI et al., 2024), have dramatically transformed natural language processing (NLP), advancing the understanding and generation of human language. As a result of this rapid development, new open-source decoder-type transformer LLMs such as Llama, Falcon, Mistral, and many others are released on a monthly basis. These models are trained on high-resource languages (primarily English), leaving many less-resource languages, such as Slovene, behind. In this work, we present the development of GaMS 1B (Generative Model for Slovene), the first Slovene open-source generative model with 1 billion parameters. The aim is to transfer recent advancements in language technologies from English to Slovene and, therefore, improve the technological development of Slovene. We release the model under open-source license. The creation of the model is fairly general and offers useful lessons to other less-resourced languages.

The main problem in training LLMs for Slovene is the lack of data. For example, the Llama 3 model (AI@Meta, 2024) was trained on 15 trillion tokens, while the currently available Slovene corpora contain around 11 billion tokens, a thousand times fewer. This means that training an LLM from scratch for Slovene is unfeasible. Hence, we adapt the already trained English OPT model (Zhang et al., 2022) to Slovene. To increase the amount of available training data, we also include texts from Croatian, Bosnian, and Serbian languages, which can improve the models' performance due to the language similarity. Taking an English model as a starting point raises the problem of the model's vocabulary, as the existing one is not adapted to Slovene, resulting in an inefficient tokenization of Slovene texts (i.e. considerably more tokens are generated compared to efficient tokenization). To solve this problem, we train a new tokenizer and employ embedding initialization methods WECHSEL (Minixhofer et al., 2022) and FOCUS (Dobler & de Melo, 2023) to transfer the embeddings from the English model to ours with the Slovene-tailored vocabulary.

An efficient evaluation of LLMs poses an additional challenge for low-resource languages. We demonstrate that models can not be directly compared based on training/validation losses observed during generative pretraining. The main reason is different vocabularies, as distributions of their output tokens differ, impacting the cross-entropy loss computation. English models are often evaluated on benchmarks testing models' reasoning, language understanding, etc. Such benchmarks are rare in Slovene, and using machine translation on complex datasets is mostly infeasible due to contextual differences between the languages. Hence, additional effort is required to obtain and adapt such benchmarks to a new language. We evaluate our models on three benchmarks already created or adapted to Slovene: the Slovene adaptation of the SuperGLUE benchmark suite (Žagar et al., 2020), the Slovene natural language inference dataset SI-NLI (Klemen et al., 2022), and the sentence simplification dataset SENTA (Žagar et al., 2024).

The paper is organized into six sections. In Section 2, we present related work on the development of large language models and transferring their knowledge to low-resource languages. In Section 3, we present the data used for training of our GaMS model. We offer a detailed technical description of GaMS model, i.e. the training of a new tokenizer, embedding transfer methods, and training details, in Section 4. In Section 5, we evaluate the models. We provide conclusions and directions for further work in Section 6.

2 RELATED WORK

New LLMs (or model families) are released on a monthly basis, with the most notable representatives being LLaMa (AI@Meta, 2024; Touvron, Lavril, et al., 2023; Touvron, Martin, et al., 2023), Falcon (Penedo et al., 2023), Phi (Li et al., 2023), Mistral (Jiang et al., 2023), and Mixtral (Jiang et al., 2024). Most of these models were trained on mainly English texts, and those trained on more languages have seen a very small proportion of Slovene texts compared to more represented languages. Therefore, the performance of these models for Slovene can be improved with additional pretraining on Slovene texts.

To spread the benefits of LLMs to languages other than English, multilingual models were developed. BLOOM (Workshop et al., 2023), YAYI 2 (Luo et al., 2023), PolyLM (Wei et al., 2023) and XGLM (Lin et al., 2022) were all trained on over 15 languages. However, they do not achieve the performance of state-of-the-art English models due to a lower number of parameters or smaller training data size. Additionally, Slovene is not included in the supported languages or is included in such a minority that the models do not work well for Slovene.

Recently, some English models were adapted for specific languages. Most notable examples are GPT-SW3 (AI-Sweden, 2024) for Swedish, Chinese LLaMa (Cui et al., 2023) and Open-Chinese-LLaMA (OpenLMLab, 2023) for Chinese, and Gervasio (Santos et al., 2024) for Portuguese. However, these models were either trained from scratch (GPT-SW3), did not use embedding transfer methods after vocabulary expansion (Chinese LLaMA and Open-Chinese-LLaMa), or were just instruction tuned for the target language (Gervasio).

Slovene is not without LLMs, though. However, existing works focused on encoder-type models, such as CroSloEngual BERT (Ulčar & Robnik-Šikonja, 2020) and SloBERTa (Ulčar & Robnik-Šikonja, 2021), or encoder-decoder-type models, such as SloT5 (Ulčar & Robnik-Šikonja, 2023). The only working open-source decoder-type model for Slovene we are aware of is GPT-sl-base (Ulčar & Robnik-Šikonja, 2022), which has only 100 million parameters and was trained on only 5 billion unique tokens and is therefore not comparable to the proposed model.

3 PRETRAINING DATA

LLMs require huge training sets. We use existing Slovene corpora for additional pretraining of our model. Our training corpora covers different types of text, such as news articles (Trendi (Kosem et al., 2023) - up to and including September 2023), academic works (KAS (Žagar et al., 2022)), web crawls (mC4 (Raffel et al., 2020), MaCoCu (Bañón et al., 2023), CC100 (Wenzek et al., 2020)), and a mixture of them (Metafida (Erjavec, 2023)). These corpora collectively contain around 10 B tokens, while Hoffman scaling laws (Hoffmann et al., 2022) suggest 20 B tokens as a suitable quantity for 1 B model. Note that pretraining of the recent Llama 3 model (AI@Meta, 2024) used even more tokens than these scaling laws suggest resulting in still better model performance. For these two reasons, we also include Croatian, Bosnian, and Serbian texts to increase our training data. We hypothesize that using these languages should improve the model's performance due to their similarity to Slovene. This was also shown in previous works, such as CroSloEngual BERT (Ulčar & Robnik-Šikonja, 2020). Additionally, we use English Wikipedia (Wikimedia Foundation, 2022) and CC-News (Hamborg et al., 2017) to prevent the model's forgetting of English. The used corpora and their properties are shown in Table 1.

We performed an additional cleaning of the KAS corpus, containing some unwanted artifacts due to the scanning of PDF documents. We cleaned these artifacts using the following heuristics. We define a set of problematic characters (Non-ASCII characters except Slovene characters (č, ž, š) and characters

Table 1: Corpora used for additional pretraining of GaMS 1B model. CBS stands for a combination of Croatian, Bosnian, and Serbian languages. The ”OPT tokenizer” column shows the number of resulting tokens when the texts are tokenized with the original OPT tokenizer, while the ”Slovene tokenizer” shows the number of tokens when the texts are tokenized with our tokenizer, described in Section 4.1.

Corpus	*Language*	*# tokens (OPT tokenizer)*	*# tokens (Slovene tokenizer)*
Metafida	Slovene	6.59 B	3.35 B
KAS	Slovene	3.61 B	1.66 B
Trendi	Slovene	1.4 B	0.68 B
mC4	Slovene	5.5 B	2.88 B
MaCoCu	Slovene	4.68 B	2.34 B
CC100	Slovene	0.54 B	0.29 B
Rižnica	Croatian	0.21 B	0.11 B
HrNews	Croatian	4.16 B	2.14 B
MaCoCu	CBS	15.65 B	8.63 B
Wikipedia	English	4.7 B	5.61 B
CC-News	English	0.4 B	0.46 B
Total	/	**47.44 B**	**28.13 B**

of other alphabets, such as Chinese, Greek, Cyrillic, etc.). We consider an unwanted artifact a sequence of tokens (texts are tokenized using NLTK (Bird et al., 2009) tokenizer) with a combined length of at least 5 characters that contain only problematic characters. We remove these sequences. We did not clean other corpora, as they were already thoroughly cleaned.

We performed near deduplication on Slovene corpora using the Onion tool (Pomikálek, 2011). Similarly to Ulčar and Robnik-Šikonja (2020), we use 9-grams with a duplicate content threshold of 0.9. The statistics, shown in Table 1, are computed on cleaned and deduplicated corpora.

4 ADDITIONAL PRETRAINING PROCESS

Due to the lack of Slovene data, we cannot train the model from scratch and have to use an already pretrained English model as a starting point. Most generative decoder-based LLMs have more than 1 B parameters as their performance scales with the number of parameters. There were not many open-source 1

Table 2: An example of an 1 B Falcon model output. The output shows clear signs of repetition and a tendency to generate too-long texts. The generation was interrupted, when the maximal number of allowed tokens was reached.

Prompt: This is the initial run of the Falcon model. My first message to the world is:
Output: I've done my research. I'm not a fan of the Falcon and I think the Falcon is a bad idea. I've spent some time on this site looking at the Falcon and the Falcon is not a very good idea. I've read about it and I have read a lot about it and the Falcon is not a very good idea. It is a very expensive, very inefficient, and very bad idea. It is an expensive, inefficient, and terrible idea. The reason I am writing this post is that I want people to see what I'm talking about and to see why I think the Falcon is a bad idea. I am not a huge fan of the Falcon and I think it's a horrible idea. The Falcon is a terrible idea. The Falcon is a terrible idea. It's an incredibly bad idea. It

B models available at the time we started with our work; e.g., Microsoft's Phi-1.5 (Li et al., 2023) was not yet published under an open-source license. Two possible choices were Falcon (Penedo et al., 2023) and OPT (Zhang et al., 2022). While Falcon models generally perform better on LLM leaderboards, such as Beeching et al. (2023), the 1 B version was trained only on a 350 B tokens subset of the RefinedWeb dataset (Penedo et al., 2023). For comparison, the larger versions were trained on the whole RefinedWeb (1.19 T tokens) and some other corpora, resulting in a training dataset of around 1.5 T tokens. Even the authors of the 1 B Falcon model advise treating this model only as a research artifact. By manually testing the 1 B Falcon model on some prompts, we found out that the model tends to repeat itself (even with sampling), generates longer outputs than necessary, and outputs meaningless sentences on a regular basis. An example of such output is shown in Table 2. When testing the 1.3 B version of the OPT model in a similar way, it made a better impression, and we chose it as our starting model.

OPT follows the GPT-3 architecture (Brown et al., 2020). The 1.3 B model has 24 layers with 32 attention heads. Its hidden (embedding) dimension is 2048, it uses Pre-LayerNorm (Xiong et al., 2020), ReLU activation function, absolute learned positional embeddings, and the encoder sequence length (context length) is 2048. It offsets the positional embeddings by 2 (instead of starting with 0, it starts with 2), appends EOS token at the beginning of the sequence, and its vocabulary size is 50.272.

We additionally pretrain two versions of this model, one with the original OPT vocabulary and the other with Slovene vocabulary (see Section 4.1). We refer to the versions with the original OPT vocabulary as OPT_GaMS models and to versions with Slovene vocabulary as GaMS models for the rest of this paper.

4.1 Building Slovene vocabulary of the model

We train the tokenizer for the new vocabulary using the CC100, KAS, Metafida, and HrNews (Ljubešić et al., 2024) corpora. We initially trained six different tokenizers, primarily differing in size. Our aim for the tokenizer is to be efficient on both English and Slovene texts. For vocabulary evaluation, we utilize the OpenSubtitles (Lison & Tiedemann, 2016) dataset, which includes Slovene and English subtitles, totaling around 19 million aligned lines in these two languages.

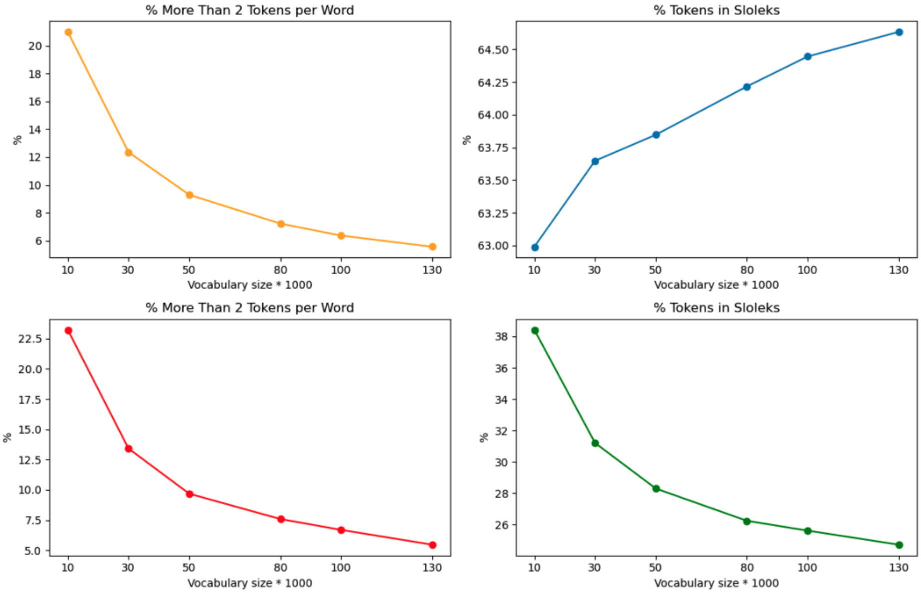
To train the tokenizer, we utilize the SentencePiece library (Kudo & Richardson, 2018) with the Byte Pair Encoding (BPE) (Sennrich et al., 2016) segmentation algorithm. We create a SentencePiece tokenizer model with a specified vocabulary size and include special tokens such as ‘<s>‘ (beginning of sequence), ‘</s>‘ (end of sequence), ‘<pad>‘ (padding token), and ‘<unk>‘ (unknown token).

We evaluate the tokenizer using three metrics. The first metric was described by Ali et al. (2023) and measures how many words are written with two or more tokens. A good tokenizer shall keep this number relatively low. The second metric assesses how many vocabulary tokens are part of the Slovene lexical database Sloleks (Dobrovoljc et al., 2019). We wish for a high value of this metric. Lastly, we create a distributional histogram displaying 10 different groups of columns, illustrating for each tokenizer the number of words written with 1, 2, ..., up to 10 or more tokens. We wish for the bulk of mass in the histogram to be on the left-hand side of the histogram. We show the results of the first two metrics, evaluated on Slovene and English subtitles datasets, in Figure 1.

The results on the Slovene and English OpenSubtitles datasets show that larger vocabularies yield better results.[1] However, the improvement in results slightly

[1] We observe that the percentage of tokens in Sloleks increases when evaluated on the Slovene dataset and decreases when evaluated on the English dataset. This trend is favorable in both

Figure 1: The evaluation of different vocabulary sizes tested on the Slovene Subtitles dataset (upper two graphs) and the English Subtitles dataset (lower two graphs).



diminishes when the vocabulary size increases from 80,000 to 100,000 tokens. As a larger vocabulary implies more model parameters, which consequently require more data for training, more required computational resources and longer training times, we have to settle for a suitable sweet spot. We opt for a vocabulary size of 80,000 tokens as our choice for the 1 B model.

4.2 Embedding transfer

Zhao et al. (2024) recently showed that vocabulary change (or expansion) can have a negative impact on the model's performance when the new embedding matrix is initialized randomly. They performed their experiments using Chinese LLaMA (Cui et al., 2023). As the Chinese language uses specific characters that are not well-represented in the vocabularies of English LLaMA models, the

cases. Initially, we have tokenized parts of words that may be similar across both languages. As the token size increases, more complete English words, which are not in the Sloleks dictionary, appear, while more complete Slovene words, which are in Sloleks, also appear.

vocabulary change (or expansion) seemed a necessary step in adapting the model for Chinese. However, if vocabulary change/expansion had a negative effect for Chinese models, it should have an even more negative impact for the Slovene model. Nevertheless, the benefit of a vocabulary change is a larger context window of the model. While the number of tokens the model can process (2048 in case of OPT) is not dependent on its vocabulary, the amount of text that can be tokenized using this number of tokens is. As seen in Table 1, tokenizing the same amount of Slovene text with OPT tokenizer results in twice as many tokens as tokenizing it with Slovene tokenizer. Hence, when using the Slovene tokenizer, the model can process Slovene texts that are double the size of those processed by the OPT tokenizer.

To keep the upsides of vocabulary change and mitigate its adverse effect on the model's performance, we tried to initialize the embedding matrix using WECHSEL (Minixhofer et al., 2022) and FOCUS (Dobler & de Melo, 2023) initialization methods. These methods initialize the embedding matrix for a new vocabulary based on the embedding matrix of the original vocabulary. Let T^s be the source tokenizer (OPT tokenizer in our case) with vocabulary V^s and corresponding embedding matrix E^s. We have a target tokenizer T^t (tokenizer from Section 4.1) with vocabulary V^t. Our goal is to initialize the embedding matrix E^t. WECHSEL and FOCUS do that by computing the similarities between tokens in a common embedding space W. We denote the representations of V^s and V^t in W with W^s and W^t. Both WECHSEL and FOCUS use FastText embeddings (Bojanowski et al., 2017) as W. We test both the original versions of these methods and our own versions, where we replace the FastText embeddings with CroSloEngual BERT embeddings (Ulčar & Robnik-Šikonja, 2020).

We denote models obtained by using WECHSEL/FOCUS as WECHSEL/FOCUS GaMS models. Additionally, OPT uses the same weights for embedding and output layer. Hence, it makes sense to transfer the output layer as well. We denote the models, where output layer is also transfered as WECHSEL/FOCUS Tied models.

4.2.1 THE WECHSEL EMBEDDINGS TRANSFER METHOD

WECHSEL (Minixhofer et al., 2022) obtains representations of vocabulary in source and target embeddings W^s and W^t by applying monolingual fastText word embeddings to V^s and V^t and aligning them using the Orthogonal Procrustes method (Schönemann, 1966; Artetxe et al., 2016) with bilingual dictionaries[2]. Based on this embeddings, it computes the cosine similarity $s_{x,y}$ between every pair $x \in V^t, y \in V^s$ using the following equation:

$$s_{x,y} = \frac{{w_x^t}^T w_y^s}{\|w_x^t\| \cdot \|w_y^s\|}, \tag{1}$$

where column vectors $w_x^t$and w_y^s denote the representations of x and y in W^t andW^s.

The target embeddings in E^t are initialized as a convex combination of embeddings in E^s. Let $\mathcal{J}_x \subset V^s$ denote the set of k nearest neighbors of $x \in V^t$ based on $s_{x,y}$ (k is the hyperparameter of the method). The embedding $e_x^t \in E^t$ is then computed using the softmax function:

$$e_x^t = \frac{\sum_{y \in \mathcal{J}_x} \exp(s_{x,y}/\tau) \cdot e_y^s}{\sum_{y' \in \mathcal{J}_x} \exp(s_{x,y'}/\tau)}, \tag{2}$$

where e_y^s denotes the embedding of $y \in V^s$ in E^s and τ denotes the temperature hyperparameter. We use $k = 10$ and $\tau = 0.1$ (these are default WECHSEL values) in our models.

4.2.2 THE FOCUS EMBEDDINGS TRANSFER METHOD

The FOCUS embeddings transfer method (Dobler & de Melo, 2023) initializes the target embeddings based on tokens that appear both in V^s and V^t (overlap). Let $O = V^s \cap V^t = \{o_1, o_2, ..., o_n\}$. The target embeddings of tokens in O are the same as their source embeddings:

$$\forall o \in O : e_o^t = e_o^s. \tag{3}$$

The set of non-overlapping (additional) target tokens is defined as $A = V^t \setminus O$. The embeddings e_a^t are computed based on similarities between tokens from A and O. Hence, FOCUS does not need W^s but needs only W^t, which is obtained by FastText. The difference between FOCUS and WECHSEL is that WECHSEL uses pretrained FastText, and FOCUS trains it on unlabeled data in the target language. Based on W^t, similarity $s_{a,o}$ is computed using Equation 1 for every

[2] WECHSEL code comes with already aligned embeddings, hence we did not need to align them.

pair $a \in A, o \in O$. For every $a \in A$, FOCUS defines the similarity score vector as:

$$c_a = [s_{a,o_1}, s_{a,o_2}, ..., s_{a,o_n}]. \tag{4}$$

Based on c_a, the vector of weights w_a is computed using sparsemax function (Martins & Astudillo, 2016):

$$w_a = \text{sparsemax}(c_a). \tag{5}$$

The target embedding $e_a^t \in E^t$ for an additional token $a \in A$ is then computed as:

$$e_a^t = \sum_{o \in O} w_{a,o} \cdot e_o^s. \tag{6}$$

We train the FastText model used with FOCUS on the same corpora as the tokenizer from Section 4.1. We train the FastText model for 3 epochs and include every token that occurs more than 10 times in the training dataset. The dimension of token vectors is set to 768.

4.2.3 USING CROSLOENGUAL BERT EMBEDDINGS

Croatian, Slovene, and English languages, which are part of our vocabulary, are also used in the CroSloEngual BERT model (CSE BERT). Hence, we try to upgrade WECHSEL and FOCUS by using the embedding matrix of CSE BERT as a common embedding space W. The reasoning is that CSE BERT embeddings of similar English, Slovene, and Croatian tokens shall be aligned since they are modeled by the same model. As CSE BERT has shown some promising results on SloBench classification tasks (Dragar, 2022), it should have good internal language knowledge. We expect that our approach will benefit the WECHSEL method more than FOCUS, as WECHSEL's bilingual alignment is not suitable for multi-lingual models such as ours. Even for mono-lingual models, we suspect that the linear alignment is the weakest point of WECHSEL, and our approach should address that. We refer to the models that are trained using CSE BERT embeddings as W as FOCUS/WECHSEL CSE models.

We use the following approach to embed the tokens from V^s and V^t using CSE BERT. Let $v \in V^s \cup V^t$ be the token we want to embed. First, we tokenize it with the CSE BERT's tokenizer. We denote this tokenization with t_v^{CSE}. Since CSE BERT vocabulary is not the same as V^s and V^t, v is tokenized using $k \geq 1$

tokens:

$$t_v^{CSE} = [t_{v,1}^{CSE}, ..., t_{v,k}^{CSE}]. \tag{7}$$

Let$e_{v,i}^{CSE}$, $1 \leq i \leq k$ denote the product of token $t_{v,i}^{CSE}$ with embedding matrix E^{CSE} of CSE BERT (the CSE BERT embedding of token $t_{v,i}^{CSE}$). We define the common space embedding $w_v \in W$ for v as:

$$w_v = \frac{1}{k} \sum_{i=1}^{k} e_{v,i}^{CSE}. \tag{8}$$

4.3 Training the 1B models

We train our models on the Slovene HPC Vega computer (60 GPU nodes, each containing 4 NVIDIA A100 GPUs with 40 GB of RAM). We use the NVidia NeMo toolkit (version 1.22, container 23.10) for training, enabling efficient parallelization over multiple nodes on the model and data levels. As NeMo does not support positional embedding offset and ReLU activation, we forked the NeMo repository[3] and added the support for the OPT models.

We train our models on 16 nodes, using tensor parallel rank 4, enabling one instance of the model to be located on a single node, which is faster than having the model split over multiple nodes. We use a batch size of 1024, which equals around 2 million tokens (batch size in tokens is obtained by multiplying batch size with the context length of the model). Given our data, this results in 22,000 training steps for the OPT_GaMS model and 13,400 training steps for the GaMS models. We use fused Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We use a cosine learning rate scheduler with minimal learning rate $\eta_{min} = 2 \cdot 10^{-5}$. The learning rate is first linearly increased from 0 to $\eta_{max} = 10 \cdot \eta_{min} = 2 \cdot 10^{-4}$ during warmup steps and then decayed using cosine function to η_{min}, being equal to η_{min} during the final constant steps. We use the following warmup and constant steps:

- OPT_GaMS: 1000 warmup steps, 1000 constant steps;
- GaMS: 2000 warmup steps, 500 constant steps.

When training the FOCUS/WECHSEL GaMS models, we freeze the inner parameters of the model for the first 1500 steps. During these steps, we train only the embedding and the output layer. This helps to avoid the catastrophic forgetting

[3] https://github.com/SloLama/NeMo

of the model, which can happen due to vocabulary change. We use 0.05% of our data as a validation set. Even though this percentage seems small, it still results in around 15 or 24 million (depending on tokenizer) validation tokens, which should be enough to detect potential overfitting. Additionally, we can not afford large validation sets due to low amount of training data.

As Muennighoff et al. (2023) showed, it might help to repeat the data when dealing with constrained data, we train the model for multiple epochs. We train the WECHSEL CSE GaMS model with both embedding and output layer transferred from the original OPT model (this is the best performing GaMS model on a single epoch according to validation loss) for 4 epochs. Additionally, we freeze the model's hidden layers (only the output and embedding layers are trained) for the entire first e poch. We train the whole model for the next 3 epochs. With a multi-epoch scenario, we set the LR scheduler's warmup steps to 10,000 and constant steps to 5,000.

Inspired by Li et al. (2023), we test training OPT_GaMS model (we choose OPT_GaMS instead of GaMS as GaMS seems to require more data due to a vocabulary change) only on "higher quality" data. We define higher quality data to be all data except web crawls; the selection includes news articles, literature, academic works, etc., and represents diverse, informative, and well-written texts. We use the following corpora: Metafida, KAS, Trendi, Rižnica, HrNews, Wikipedia, and CC-News. This results in around **21 B** tokens, encoded with OPT tokenizer. We train the model for 10,050 steps and set the LR scheduler's warmup and constant steps to 1,000 and 500, respectively. We refer to this model as OPT_GaMS Quality Data.

The training and validation cross-entropy losses observed during the training are shown in Figure 2. The plots were obtained using Weights & Biases platform.[4] While GaMS losses seem to be much larger than OPT_GaMS losses, the losses of these two model groups cannot be directly compared due to different vocabularies. Note that the loss is computed on different distributions (even though the training data is the same, it is tokenized into different tokens - even the ratios between languages are different as OPT tokenizer uses more tokens on average to tokenize Slovene words than Slovene tokenizers). To avoid unfair comparisons, we compare the losses of GaMS models. It is evident that FOCUS

[4] https://wandb.ai/site

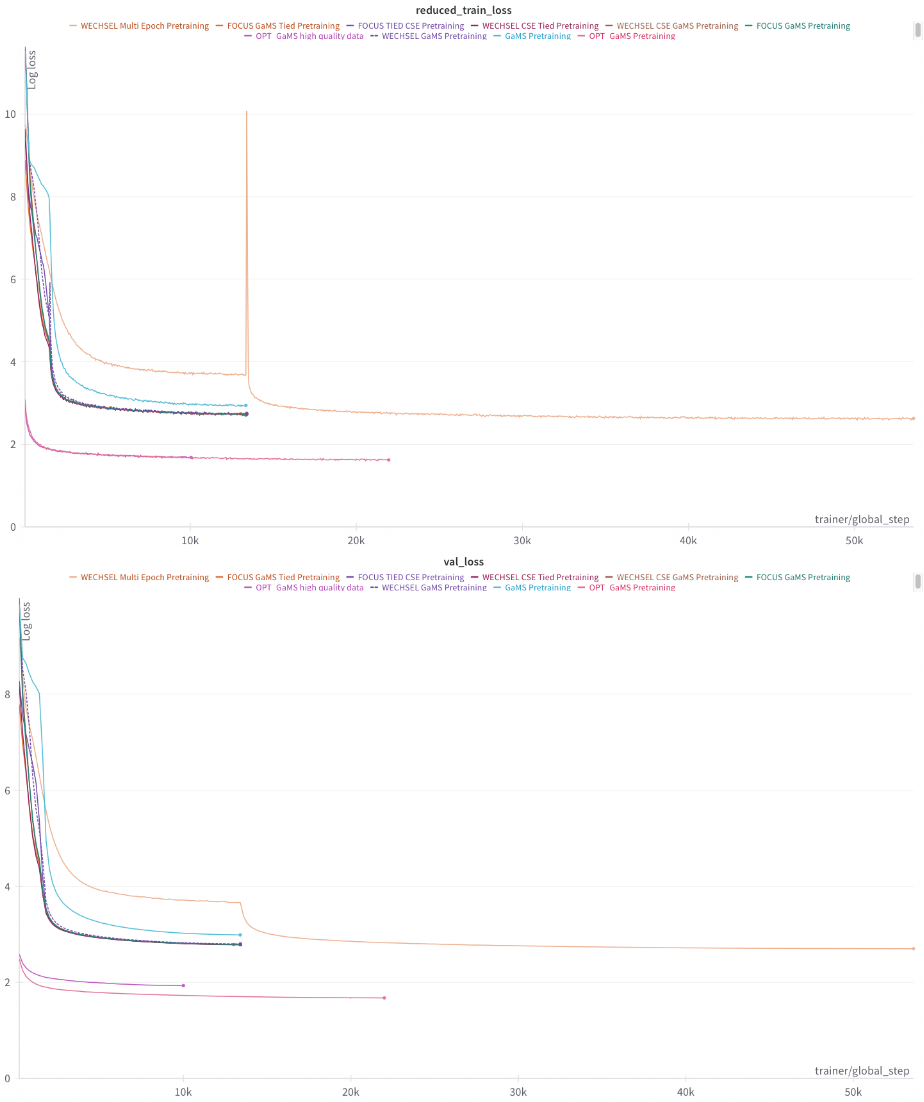
and WECHSEL improve the model performance compared to random initialization of the embedding matrix. While different transfer approaches behave differently in the early stages of the training, their losses all converge to a similar value (validation losses differ by less than **0.02**, showing no significant differ-ence in the performance of these methods). Although Figure 2 does not show this clearly, using multiple epochs actually reduces the validation loss (the final validation loss of multi-epoch model is **2.699**, while the final validation loss of its single-epoch counterpart is **2.781**). Furtherher, training the OPT_GaMS model only on ”higher quality” data does not improve its performance.

5 EVALUATION

LLMs are commonly benchmarked for knowledge, reasoning, safety, natural language understanding, etc. The commonly used benchmarking suites for LLM evaluation in English are GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), BIGBench (Srivastava et al., 2023), Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021), etc. The benchmarks for Slovene are very limited, as due to the complexity of most LLM benchmarks, obtaining them via pure machine translation is not a viable solution. Some SuperGLUE tasks are unsuitable even for human translation due to contextual differences between languages (such as the Word in Context task) and have to be rewritten for Slovene. Besides classification tasks contained in the Slovene SuperGLUE (Žagar et al., 2020) benchmarking suite, we used two more datasets: a natural language inference classification dataset SI-NLI (Klemen et al., 2022), which is already part of SloBench (Dragar, 2022), and sentence simplification task SENTA (Žagar et al., 2024) that tests text generation abilities of LLMs.

In our evaluation scenario, all models are evaluated using in-context learning, with few-shot prompts (models are not fine-tuned on given tasks but shown a few solved examples in the prompt). The in-context examples are randomly sampled from the training set (each test instance is given different examples). None of the models, apart from OPT_GaMS INZ, are instruction-tuned. The OPT_GaMS INZ model is LoRA (Hu et al., 2022) tuned on the QA dataset that was provided to us by Inštitut za novejšo zgodovino (INZ). The dataset consists of approximately 7,000 questions and answers and is not suitable for general-purpose instruction tuning, as it contains only one task. However, this fine-

Figure 2: The training (top) and validation (bottom) cross-entropy losses observed during the training. Note that the losses of OPT_GaMS models can not be directly compared to the losses of GaMS models due to differences in the distributions.



tuning helps with the evaluation of question-answering tasks, as it helps the model to generate the answer in the correct form. All models are evaluated using greedy sampling during the generation phase, i.e. the most probable token, according to the model, is always selected as the next generated token.

5.1 Classification tasks

The number of few-shot examples and number of test set instances for each dataset from the Slovene SuperGLUE suite and the SI-NLI dataset are shown in Table 3. The number of few-shot examples in prompt (k) is determined based on the models' performances on the validation set. The number of test set instances is quite low for BoolQ (30) and RTE (29) because only human-translated examples are used for the evaluation.

Table 3: The number of test examples and the number of in-context examples in prompts (k) per data set in SupeGLUE tasks and SI-NLI.

Task	k	*# test examples*
BoolQ	3	30
CB	5	250
COPA	5	500
MultiRC	2	333
RTE	3	29
WSC	4	146
SI-NLI	5	998

To adapt the classification tasks to generative LLMs, we wrote our own frame-work for the evaluation of generative models, where we specify the expected form of an answer in the prompt. We observe that our 1 B models struggle to understand what output is required to complete the tasks. This is typical for models below 5 B parameters; for example, Li et al. (2023) observed similar behavior for their Phi model. This behavior is not present in larger generative models for English, especially the ones trained for instruction following. Hence, we measure the percentage of invalid predictions where a model did not gener-ate the answer in a required form. We measure other metrics for each task only on valid predictions. The alternative would be to label the invalid predictions as wrong answers, but in this way, we cannot distinguish between invalid and wrong predictions. We also observe a high correlation between the majority label of few-shot examples and the models' output. Hence, we hypothesize that few-shot examples did not help the model to understand the tasks but only helped it with the form of the answer - more few-shot examples resulted in fewer invalid predictions.

The results for Slovene SuperGLUE tasks are shown in Tables 4 and 5. Overall, the performance of the models is quite similar and there is no model that would outperform others across all tasks. The models are outperformed by the representation model CroSloEngual BERT, which was fine-tuned on these tasks. As this model has seen significantly more training instances, the comparison is not fair but the score indicates what is achievable with relatively small LLMs. The most difficult task for the models, according to the percentages of invalid predictions, is MultiRC. In this task, the model is given a text, a question, and a list of answers. The model has to return the numbers of correct answers. We could make this task easier for the models by giving them each answer separately and asking them to classify them as correct or wrong. However, as the purpose of the task is to check whether the model can select the correct answers from multiple choices, we decided to present it in this more challenging form. OPT_GaMS INZ model produced a significantly lower percentage of invalid predictions on this task than other models, suggesting that instruction tuning should make the task less challenging.

Table 4: Test set results with 95 % confidence intervals for Slovene Super GLUE tasks BoolQ, CB, and COPA. Columns Acc. represent models' accuracy, and columns Inv. pred. represent the percentage of invalid predictions for each model. Confidence intervals are computed using standard error estimation for accuracy, and using quantile bootstrap for F1-score. The results for CroSloEngual BERT are copied from SloBench.

	BoolQ		*CB*			*COPA*	
Model	*Acc.*	*Inv. pred.*	*Acc.*	F_1	*Inv. pred.*	*Acc.*	*Inv. pred.*
OPT_GaMS	0.57 [0.38, 0.75]	0 %	0.44 [0.38, 0.50]	0.32 [0.26, 0.39]	0 %	0.46 [0.42, 0.51]	0 %
GaMS	0.50 [0.31, 0.69]	0 %	0.43 [0.37, 0.50]	0.30 [0.25, 0.33]	1.20 %	0.49 [0.44, 0.54]	17.20 %
WECHSEL GaMS	0.67 [0.49, 0.85]	0 %	0.50 [0.44, 0.56]	0.39 [0.32, 0.47]	1.20 %	0.48 [0.44, 0.52]	0.20 %
FOCUS GaMS	0.67 [0.49, 0.85]	0 %	0.51 [0.45, 0.58]	0.38 [0.31, 0.46]	1.60 %	0.48 [0.43, 0.53]	27.80 %
WECHSEL CSE	0.57 [0.38, 0.75]	0 %	0.50 [0.44, 0.56]	0.34 [0.30, 0.38]	0.40 %	0.48 [0.44, 0.53]	2.80 %
WECHSEL CSE Tied	0.47 [0.28, 0.66	0 %	0.51 [0.45, 0.57]	0.38 [0.32, 0.46]	2.40 %	0.48 [0.44, 0.53]	0.40 %
FOCUS CSE Tied	0.50 [0.31, 0.69]	0 %	0.48 [0.42, 0.54]	0.36 [0.29, 0.44]	0.40 %	0.47 [0.43, 0.51]	3.40 %
FOCUS GaMS Tied	0.53 [0.34, 0.72]	0 %	0.48 [0.42, 0.54]	0.36 [0.29, 0.44]	1.20 %	0.48 [0.43, 0.53]	12.00 %
OPT_GaMS Quality Data	0.60 [0.41, 0.79]	0 %	0.44 [0.37, 0.50]	0.35 [0.28, 0.43]	0.80 %	0.48 [0.44, 0.52]	0 %
OPT_GaMS INZ	0.60 [0.41, 0.79]	0 %	0.44 [0.37, 0.50]	0.32 [0.26, 0.40]	0 %	0.45 [0.41, 0.49]	0 %
WECHSEL Multi-Epoch	0.60 [0.41, 0.79]	0 %	0.51 [0.45, 0.57]	0.38 [0.31, 0.46]	0.80 %	0.46 [0.42, 0.51]	1.20 %
CroSloEngual BERT	0.73	/	0.79	0.74	/	0.57	/

Table 5: Test set results with 95 % confidence intervals for Slovene Super GLUE tasks MultiRC, RTE and WSC. Columns Acc. represent models' accuracy, column EM repre-sents the exact match between predictions and true labels, and columns Inv. pred. represent the percentage of invalid predictions for each model. Confidence intervals are computed using standard error estimation for accuracy and exact match, and using quantile bootstrap for F1-score. The results for CroSloEngual BERT are copied from SloBench.

	MultiRC			*RTE*		*WSC*	
Model	*EM*	F_1	*Inv. pred.*	*Acc.*	*Inv. pred.*	*Acc.*	*Inv. pred.*
OPT_GaMS	0.15 [0.02, 0.28]	0.43 [0.32, 0.54]	90.09 %	0.41 [0.22, 0.60]	0 %	0.51 [0.43, 0.60]	0 %
GaMS	0.03 [-0.03, 0.09]	0.16 [0.12, 0.20]	89.49 %	0.43 [0.23, 0.62]	3.45 %	0.42 [0.34, 0.50]	0 %
WECHSEL GaMS	0.15 [0.03, 0.26]	0.37 [0.30, 0.43]	87.69 %	0.43 [0.23, 0.62]	3.45 %	0.47 [0.38, 0.55]	0 %
FOCUS GaMS	0.11 [0.00, 0.21]	0.36 [0.29, 0.44]	88.59 %	0.54 [0.34, 0.73]	3.45 %	0.50 [0.42, 0.58]	0 %
WECHSEL CSE	0.06 [-0.01, 0.12]	0.26 [0.20, 0.31]	84.08 %	0.43 [0.23, 0.62]	3.45 %	0.45 [0.36, 0.53]	0 %
WECHSEL CSE Tied	0.12 [0.03, 0.21]	0.21 [0.17, 0.25]	84.38 %	0.46 [0.27, 0.66]	3.45 %	0.55 [0.47, 0.63]	0 %
FOCUS CSE Tied	0.09 [0.01, 0.17]	0.26 [0.21, 0.31]	83.48 %	0.43 [0.23, 0.62]	3.45 %	0.55 [0.47, 0.64]	0 %
FOCUS GaMS Tied	0.05 [0.02, 0.08]	0.22 [0.20, 0.24]	36.64 %	0.46 [0.27, 0.66]	3.45 %	0.49 [0.41, 0.58]	0 %
OPT_GaMS Quality Data	0.12 [0.04, 0.19]	0.32 [0.25, 0.39]	79.28 %	0.38 [0.19, 0.57]	0 %	0.47 [0.39, 0.55]	0 %
OPT_GaMS INZ	0.07 [0.04, 0.09]	0.34 [0.32, 0.37]	2.10 %	0.38 [0.19, 0.57]	0 %	0.45 [0.36, 0.53]	0 %
WECHSEL Multi-Epoch	0.13 [0.05, 0.21]	0.28 [0.22, 0.33]	79.28 %	0.50 [0.30, 0.70]	3.45 %	0.54 [0.46, 0.62]	0 %
CroSloEngual BERT	0.09	0.52	/	0.66	/	0.61	/

The results for the SI-NLI dataset are shown in Table 6. The performance of our models is quite similar, and the confidence intervals overlap. All models return invalid predictions for approximately half of the test instances (the best-performing model with respect to that metric is WECHSEL CSE, with 44.69 % of invalid predictions). The reason for these similarities is that all models perform poorly due to lack of task understanding. Hence, the models should be instruction-tuned first in order to spot any significant differences between them. The models are significantly outperformed by GPT and BERT models; again the comparison is not fair as BERT models were fine-tuned on this data set and GPT-3.5-Turbo is significantly larger.

5.2 Sentence simplification

The models introduced in this paper are generative. Therefore, it makes sense to evaluate them on language generation tasks. We choose sentence simplification task SENTA (Žagar et al., 2024). The model is given a sentence and asked to simplify it. Here, we observe that our models perform better than in classification tasks but there are still some problems with the task understanding, as the models sometimes return ”Poenostavi naslednji stavek.” (eng. ”Simplify the given sentence.”) as an answer in case of few-shot prompts. They return

Table 6: Test set results with 95 % confidence intervals for the SI-NLI dataset. Columns Inv. pred. represent the percentage of invalid predictions for each model. Confidence intervals are computed using standard error estimation for accuracy and using quantile bootstrap for F_1-score. The results for GPT-3.5-Turbo, SloBERTa, and CroSloEngual BERT are copied from SloBench.

Model	*Accuracy*	*Entailment F_1*	*Neutral F_1*	*Contradiction F_1*	*Inv. pred.*
OPT_GaMS	0.32 [0.27, 0.36]	0.38 [0.32, 0.45]	0.17 [0.10, 0.24]	0.34 [0.27, 0.40]	51.40 %
GaMS	0.29 [0.25, 0.33]	0.31 [0.24, 0.38]	0.32 [0.25, 0.38]	0.25 [0.19, 0.32]	50.00 %
WECHSEL GaMS	0.33 [0.29, 0.37]	0.39 [0.33, 0.45]	0.33 [0.27, 0.39]	0.26 [0.20, 0.32]	44.69 %
FOCUS GaMS	0.34 [0.30, 0.38]	0.40 [0.34, 0.46]	0.37 [0.31, 0.44]	0.20 [0.13, 0.26]	49.40 %
WECHSEL CSE	0.32 [0.28, 0.36]	0.38 [0.32, 0.43]	0.37 [0.30, 0.43]	0.17 [0.11, 0.24]	47.80 %
WECHSEL CSE Tied	0.35 [0.31, 0.39]	0.40 [0.34, 0.46]	0.41 [0.35, 0.47]	0.20 [0.14, 0.27]	48.30 %
FOCUS CSE Tied	0.34 [0.30, 0.38]	0.38 [0.32, 0.44]	0.38 [0.32, 0.44]	0.23 [0.17, 0.30]	47.19 %
FOCUS GaMS Tied	0.32 [0.28, 0.36]	0.37 [0.31, 0.43]	0.37 [0.31, 0.44]	0.20 [0.14, 0.26]	47.19 %
OPT_GaMS Quality Data	0.31 [0.27, 0.35]	0.38 [0.32, 0.44]	0.28 [0.22, 0.35]	0.28 [0.22, 0.35]	47.39 %
OPT_GaMS INZ	0.30 [0.26, 0.35]	0.36 [0.29, 0.42]	0.25 [0.18, 0.32]	0.29 [0.23, 0.36]	53.31 %
WECHSEL Multi-Epoch	0.30 [0.26, 0.34]	0.37 [0.31, 0.43]	0.37 [0.31, 0.43]	0.17 [0.11, 0.24]	51.10 %
GPT-3.5-Turbo	0.86	0.85	0.82	0.90	/
SloBERTa	0.74	0.76	0.71	0.64	/
CroSloEngual BERT	0.66	0.69	0.63	0.66	/

this sentence, as this is the instruction added to each example in the prompt and is consequently the most common sentence in the prompt.

We evaluate our models using different values k of few-shot examples. We test values $k \in \{0, 3, 5, 10\}$. We use SARI score[5] as an evaluation metric. SARI score is commonly used to evaluate text simplification systems. It compares the system's output to both the input and reference output. It computes the F_1-score for added and preserved tokens and precision for deleted words. It is computed using the following equation:

$$\text{SARI} = \frac{F_{1,add} + F_{1,keep} + P_{del}}{3}, \tag{9}$$

where $F_{1,add}$ and $F_{1,keep}$ represent the 4-gram F_1 score for add/keep operations and P_{del} denotes the 4-gram precision score for delete operations. The goal is to have as high F_1 and precision scores as possible, meaning that higher SARI score is better.

The results are shown in Table 7. All models perform similarly (no significant differences between their SARI scores). Using a larger number of few-shot examples seems to improve the performance of the majority of the models

[5] https://huggingface.co/spaces/evaluate-metric/sari

(the exception here is LoRA-tuned OPT_GaMS INZ, which works best in the 0-shot scenario). Surprisingly, our models perform similarly or better than GPT-3.5-Turbo. Our best-performing model (WECHSEL GaMS in the 10-shot scenario) also outperforms the best-performing SloT5 model that was trained on this task. However, the differences in the SARI scores are not significant. We believe that the performance of our models could improve drastically with instruction-tuning, as the models would better understand the task instruction.

Table 7: SARI scores with 95 % confidence intervals on SENTA task. Confidence intervals were computed using quantile bootstrap method. Value of k in columns denotes the number of shown examples in few-shot prompts. The results for GPT and T5 models are copied from Žagar et al. (2024).

Model	$k = 0$	$k = 3$	$k = 5$	$k = 10$
OPT_GaMS	39.38 [38.63, 40.16]	38.51 [37.60, 39.46]	39.49 [38.67, 40.40]	39.67 [38.80, 40.60]
GaMS	39.58 [38.76, 40.47]	38.92 [37.96, 39.86]	37.98 [37.10, 38.90]	39.18 [38.37, 40.06]
WECHSEL GaMS	39.34 [38.59, 40.15]	39.53 [38.55, 40.43]	**39.87 [39.01, 40.77]**	**41.62 [40.82, 42.30]**
FOCUS GaMS	38.50 [37.77, 39.37]	**40.16 [39.27, 41.07]**	39.67 [38.81, 40.57]	41.16 [40.41, 41.89]
WECHSEL CSE	39.02 [38.28, 39.83]	39.42 [38.49, 40.35]	39.22 [38.37, 40.05]	40.54 [39.79, 41.26]
WECHSEL CSE Tied	38.77 [37.96, 39.60]	38.67 [37.79, 39.61]	39.29 [38.41, 40.20]	40.91 [40.13, 41.71]
FOCUS CSE Tied	38.93 [38.15, 39.77]	38.95 [38.02, 39.92]	39.38 [38.54, 40.25]	40.98 [40.15, 41.79]
FOCUS GaMS Tied	38.80 [37.99, 39.67]	40.05 [39.19, 40.97]	39.74 [38.86, 40.64]	41.50 [40.79, 42.20]
OPT_GaMS Quality Data	38.76 [37.97, 39.58]	37.62 [36.72, 38.47]	38.48 [37.64, 39.44]	39.02 [38.14, 39.91]
OPT_GaMS INZ	**40.29 [39.49, 41.10]**	37.88 [36.99, 38.88]	38.58 [37.72, 39.54]	38.90 [38.00, 39.85]
WECHSEL Multi-Epoch	38.80 [37.96, 39.62]	40.06 [39.23, 40.96]	39.80 [38.97, 40.62]	40.99 [40.23, 41.67]
GPT-3.5-Turbo	38.76			
SloT5-small	39.79			
mT5-small	39.09			
SloT5-large	41.01			

We can conclude that our 1 B Slovene models are not suitable for in-context learning of classification tasks but work well in generative tasks. Their performance on classification tasks with fine-tuning remains part of the future work.

6 CONCLUSION

In this work, we presented the new 1 B Slovene generative model GaMS,[6] which is based on the English OPT model. The model is the first fully open-source generative language model for Slovene. Based on the analysis of different vocabulary sizes, we created a new tokenizer that was trained on Slovene, En-

[6] https://huggingface.co/cjvt/OPT_GaMS-1B

glish, and Croatian texts. We tested different embedding initialization methods and showed that they reduce both training and validation loss for next token prediction compared to random initialization.

The main challenge that we face in this work is a robust evaluation of the models. Direct comparison of training/validation losses for models using different vocabularies is not sensible, as the distributions of tokens (on which the loss is computed) are different. The comparison of models on classification benchmarking tasks is inconclusive, as the models do not really understand the tasks due to their size and lack of instruction tuning. We showed that our models perform better on generative tasks like sentence simplification but we need more tasks to get reliable conclusions on models performance.

In the future work, we will develop an instruction-following dataset and instruction-tune our models. This might improve the models performance on classification tasks, as the models will understand the evaluation tasks. For classification tasks, fine-tuning of models is also sensible. Additionally, we plan to train and release a larger model, where the differences between embedding initialization methods should be more significant.

7 ACKNOWLEDGMENTS

The work was supported by the Slovenian Research and Innovation Agency (ARIS) research project PoVeJMo (Adaptive Natural Language Processing with the Help of Large Language Models), core research programme P6-0411, and projects J7-3159, and L2-50070.

REFERENCES

AI-Sweden (Ed.). (2024). *GPT-SW3.* Retrieved May 28, 2024, from https://huggingface.co/AI-Sweden-Models/gpt-sw3-40b

AI@Meta. (2024). *Llama 3 Model Card.* https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

Ali, M., Fromm, M., Thellmann, K., Rutmann, R., Lübbering, M., Leveling, J., ... Flores-Herr, N. (2023). Tokenizer Choice For LLM Training: Negligible or Crucial? *CoRR*. https://doi.org/10.48550/arXiv.2310.08754

Artetxe, M., Labaka, G., & Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 2289–2294). https://aclanthology.org/D16-1250

Bañón, M., Chichirau, M., Esplà-Gomis, M., Forcada, M. L., Galiano-Jiménez, A., García-Romero, C., Kuzman, T., Ljubešić, N., van Noord, R., Pla Sempere, L., Ramírez-Sánchez, G., Rupnik, P., Suchomel, V., Toral, A., & Zaragoza-Bernabeu, J. (2023). *Slovene web corpus MaCoCu-sl 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1795

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., ... Wolf, T. (2023). *Open LLM Leaderboard.* https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard. Hugging Face.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.". https://books.google.si/books?id=KGIbfiiP1i4C

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://aclanthology.org/Q17-1010

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language Models are Few-Shot Learners. In *Advances in neural information processing systems,* 33, 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Cui, Y., Yang, Z., & Yao, X. (2023). *Efficient and Effective Text Encoding for Chinese LLaMA and Alpaca.* arXiv. https://arxiv.org/abs/2304.08177

Dobler, K., & de Melo, G. (2023). FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 13440–13454). https://aclanthology.org/2023.emnlp-main.829

Dobrovoljc, K., Krek, S., Holozan, P., Erjavec, T., Romih, M., Arhar Holdt, Š., Čibej, J., Krsnik, L., & Robnik-Šikonja, M. (2019). *Morphological lexicon Sloleks 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1230

Dragar, F. (2022). *SloBench: Slovenian Natural Language Processing Benchmark.* https://repozitorij.uni-lj.si/IzpisGradiva.php?lang=eng&id=136261

Erjavec, T. (2023). *Corpus of combined slovenian corpora metaFida 1.0. Slovenian language resource repository CLARIN.SI.* http://hdl .handle.net/11356/1775

Hamborg, F., Meuschke, N., Breitinger, C., & Gipp, B. (2017). news-please: A generic news crawler and extractor. In *Proceedings of the 15th international symposium of information science* (pp. 218–223). https://api.semanticscholar.org/CorpusID:5830937

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2021).

Measuring Massive Multitask Language Understanding. In *International conference on learning representations.* https://openreview.net/forum?id=d7KBjmI3GmQ

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. In *Advances in neural information processing systems,* 35, 30016–30030. https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. In *International conference on learning representations.* https://openreview.net/forum?id=nZeVKeeFYf9

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Le Saco, T., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. (2023). *Mistral 7b.* arXiv. https://arxiv.org/abs/2310.06825

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., ... Sayed, W. E. (2024). *Mixtral of Experts.* arXiv. https://arxiv.org/abs/2401.04088

Klemen, M., Žagar, A., Čibej, J., & Robnik-Šikonja, M. (2022). *Slovene natural language inference dataset SI-NLI.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1707

Kosem, I., Čibej, J., Dobrovoljc, K., Erjavec, T., Ljubešić, N., Ponikvar, P., Šinkec, M., & Krek, S. (2023). *Monitor corpus of Slovene Trendi 2023-09.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1879

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing: System demonstrations* (pp. 66–71). https://aclanthology.org/D18-2012

Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., & Lee, Y. T. (2023). *Textbooks are all you need ii:* ***phi-1.5*** *technical report.* arXiv. https://arxiv.org/abs/2309.05463

Lin, X. V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., ... Li, X. (2022). Few-shot Learning with Multilingual Generative Language Models. In *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 9019–9052). https://aclanthology.org/2022.emnlp-main.616

Lison, P., & Tiedemann, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)* (pp. 923–929). European Language Resources Association (ELRA). https://aclanthology.org/L16-1147

Ljubešić, N., Suchomel, V., Rupnik, P., Kuzman, T., & van Noord, R. (2024). Language models on a diet: Cost-efficient development of encoders for closely-related languages via additional pretraining. In *Proceedings of the 3rd annual meeting of the special interest group on under-resourced languages @ lrec-coling 2024* (pp.

189–203). https://aclanthology.org/2024.sigul-1.23

Luo, Y., Kong, Q., Xu, N., Cao, J., Hao, B., Qu, B., ... Zeng, D. (2023). *YAYI 2: Multilingual Open-Source Large Language Models.* arXiv. https://arxiv.org/abs/2312.14862

Martins, A., & Astudillo, R. (2016). From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In *Proceedings of the 33rd international conference on machine learning* (pp. 1614–1623). https://proceedings.mlr.press/v48/martins16.html

Minixhofer, B., Paischer, F., & Rekabsaz, N. (2022). WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 3992–4006). https://aclanthology.org/2022.naacl-main.293

Muennighoff, N., Rush, A. M., Barak, B., Scao, T. L., Tazi, N., Piktus, A., Tazi, N., Pyysalo, S., Wolf, T., & Raffel, C. (2023). Scaling Data-Constrained Language Models. In *Thirty-seventh conference on neural information processing systems* (pp. 50358–50376). https://proceedings.neurips.cc/paper_files/paper/2023/file/9d89448b63ce1e2e8dc7af72c984c196-Paper-Conference.pdf

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., ... Zoph, B. (2024). *GPT-4 Technical Report.* arXiv. https://arxiv.org/abs/2303.08774

OpenLMLab (Ed.). (2023). *Open-Chinese-LLaMA.* Retrieved May, 28, 2024, from https://huggingface.co/openlmlab/open-chinese-llama-7b-patch

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Alobeidli, H., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., & Launay, J. (2023). The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data Only. In *Advances in neural information processing systems,*36, pp. 79155–79172. https://proceedings.neurips.cc/paper_files/paper/2023/file/fa3ed726cc5073b9c31e3e49a807789c-Paper-Datasets_and_Benchmarks.pdf

Pomikálek, J. (2011). *Removing boilerplate and duplicate content from web corpora.* Doctoral theses, dissertations, Masaryk university, Faculty of informatics, Brno, Czech Republic. https://theses.cz/id/nqo9nn/

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res., 21*(1). https://dl.acm.org/doi/abs/10.5555/ 3455716.3455856

Santos, R., Silva, J. R., Gomes, L., Rodrigues, J., & Branco, A. (2024). Advancing Generative AI for Portuguese with Open Decoder Gervásio PT*. In *Proceedings of the 3rd annual meeting of the special interest group on under-resourced languages @ lrec-coling 2024* (pp. 16–26). https://aclanthology.org/2024.sigul-1.3

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem.

Psychometrika. https://doi.org/10.1007/BF02289451

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1715–1725). https://aclanthology.org/P16-1162

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A., Abid, A., Fisch, A., ... Wu, Z. (2023). Beyond the imitation game: quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 1–95. https://openreview .net/pdf?id=uyTL5Bvosj

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Roziere, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models.* arXiv. https:// arxiv.org/abs/2302.13971

Touvron, H., Martin, L., Stone, K. R., Albert, P., Almahairi, A., Babaei, Y., ... Scialom, T. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models.* arXiv. https:// arxiv.org/abs/2307.09288

Ulčar, M., & Robnik-Šikonja, M. (2020). FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In *Text, speech, and dialogue TSD 2020,* 12284. https://doi.org/10.1007/978-3-030-58323-1_11

Ulčar, M., & Robnik-Šikonja, M. (2021). SloBERTa: Slovene monolingual large pretrained masked language model.. https://api.semanticscholar.org/CorpusID:250066999

Ulčar, M., & Robnik-Šikonja, M. (2022). *GPT-sl-base.* Retrieved May, 28, 2024, from https://huggingface.co/cjvt/gpt-sl-base

Ulčar, M., & Robnik-Šikonja, M. (2023). Sequence-to-sequence pretraining for a less-resourced Slovenian language. *Frontiers in Artificial Intelligence, 6*. https:// www.frontiersin.org/articles/10.3389/frai.2023.932519

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). SuperGLUE: a stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd international conference on neural information processing systems.* Curran Associates Inc. https://dl.acm.org/ doi/10.5555/3454287.3454581

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP* (pp. 353–355). https://aclanthology.org/W18-5446

Wei, X., Wei, H., Lin, H., Li, T., Zhang, P., Ren, X., Li, M., Wan, Y., Cao, Z., Xie, B., Hu, T., Li, S., Hui, B., Yu, B., Liu, D., Yang, B., Huang, F., & Xie, J. (2023). *Polylm: An open source polyglot large language model.* arXiv. https://arxiv.org/abs/2307.06018

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., & Grave,

E. (2020). CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the twelfth language resources and evaluation conference* (pp. 4003–4012). https://aclanthology.org/2020.lrec-1.494

Wikimedia Foundation (Ed.). (2022). *Wikimedia Downloads.* Retrieved May 8, 2024, from https://huggingface.co/datasets/wikipedia

Workshop, B., :, Scao, T. L., Fan, A., Akiki, C., Pavlick, E., ... Wolf, T. (2023). *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model.* arXiv. https://arxiv.org/abs/2211.05100

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., ... Liu, T.-Y. (2020). On layer normalization in the transformer architecture. In *Proceedings of the 37th international conference on machine learning.* https://dl.acm.org/doi/abs/10.5555/3524938.3525913

Žagar, A., Kavaš, M., Robnik-Šikonja, M., Erjavec, T., Fišer, D., Ljubešić, N., Ferme, M., Borovič, M., Boškovič, B., Ojsteršek, M., & Hrovat, G. (2022). *Abstracts from the KAS corpus KAS-abs 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/ 1449

Žagar, A., Klemen, M., Robnik-Šikonja, M., & Kosem, I. (2024). SENTA: Sentence simplification system for Slovene. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (lrec-coling 2024)* (pp. 14687–14692). https://aclanthology.org/2024.lrec-main.1279

Žagar, A., Robnik-Šikonja, M., Goli, T., & Arhar Holdt, Š. (2020). *Slovene translation of SuperGLUE.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1380

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Singh Koura, P., Sridhar, Al., & Zettlemoyer, L. (2022). *OPT: Open Pre-trained Transformer Language Models.* arXiv. https://arxiv.org/abs/ 2205.01068

Zhao, J., Zhang, Z., Zhang, Q., Gui, T., & Huang, X. (2024). *LLaMA Beyond English: An Empirical Study on Language Capability Transfer.* arXiv. https://arxiv.org/abs/2401.01055

GENERATIVNI MODEL Z MILIJARDO PARAMETROV ZA JEZIK Z MANJ VIRI

Veliki jezikovni modeli so osnovna infrastruktura za sodobno obdelavo naravnega jezika. Za angleščino obstajajo številni komercialni in odprtokodni modeli, na primer ChatGPT, Llama, Falcon in Mistral. Ker so ti modeli učeni večinoma na angleških besedilih, sta njihovo znanje in poznavanje jezikov ter družb z manj viri površna. Predstavljamo razvoj novega generativnega velikega jezikovnega modela za jezik z malo viri. Za slovenski model, imenovan GaMS 1B (Generativni Model za Sloveščino), z 1 milijardo parametrov smo razvili nov tokenizator, prilagojen slovenščini, hrvaščini in angleščini, ter uporabili metodi inicializacije vektorskih vložitev FOCUS in WECHSEL za prenos vložitev iz obstoječega angleškega modela OPT. Zgrajene modele smo ovrednotili na slovenski zbirki klasifikacijskih učnih množic in na generativni nalogi poenostavljanja stavkov SENTA. Pri evalvaciji smo uporabili le učenje v kontekstu z nekaj učnimi primeri ter modele, ki še niso prilagojeni za sledenje navodilom. Pri takih nastavitvah so na klasifikacijskih nalogah zgrajeni generativni modeli zaostali za obstoječimi slovenskimi modeli tipa BERT, ki so bili prilagojeni za dane naloge. Pri nalogi poenostavljanja stavkov modeli GaMS dosegajo primerljive ali boljše rezultate kot model GPT-3.5-Turbo.

Keywords: veliki jezikovni modeli, generativni modeli, prenos znanja, OPT model, GaMS model, jezikovno prilagajanje