GPT-SW3: An Autoregressive Language Model for the Scandinavian Languages

Ariel Ekgren[1][*]**, Amaru Cuba Gyllensten**[1]**, Felix Stollenwerk**[1]**, Joey Öhman**[1]**,
Tim Isbister**[1]**, Evangelia Gogoulou**[2]**, Fredrik Carlsson**[2]**, Judit Casademont**[1]**,
Magnus Sahlgren**[1]

[1]AI Sweden, Sweden
[2]RISE, Sweden
[*]Corresponding author: `ariel.ekgren@ai.se`

Abstract

This paper details the process of developing the first native large generative language model for the North Germanic languages, GPT-SW3. We cover all parts of the development process, from data collection and processing, training configuration and instruction finetuning, to evaluation, applications, and considerations for release strategies. We discuss pros and cons of developing large language models for smaller languages and in relatively peripheral regions of the globe, and we hope that this paper can serve as a guide and reference for other researchers that undertake the development of large generative models for smaller languages.

Keywords: Large Language Models, Low-Resource Languages, Multilinguality

1. Introduction

There is a growing interest in building and applying Large Language Models (LLMs) for languages other than English. This interest has been fuelled partly by the unprecedented popularity of ChatGPT[1] that has propelled LLMs to the forefront of general awareness, and partly by the rapid commoditization of frameworks and infrastructure for training LLMs, which has drastically lowered the threshold for researchers to train and utilize LLMs. However, even with the existence of accessible frameworks such as Hugging Face Transformers[2] and commoditized compute infrastructure either through cloud or various national (and international) supercomputer initiatives, there are significant challenges to develop LLMs for smaller languages.

The perhaps most obvious challenge is access to sufficient amounts of diverse, high-quality data. Apart from the basic question whether sufficient amounts of data at all exists for a smaller language, there may be additional complicating issues related to compliance with regulatory frameworks such as GDPR, the EU AI Act, and questions pertaining to copyright. We describe our data collection efforts in Section 3 (and in a separate paper, Öhman et al. (2023)). Another challenge for prospective developers of LLMs is access to sufficient amounts of compute. Some countries have national compute infrastructure devoted to researchers, but such infrastructure may have limited GPU-resources, and access is typically regulated via specific allocation tiers, which may not be suitable for large-scale projects such as LLM training. On the other hand, cloud providers are typically always an easily accessible option, but can be prohibitively costly.

We have faced all of these challenges in our work on developing the first native LLM for the Scandinavian (or, more accurately, *North Germanic*) languages. The LLM, which we call **GPT-SW3**, is a continuation of our previous Swedish-only model (Ekgren et al., 2022a). GPT-SW3 is a collection of large decoder-only pretrained Transformer language models trained with a causal language modeling objective on a dataset containing approximately 320B tokens in Swedish, Norwegian, Danish, Icelandic, and English, as well as a set of 4 programming languages (Python, JavaScript, SQL and Shell script). The suite of models ranges from 126M to 40B parameters, and instruction-tuned versions are also available for some of these models. This paper details the entire development process, from data collection and processing, training configuration and instruction-tuning, to evaluation and considerations for model release.

2. Related Work

The era of LLMs arguably started with the 175B parameter GPT-3 model that was introduced in 2020 (Brown et al., 2020). During the last 3 years, we have seen a steady stream of new LLMs, exemplified by the models listed in Table 1. What counts as a "large" language model is of course not rigorously defined. Our compilation in Table 1 lists models that have been trained from scratch with more than 20 billion parameters, but this threshold is arbitrary, and could as well be 1B or 100B.

The majority of current LLMs are built from En-

---

[1]`chat.openai.com`
[2]`huggingface.co/docs/transformers`

Size	Language	Open	Name	Reference
20B	English	Yes	GPT-NeoX	Black et al. (2022b)
30B	English	Yes	MPT	Team (2023)
34B	Chinese, English	Yes	Yi	Young et al. (2024)
34B	Finnish, English	Yes	Poro	huggingface.co/LumiOpen/Poro-34B
40B	**Swedish, Norwegian Danish, Icelandic Faroese, English**	**Yes**	**GPT-SW3**	**This paper**
45B	Multilingual	Yes	Mixtral	Jiang et al. (2024)
50B	English	No	BloombergGPT	Wu et al. (2023)
65B	Multilingual	Yes	LLaMA	Touvron et al. (2023a)
70B	Multilingual	Yes	LLaMA-2	Touvron et al. (2023b)
70B	English	No	Chinchilla	Hoffmann et al. (2022b)
72B	Chinese, English	Yes	Qwen	Bai et al. (2023)
100B	Russian	Yes	YaLM	github.com/yandex/YaLM-100B
120B	English	No	Galactica	Taylor et al. (2022)
130B	Chinese, English	Yes	GLM	Zeng et al. (2022)
137B	English	No	LaMDA	Thoppilan et al. (2022)
175B	English	No	GPT-3	Brown et al. (2020)
175B	English	Yes	OPT	Zhang et al. (2022)
176B	Multilingual	Yes	BLOOM	BigScience (2022)
178B	English	No	Jurassic-1	Lieber et al. (2021)
180B	Multilingual	Yes	Falcon	Almazrouei et al. (2023)
200B	Chinese	No	PanGu-α	Zeng et al. (2021)
260B	Chinese	No	Ernie 3.0	Wang et al. (2021)
280B	English	No	Gopher	Rae et al. (2021)
314B	Multilingual	Yes	Grok-1	github.com/xai-org/grok-1
530B	English	No	Megatron-Turing	Smith et al. (2022)
540B	English	No	PaLM	Chowdhery et al. (2022)
?	Multilingual	No	Mistral	chat.mistral.ai
?	Multilingual	No	Gemini	gemini.google.com
?	Multilingual	No	GPT-4	openai.com/gpt-4
?	Multilingual	No	Claude	anthropic.com/claude

Table 1: LLMs with a parameter count of more than 20 billion, sorted by ascending size. The "language" column indicates the languages in the pretraining data (excluding code, which is present in a majority of models), and the "open" column indicates whether the model weights are accessible for download.

glish data. There are however a growing number of exceptions to this, including a number of Chinese models (Yi, Qwen, GLM, PanGu-α, and Ernie 3.0), one Russian (YaLM) and one Finnish/English model (Poro), as well as a number of models that count as multilingual since they include a number of different languages; examples include Falcon, Mixtral, LLaMA (1 and 2), BLOOM, Grok-1 as well as the commercial models Mistral, Gemini, GPT-4 and Claude. GPT-SW3 is unique in the sense that it is the only model trained specifically on the North Germanic languages (Swedish, Norwegian, Danish, Icelandic and Faroese), and as such it is also the only current LLM built to represent a specific language group.

Most of the early LLMs from 2020 and 2021, such as GPT-3, Jurassic-1, PanGu-α, Ernie 3.0 and Gopher, were not (and are still not) publicly released. However, from 2022 onwards there has been a noticeable shift in release strategies from developers of LLMs, with an increasing number of models being released under various forms of more or less permissive licenses that allow for downloading of model weights (what we refer to as "open" in Table 1). GPT-SW3 is also publicly released under a permissive license. Section 8 provides a more detailed discussion about our considerations regarding release strategies.

3. Data

The arguably most challenging aspect of building an LLM for a (set of) smaller languages is finding sufficient amounts of text data with sufficient quality and variety. Since there are no readily available large data collections for LLM pretraining in the North Germanic languages, we compiled our own training data, which we call *The Nordic Pile* (Öhman et al., 2023).

Our training data consists of text data collected

	Swedish	English	Norwegian	Danish	Icelandic	Other	Code	Total
Articles	16.49 GB	173.52 GB	0.01 GB	0.19 GB				190.21 GB
Books	1.15 GB	94.14 GB	0.04 GB	0.06 GB				95.39 GB
Conversational	65.61 GB	81.67 GB	0.57 GB	2.84 GB	0.07 GB	0.01 GB		150.77 GB
Math	4.58 GB	4.98 GB	0.01 GB	0.01 GB		0.19 GB		9.77 GB
Miscellaneous	28.85 GB	56.31 GB	48.48 GB	13.85 GB	10.26 GB	1.8 GB		159.55 GB
Web CC	188.94 GB	60.36 GB	90 GB	111.33 GB	8.79 GB	2.05 GB		461.47 GB
Web Sources	7.83 GB	0.61 GB	0.03 GB	1.85 GB				10.32 GB
Wikipedia	1.03 GB	14.77 GB	0.48 GB	0.38 GB	0.05 GB			16.71 GB
Code							114.5 GB	114.5 GB
Total	314.48 GB	486.36 GB	139.62 GB	130.51 GB	19.17 GB	4.05 GB	114.5 GB	1,208.69 GB

Table 2: Data sizes for each language and category after cleaning and processing.

from various open general data sources, such as MC4 (Xue et al., 2021), OSCAR (Suárez et al., 2019; Ortiz Suárez et al., 2020), OPUS (Tiedemann and Nygaard, 2004), Wikipedia and The Pile (Gao et al., 2021a), as well as language-specific corpora such as the Norwegian Colossal Corpus (Kummervold et al., 2021), the Danish and Icelandic Gigaword corpora (Strømberg-Derczynski et al., 2021; Barkarson et al., 2022), and various data repositories, websites and discussion forums in Swedish. We also include a set of four different programming languages from the CodeParrot[3] collection (Python, JavaScript, SQL and Shell script). Table 2 summarizes the various data categories across the various languages included in the training data.

We performed several steps of data processing on the collected data, including normalization, quality filtering and deduplication (both exact and fuzzy). The normalization takes care of non-printing characters, and normalizes whitespace and Unicode characters. The quality filtering applies a set of heuristics inspired by Gopher and ROOTS (Rae et al., 2021; Laurençon et al., 2022), and the fuzzy deduplication utilizes MinHash LSH (Broder, 1997). Our training data, processing steps, and arguments for selection and filtering of sources is described in more detail in a separate publication (Öhman et al., 2023).

We weight the different languages and categories (cf. Table 2) in such a way that the composition of the training data changes, while its total size stays the same. Note that this implies that some data are used multiple times while other data are discarded. More details can be found in App. A. After weighting, we end up with the following distribution of data in terms of languages:

- Swedish: 35.3%
- English: 23.4%
- Norwegian: 17.3%
- Danish: 14.8%

- Icelandic: 2.7%
- Code: 6.5%

4. Tokenizer

We employed the SentencePiece library (Kudo and Richardson, 2018) to train a Byte-Pair Encoding (Sennrich et al., 2016) tokenizer on a representative 1% sample of the model training data. The tokenizer has a vocabulary size of 64,000. Our reason for using a slightly larger vocabulary size compared to other LLMs (e.g. GPT-3, OPT, and GPT-NeoX have a vocabulary size of 50k tokens, while LLaMA only uses 32k tokens in the vocabulary) is that we want to improve the performance of the smaller languages included in our data, such as Icelandic.

Our tokenizer works without explicit pretokenization. However, it splits digits and uses SentencePiece's dummy prefix and the byte fallback feature. We also added repeated whitespace tokens (Black et al., 2022b) and special code tokens like $<|python|>$ to the tokenizer's vocabulary, in order to improve the way code data is handled. Note that the special code tokens are present in the code data as well. We describe the tokenizer's features, training and evaluation in more detail in a separate paper (Stollenwerk, 2023).

After tokenization, our training data consists of around 320B tokens.

5. Training

We trained our models on 160 40GB A100 GPUs using the Nemo Megatron framework (Narayanan et al., 2021). We trained models of increasing size, starting out with the smaller models. This strategy was employed to identify problems with the pretraining procedure early on, before training the larger models.[4]

---

[4]One such problem did occur: In the initial runs of the small models, we assumed that the tokenization and binarization script would delimit documents by end-of-text-tokens. It did no such thing, which lead to frequent and

Size	lr	batch	heads	depth	emb. dim.
126M	3e-4	256	12	12	768
356M	3e-4	256	16	24	1,024
1.3B	2e-4	512	32	24	2,048
6.7B	1.2e-4	1,000	32	32	4,096
20B	1.4e-4	1,920	48	44	6,144
40B	1.1e-4	1,920	64	48	8,192

Table 3: Hyperparameters used for the GPT-SW3 models of different sizes. All models have the same vocabulary size (64,000) and sequence length (2,048). The number of model parameters are denoted by Size, while lr corresponds to the maximum learning rate.
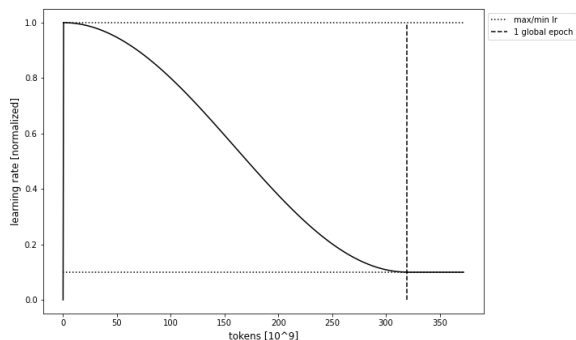
Figure 1: Normalized learning rate schedule. The maxima of the learning rate are given in Table 3.

Table 3 shows the most important hyperparameters for the various model sizes. All models have the same vocabulary size (64,000) and sequence length (2,048). The feed-forward dimension is always four times the embedding dimension. The models were trained using packing, meaning that each sample in a batch can consist of multiple documents[5] delimited by end-of-text-tokens[6]. We did *not* use attention-masking between documents.

The learning rate schedule we employed is a function of the amount of data, as visualized in Figure 1. It is the same for all model sizes apart from a global factor, the maximum learning rate listed in Table 3. The training process starts with a short warm-up period that amounts to 0.5B tokens, during which the learning rate is increased from 0 to its maximum. Afterwards, we use a cosine decay to the minimum learning rate ($\frac{1}{10}$ of the maximum

---

unexpected context-switching during generation. Thankfully, this was discovered and remedied early on due to the training strategy.

[5]We have ∼4 documents per sample on average.

[6]Due to the nature of dataloading in Nemo Megatron, this also means that documents belonging to the same data sample also belong to the same dataset. Combined with cross-document attention, this could have a slightly adverse effect on the end-of-text token as documents across end-of-text boundaries are not completely independent.

Size	GPUs	FLOP/s	Utilization
126M	64	1.71×10^{15}	8.58%
356M	32	1.57×10^{15}	15.69%
1.3B	128	5.21×10^{15}	13.06%
6.7B	160	8.63×10^{15}	17.28%
20B	160	1.89×10^{16}	37.91%
40B	160	1.96×10^{16}	39.23%

Table 4: Achieved Model FLOP/s and Utilization (w.r.t. peak theoretical FLOP/s) for the various model sizes during a single job.

Figure 2: Validation loss during training.

learning rate) for another 319.3B tokens. Finally, training is continued at a constant learning rate until up to 372.2B tokens are reached.

Table 4 shows the model FLOP/s we achieved for the various model sizes, as well as the utilization w.r.t. peak theoretical FLOP/s. For the larger models, the utilization numbers are comparable to those reported by NVIDIA in Narayanan et al. (2021) whereas the smaller models show much poorer utilization. We believe this can be attributed to over-parallelization of the smaller models.

The validation loss during training can be seen in Figure 2. As expected, the larger models reach lower validation loss, and largely follow the expected scaling behavior (see Appendix B for more details). Contrary to some other works (Zhang et al., 2022), we did not experience any divergence during training, but we did observe the occasional gradient spike (with no catastrophic long-term effect).

5.1. Energy consumption

We estimate that the total compute budget for our training runs is something like 560k GPU hours. This is obviously a very rough estimate, and likely to be somewhat on the high end. The average carbon intensity in Sweden, where the models were trained, is estimated to be around 10 grams of carbon dioxide per kilowatt-hour (gCO_2/KWh).[7] Using the `ML CO₂ IMPACT calculator`[8] we calculate our total carbon emissions to be roughly 14,462 kg CO_2. This is approximately on the same level of

---

[7]https://bit.ly/4anuJyK
[8]mlco2.github.io/

carbon emissions as is generated by manufacturing *one* 80 kWh lithium-ion battery used in electric cars.[9] Considering the relatively low carbon intensity of the power system used to train our models, our carbon footprint is significantly lower than that of other similar models (see e.g. the emissions estimated for the BLOOM model (Luccioni et al., 2022)).

6. Instruction finetuning

Due to the popularity and effectiveness of instruction-tuned models such as ChatGPT, we also produce a set of instruction-tuned models. The models were fine-tuned using instruction tuning (Ouyang et al., 2022) data from multiple sources: Open Assistant[10] (Köpf et al., 2023), The Open Instruction Generalist (OIG) dataset[11], Dolly[12], and a dataset compiled specifically for this study based on FASS (The Swedish pharmaceutical formulary, *Farmaceutiska Specialiteter i Sverige*).

For the OIG data, we selected high-quality subsets, which encompassed a wide range of topics and dialog styles. The datasets selected were abstract infill, HC3 human, SODA dialog, CHIP2, image prompts instructions, SQLv1, conversation FinQA, MathQA FLANv2 Kojma COT, SQLv2, CUAD, NI, SQuAD v2, essays, OpenAI Summarize TLDR, SQuAD v2 more negative, grade school math instructions, Rallio SODA upgraded 2,048, and unnatural instructions. The rest of the datasets from OIG were discarded, leaving us with a considerably smaller data set than the original OIG.

We formatted the instruction data into a unified turn-based format, where an initial user query is followed by an assistant response, which in turn is (potentially) followed by a follow up user query, and so on. This turn-based query-response format was formatted in two ways[13]:

- An unrolled format, where the query-response-query-turns are simply delimited by double newlines:

      Query

      Response

      ...

- An explicit chat format inspired by the chatml format:[14]

      <eos><bos>User: Query
      <bos>Assistant: Response
      <bos>...

Where `<eos>` is the special document-delimiter token used during pretraining, and `<bos>` is a special turn-delimiter token only used during instruction-finetuning.

In both cases, we employed stochastic merging of independent conversations using a simple concatenation strategy: Sample a *number of samples* N according to a geometric distribution, randomly sample N conversations from the dataset, and let the concatenation of these N conversations be your new datapoint. This was done to improve the models context-switching capabilities.

To accommodate our multilingual focus, the OpenAssistant and Dolly data sets were translated. The OpenAssistant data set was translated from English to Swedish, Danish, Norwegian, and Icelandic, while the Dolly data set was translated from English to Swedish and Danish. The translations were done with the GPT-SW3 base models.

The fine-tuning process was applied consistently across models of different scales, including 356M, 1.3B, 6.7B, and 20B parameters. We used a sequence length of 2,048, a batch size of 160, and an initial learning rate of 2×10^{-5}, which we gradually reduced with cosine decay to a minimum of 2×10^{-6} over the course of 2,069 global steps. This approach consumed a total of 331,040 samples, with a warm-up period spanning 360 steps.

7. Evaluation

Since we currently lack suitable evaluation benchmarks for generative language models in the North Germanic languages, we use language modeling perplexity on a set of held-out data to compare our models. We use character length normalization (Cotterell et al., 2018; Mielke, 2019) rather than token length for calculating perplexity formula, since token length favours tokenizers that use more tokens per sentence. We thus calculate perplexity as:

$$PPL_c(X) = \exp\left\{ -\frac{1}{\mathbf{c}} \sum_{i=1}^{\mathbf{t}} \log p(T_i|T_{<i}) \right\}$$

$\mathbf{c}$ = Character length of X

T = Tokenization of X (1)

$\mathbf{t}$ = Token length of T

---

[13] This means that each conversation occurs twice in the final training data, once as chat and once in the unrolled format.

Task	0-shot						5-shot					
	126M	356M	1.3B	6.7B	20B	40B	126M	356M	1.3B	6.7B	20B	40B
ANLI Round 1	0.336	0.298	0.315	0.337	0.322	**0.368**	0.334	0.313	0.332	0.317	0.330	**0.350**
ANLI Round 2	0.317	0.338	0.345	0.333	0.343	**0.358**	0.341	**0.359**	0.340	0.332	0.333	0.350
ANLI Round 3	0.322	0.325	0.311	0.332	0.333	**0.391**	0.330	0.319	0.336	0.330	0.343	**0.364**
WSC	0.365	0.365	0.365	0.413	0.394	**0.548**	0.365	0.365	0.394	0.365	**0.519**	0.423
HellaSwag	0.279	0.322	0.393	0.457	0.502	**0.532**	0.280	0.321	0.387	0.452	0.504	**0.532**
Winogrande	0.493	0.517	0.571	0.617	0.632	**0.656**	0.522	0.527	0.557	0.607	0.657	**0.674**
PIQA	0.584	0.642	0.707	0.735	0.768	**0.772**	0.600	0.646	0.708	0.739	0.765	**0.776**
ARC (Easy)	0.388	0.408	0.549	0.609	**0.692**	0.687	0.412	0.473	0.588	0.647	0.707	**0.718**
ARC (Cha.)	0.204	0.200	0.253	0.294	0.352	**0.368**	0.191	0.213	0.276	0.312	0.360	**0.387**
OpenBookQA	0.140	0.186	0.214	0.220	0.268	**0.274**	0.148	0.188	0.228	0.260	0.240	**0.300**
HeadQA	0.224	0.236	0.266	0.278	**0.308**	0.302	0.227	0.243	0.273	0.295	0.233	**0.330**
Average	0.332	0.349	0.390	0.420	0.447	**0.478**	0.341	0.361	0.402	0.424	0.454	**0.473**

Table 5: LM Evaluation Harness accuracy scores of GPT-SW3 in 0-shot setting (left) and 5-shot setting (right). The best performance for each task and setting is marked in boldface.

Model	SE	DA	NO	EN
GPT-SW3 40B	**1.9240**	**1.8698**	**1.9270**	1.9660
GPT-SW3 20B	1.9458	1.8932	1.9491	1.9928
GPT-SW3 6.7B	1.9781	1.9229	1.9795	2.0152
GPT-SW3 1.3B	2.0665	2.0192	2.0741	2.1166
GPT-SW3 356M	2.1973	2.1568	2.2130	2.2477
GPT-SW3 126M	2.3748	2.3455	2.3992	2.4297
GPT-NeoX 20B	2.3807	2.3378	2.4245	1.9377
Falcon 40B	2.0194	2.2379	2.2705	**1.8152**
Falcon 7B	2.6546	2.7355	2.7740	1.8743
Falcon-RW 1B	3.7672	3.7187	3.7806	1.9765

Table 6: Evaluation of perplexity normalized on characters on held-out data for Swedish, Danish, Norwegian and English. The best score per language is marked in boldface.

Table 6 shows the perplexity scores for GPT-SW3 in comparison with GPT-NeoX (20B) and the recent Falcon models (1B, 7B and 40B) which have been trained on a small amount of Swedish data (1B tokens). It is obvious, and perhaps not very surprising, that GPT-SW3 reach the lowest language modeling perplexity on Swedish, Danish and Norwegian data, and that larger models reach lower perplexity.

The fact that GPT-SW3 has been trained on English data, and seems to perform well w.r.t. language modeling perplexity, suggests that we can also take advantage of the English-language Language Model Evaluation Harness (Gao et al., 2021b) to benchmark our models. The LM Evaluation Harness framework contains a large number (200+) of different evaluation tasks. We select a small subset of these to benchmark our models (ANLI, WSC, HellaSwag, Winogrande, PIQA, ARC, OpenBookQA, and HeadQA). Table 5 shows the results of GPT-SW3 in a 0-shot setting (left side of the table) and 5-shot setting (right side of the table). Unsurprisingly, the larger models perform better, with the 40B model performing best overall.

Table 7 shows a comparison between two of our base models and their respective instruction-tuned variants in both 0-shot and 5-shot setting. The general tendency is (perhaps unsurprisingly) that instruction-tuning is beneficial for the models when applied to the LM Harness tasks. The 6.7B instruction-tuned model even outperforms the 20B base model on average on LM Harness, and the 20B instruction-tuned version approaches the performance of the 40B model.

Table 8 shows a comparison between GPT-SW3 40B, GPT-NeoX (20B), and GPT-3 DaVinci (presumably 175B) on LM Harness in a 0-shot setting. The models perform more or less comparably over all tests, with DaVinci outperforming GPT-NeoX and GPT-SW3 in 8 out of 13 tests, GPT-SW3 outperforming the other in 5 tests, and GPT-NeoX outperforming the others in only one test. It should be noted that these models are strictly not comparable due to significant differences in training data and parameter count; DaVinci is by far the largest of these models with (presumably) 175B parameters compared to 40B for GPT-SW3 and 20B for GPT-NeoX. On the other hand, GPT-SW3 has been trained on significantly less English data than the other models, but still performs comparably on these tests.

8. Release plan

As we touched upon in Section 2, it is not obvious that new LLMs are released openly, and different developers have opted for different release strategies. Some opt for a completely open release where the weights of the model can be downloaded freely and the user is permitted to both modify and redistribute the weights, as well as to integrate the model in various types of applications, both academic and commercial. Others opt to not share the model weights at all, due to reasons such as commercial advantage, concerns about the potential for misuse, or legal restrictions relating to, e.g., the General Data Protection Regulation (GDPR). Solaiman (2023) provides a good discussion and

Task	0-shot				5-shot			
	6.7B	6.7B-instruct	20B	20B-instruct	6.7B	6.7B-instruct	20B	20B-instruct
ANLI Round 1	**0.337**	0.329	0.322	**0.372**	**0.317**	0.309	0.330	**0.328**
ANLI Round 2	0.333	**0.376**	0.343	**0.381**	0.332	**0.341**	0.333	**0.359**
ANLI Round 3	0.332	**0.361**	0.333	**0.378**	0.330	**0.331**	0.343	**0.372**
WSC	**0.414**	0.385	**0.394**	0.365	0.365	**0.490**	**0.519**	0.375
HellaSwag	0.457	**0.503**	0.502	**0.528**	0.452	**0.502**	0.504	**0.527**
Winogrande	**0.617**	**0.617**	0.632	**0.639**	0.607	**0.616**	**0.657**	0.646
PIQA	0.735	**0.765**	**0.768**	0.764	0.739	**0.762**	0.766	**0.773**
ARC (Easy)	0.609	**0.679**	**0.692**	0.677	0.650	**0.686**	**0.707**	0.702
ARC (Cha.)	**0.294**	0.352	0.352	**0.355**	0.312	**0.360**	0.360	**0.382**
OpenBookQA	0.220	**0.274**	**0.268**	0.250	0.260	**0.288**	0.240	**0.282**
HeadQA	0.278	**0.318**	0.309	**0.310**	0.295	**0.334**	0.233	**0.323**
Average	0.421	**0.451**	0.447	**0.457**	0.424	**0.456**	0.454	**0.461**

Table 7: LM Evaluation Harness accuracy scores of our instruct models (6.7B and 20B) compared with their same-size base model counterparts in 0-shot setting (left) and 5-shot setting (right). The best performance for each task and setting is marked in boldface.

	GPT-NeoX	DaVinci	GPT-SW3 40B
ANLI Round 1	0.340	0.363	**0.368**
ANLI Round 2	0.343	**0.375**	0.358
ANLI Round 3	0.354	0.369	**0.391**
WSC	0.500	**0.548**	**0.548**
HellaSwag	0.535	**0.592**	0.532
Winogrande	0.661	**0.699**	0.656
SciQ	0.928	0.949	**0.955**
PIQA	0.779	**0.791**	0.772
ARC (Easy)	0.723	**0.762**	0.687
ARC (Challenge)	0.380	**0.435**	0.368
OpenBookQA	0.290	**0.336**	0.274
LogiQA	0.230	0.227	**0.290**
PROST	**0.296**	0.267	0.263
Average	0.489	**0.516**	0.497

Table 8: Comparison between GPT-NeoX 20B, OpenAIs DaVinci (175B), and GPT-SW3 40B model on LM Harness (0-shot). The best score for each task is marked in boldface.

overview of the complexities involved in defining a suitable release strategy for an LLM.

As a compromise between openness and caution, our release strategy consisted of two phases:

1. An initial *restricted pre-release*, which included manual audit of applications where access was granted to the model weights for organizations and individuals in the Nordic NLP ecosystem that aimed to use the models for research purposes. Usage was also restricted by a slightly modified version of the BigScience Responsible AI License (RAIL).[15] Our intention was to use this pre-release phase for collecting input on model behavior, flaws and limitations in order to be able to make a more informed decision about open release. The restricted pre-release lasted approximately 6 months, and showed no significant flaws or adversarial effects.

2. Due to the positive outcome of the restricted

pre-release phase, we have now released the weights of the GPT-SW3 models openly under a slightly modified version of the Apache 2 license, which allows for modification, redistribution, research and commercialization of the model weights. We believe that an open release strategy is beneficial for value creation, transparency, democratization, and reproducibility.

The GPT-SW3 models are available at: **huggingface.co/AI-Sweden-Models**.

9. Discussion

This paper has detailed the development process for our family of North Germanic LLMs. It is at this point a perfectly reasonable question to ask why we at all should build a native LLM for a set of small languages with limited resources when the dominant LLMs of large corporations already can handle these languages in a reasonable (and often even superior) way. We have several answers to this question.

We believe that there is a desire and need for cultural and linguistic **representativeness** by informed choices and processing of data sources, **transparency** in all design choices and throughout the entire development process, **democratizing** access to natively built LLMs by open or hosted release, and open **validation** of model capacities as well as **utilization** of existing national compute infrastructure. Perhaps most importantly, the main goal of the GPT-SW3 initiative has been to give the Nordic research community full access to the weights of a native-language LLM, something that is not currently possible with other existing LLMs.

Developing LLMs for smaller languages is admittedly a challenging endeavor when it comes to data availability, but it also opens up opportunities to select and process data sources in a more informed manner that is guided by considerations of

both population representativeness, application domains and regulatory compliance. We have taken a first step in this direction, but our efforts have been constrained by limited funding and compute resources, something we believe is an all too common situation for many developers in small countries. Our initiative has been made possible by a national collaboration with a number of other organisations that have contributed to the development in various ways, e.g. by providing access to large-scale compute. One major advantage of our initiative is the geographical location of the computer used to train our models, which is connected to a power grid with minimal carbon intensity. As such, our training runs have caused significantly lower carbon emissions than other similar projects that run on more carbon intensive power grids.

We concede that evaluation remains an issue for LLMs built for smaller languages such as the North Germanic ones. We are actively working on evaluation resources and process for Scandinavian LLMs, and we are also running a *validation project*, where stakeholders from all sectors of society validate the models for actual use in real-world applications, which span from relatively simple text generation tasks to more complex decision support functionalities. An important finding so far in the validation project is that there is a tangible need for LLMs that allow for the possibility to modify, fine-tune, and host the models locally. This will likely remain an important factor for LLM adoption, even if the available models are slightly less capable than the leading proprietary models.

We conclude this paper with a short note on risks in relation to LLMs. We think the current debate on the potential for apocalyptic risks incurred by LLMs is poorly nuanced and greatly exaggerated, leading to amplified and unnecessary polarization. We think a more realistic risk is the concentration of power and capital that will inevitably occur when only a small set of companies have the resources and abilities to develop, serve and distribute LLMs. Open and nationally driven LLM initiatives are vital counterparts to such developments. Another realistic risk is inflated expectations that may occur when models are not publicly accessible for validation, modification, and further development. Open development will serve to counteract this risk.

Our position is thus that open initiatives to develop LLMs for smaller languages are important and should be supported rather than hindered by regulation, funding sources, and infrastructure access programs.

Acknowledgements

10. Bibliographical References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Adrien Barbaresi. 2021. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131. Association for Computational Linguistics.

Starkaður Barkarson, Steinþór Steingrímsson, and Hildur Hafsteinsdóttir. 2022. Evolving large text corpora: Four versions of the Icelandic Gigaword

corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2371–2381, Marseille, France. European Language Resources Association.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The Pushshift Reddit Dataset. volume 14, pages 830–839. AAAI.

BigScience. 2022. Bigscience language open-science open-access multilingual (BLOOM) language model. international. https://huggingface.co/bigscience/bloom.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022a. GPT-NeoX-20B: An open-source autoregressive language model.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022b. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

A.Z. Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pages 21–29.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with Pathways.

Edith Cohen. 2016. *Min-Hash Sketches*, pages 1282–1287. Springer New York, New York, NY.

Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541, New Orleans, Louisiana. Association for Computational Linguistics.

Ariel Ekgren, Amaru Cuba Gyllensten, Evangelia Gogoulou, Alice Heiman, Severine Verlinden, Joey Öhman, Fredrik Carlsson, and Magnus Sahlgren. 2022a. Lessons learned from GPT-SW3: Building the first large-scale generative language model for Swedish. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3509–3518, Marseille, France. European Language Resources Association.

Ariel Ekgren, Amaru Cuba Gyllensten, Evangelia Gogoulou, Alice Heiman, Severine Verlinden, Joey Öhman, Fredrik Carlsson, and Magnus Sahlgren. 2022b. Lessons learned from GPT-SW3: Building the first large-scale generative language model for Swedish. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3509–3518, Marseille, France. European Language Resources Association.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima,

Shawn Presser, and Connor Leahy. 2021a. The Pile: An 800GB dataset of diverse text for language modeling.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021b. A framework for few-shot language model evaluation.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. 2022. Scaling laws for neural machine translation. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022a. Training compute-optimal large language models.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. 2022b. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*.

K. Ishiguro. 2021. *Klara and the Sun: A novel*. Knopf Doubleday Publishing Group.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016c. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Taku Kudo and John Richardson. 2018. Sentence-Piece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Link"oping University Electronic Press, Sweden.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. OpenAssistant conversations – democratizing large language model alignment.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gérard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Romero Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Vu Minh Chien, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Ifeoluwa Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Luccioni, and Yacine Jernite. 2022. The bigscience ROOTS corpus: A

1.6TB composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *ACL*.

Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. Technical report, AI21 Labs.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2022. Estimating the carbon footprint of BLOOM, a 176B parameter language model.

Udi Manber and Gene Myers. 1993. Suffix arrays: A new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948.

Sabrina J. Mielke. 2019. Can you compare perplexity across different segmentations?

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. 2021. Efficient large-scale language model training on GPU clusters using Megatron-LM. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John F. J. Mellor, Irina Higgins, Antonia Creswell, Nathan McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, L. Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, N. K. Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Tobias Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew G. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem W. Ayoub, Jeff Stanway, L. L. Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training Gopher. *ArXiv*, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Magnus Sahlgren, Fredrik Carlsson, Fredrik Olsson, and Love Börjeson. 2021. It's basically the same language anyway: the case for a Nordic language model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 367–372, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared

Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model.

Irene Solaiman. 2023. The gradient of generative AI release: Methods and considerations.

Felix Stollenwerk. 2023. GPT-SW3 tokenizer.

Leon Strømberg-Derczynski, Manuel Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henrichsen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm, and Daniel Varab. 2021. The Danish Gigaword corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 413–421, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9–16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science.

MosaicML NLP Team. 2023. Introducing mpt-30b: Raising the bar for open-source foundation models. Accessed: 2023-06-22.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. LaMDA: Language models for dialog applications.

Jörg Tiedemann and Lars Nygaard. 2004. The OPUS corpus - parallel and free: `http://logos.uio.no/opus`. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Shuohuan Wang, Yu Sun, Yang Xiang, Zhihua Wu, Siyu Ding, Weibao Gong, Shikun Feng, Junyuan Shang, Yanbin Zhao, Chao Pang, Jiaxiang Liu, Xuyi Chen, Yuxiang Lu, Weixin Liu, Xi Wang, Yangfan Bai, Qiuliang Chen, Li Zhao, Shiyong Li, Peng Sun, Dianhai Yu, Yanjun Ma, Hao Tian, Hua Wu, Tian Wu, Wei Zeng, Ge Li, Wen Gao, and Haifeng Wang. 2021. ERNIE 3.0 Titan: Exploring larger-scale knowledge enhanced pre-training for language understanding and generation.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2022. GLM-130B: An open bilingual pre-trained model.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, Chen Li, Ziyan Gong, Yifan Yao, Xinjing Huang, Jun Wang, Jianfeng Yu, Qi Guo, Yue Yu, Yan Zhang,

Jin Wang, Hengtao Tao, Dasen Yan, Zexuan Yi, Fang Peng, Fangqing Jiang, Han Zhang, Lingfeng Deng, Yehong Zhang, Zhe Lin, Chao Zhang, Shaojie Zhang, Mingyue Guo, Shanzhi Gu, Gaojun Fan, Yaowei Wang, Xuefeng Jin, Qun Liu, and Yonghong Tian. 2021. PanGu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models.

Joey Öhman, Severine Verlinden, Ariel Ekgren, Amaru Cuba Gyllensten, Tim Isbister, Evangelia Gogoulou, Fredrik Carlsson, and Magnus Sahlgren. 2023. The nordic pile: A 1.2tb nordic dataset for language modeling.

A. Data Weighting

We present some details regarding the weighting of the training data (see Sec. 3). First, a target distribution in terms of languages and categories is defined. This means that we specify the fraction of the total dataset that should correspond to a given language and category. Our choices are listed in Table 9.

In order to obtain this target distribution, the datasets need to be weighted, i.e. either downsampled (in case there is more data available than wanted) or upsampled (in case there is less data available than wanted). We do this in such a way that the total amount of training data remains constant at 320B tokens. Table 10 lists the number of epochs for each individual dataset needed to achieve these conditions. Note that English datasets are mostly downsampled, while the North Germanic languages are upsampled.

B. Scaling Analysis

Scaling laws describe how the upstream or downstream performance of LLMs depend on the model size N and dataset size D. Hoffmann et al. (2022b) showed that the loss for their family of monolingual, English models can accurately be described by the functional form

$$L(N, D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta} \qquad (2)$$

with fit parameters $E = 1.69$, $A = 406.4$, $B = 410.7$, $\alpha = 0.34$ and $\beta = 0.28$. This assumes that the learning rate follows the schedule depicted in Figure 1 for all dataset sizes D. For this reason, we can only apply the above scaling law for the dataset size $D \approx 320$B tokens (cf. Sec. 5). In that case, Eq. (2) reduces to

$$\begin{aligned} \widetilde{L}(N) &:= L(N, D = 320\text{B}) \\ &= \widetilde{E} + \frac{A}{N^\alpha} \end{aligned} \qquad (3)$$

with $\widetilde{E} := E + B/(320 \cdot 10^9)^\beta = 1.94$.

In Figure 3, we show the validation loss $\widetilde{L}(N)$ for our models as a function of the model size. Note that the loss for the 20B parameter model is exceptionally large. A comparison with Figure 2 reveals that the learning curve for this very model size displays exceptional behaviour around $D \approx 320$B tokens ≈ 156M samples. We thus treat it as an anomaly and exclude it from the fit to our data. In that case, our model's scaling behaviour can accurately be described by the functional form of Eq. (3), with the fit parameters

$$\begin{aligned} \widetilde{E}_{\text{GPT-SW3}} &= 1.942 \pm 0.002 & (4) \\ A_{\text{GPT-SW3}} &= 702.6 \pm 15.2 & (5) \\ \alpha_{\text{GPT-SW3}} &= 0.348 \pm 0.001 & (6) \end{aligned}$$
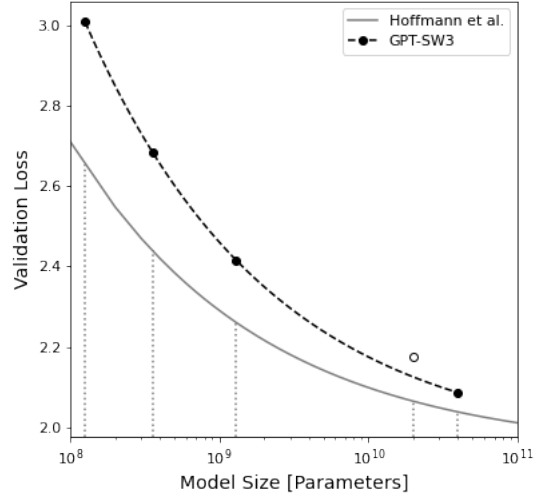

Figure 3: Scaling behaviour of GPT-SW3. The validation loss is shown as a function of the model size, while the dataset size is kept constant at 320B tokens for all models. The 20B parameter model (empty circle) is excluded from the fit (dashed curve). The gray, solid curve represents the scaling law from Hoffmann et al. (2022b).

Note that while $\widetilde{E}_{\text{GPT-SW3}}$ and $\alpha_{\text{GPT-SW3}}$ are very much in accordance with the results from Hoffmann et al. (2022b), $A_{\text{GPT-SW3}} \gg A$ deviates significantly from its counterpart. Whether this is can be attributed to our multilingual setting or has other causes is an interesting research question which we leave for future work.

	Swedish	English	Norwegian	Danish	Icelandic	Code	Total
Articles	2.27	1.96		0.03			4.25
Books	0.12	5.78					5.90
Conversational	10.34	6.19	0.11	0.49	0.03		17.17
Math	1.59	0.70					2.29
Miscellaneous	4.09	3.33	9.56	4.79	1.97		23.73
Web CC	15.49	1.87	7.54	9.11	0.72		34.73
Web Sources	1.10			0.23			1.34
Wikipedia	0.29	3.59	0.12	0.10	0.01		4.11
Code						6.48	6.48
Total	35.30	23.41	17.33	14.75	2.73	6.48	100.0

Table 9: Target distribution in terms of languages and categories. The numbers denote the fraction of the total dataset in percent. Empty cells correspond to non-existing datasets, equivalent to 0. Compare to Table 2.

	Swedish	English	Norwegian	Danish	Icelandic	Code
Articles	1.90	0.15		1.90		
Books	2.11	0.84				
Conversational	2.11	0.84	2.11	2.11	2.11	
Math	1.69	0.67				
Miscellaneous	2.11	0.84	2.11	2.11	2.11	
Web CC	1.05	0.42	1.05	1.05	1.05	
Web Sources	1.69			1.69		
Wikipedia	3.16	3.16	3.16	3.16	3.16	
Code						0.74

Table 10: Epochs needed in order to achieve the target distribution (see Table 9) while keeping the total amount of data constant. Empty cells correspond to non-existing datasets, equivalent to 0.