Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4275–4284
Marseille, 20-25 June 2022

PAGnol: An Extra-Large French Generative Model

Julien Launay[*1,2] **E.L. Tommasone**[*1] **Baptiste Pannier**[1] **François Boniface**[†]
Amélie Chatelain[1] **Alessandro Cappelli**[1] **Iacopo Poli**[1] **Djamé Seddah**[3]
[1] LightOn [2] LPENS, École Normale Supérieure [3] Inria, Paris
{julien,luca,baptiste,amelie,alessandro,iacopo}@lighton.ai
djame.seddah@inria.fr

Abstract

Access to large pre-trained models of varied architectures, in many different languages, is central to the democratization of NLP. We introduce PAGnol, a collection of French GPT models. Using scaling laws, we efficiently train PAGnol-XL (1.5B parameters) with the same computational budget as CamemBERT, a model 13 times smaller. PAGnol-XL is the largest model trained from scratch for the French language. We plan to train increasingly large and performing versions of PAGnol, exploring the capabilities of French extreme-scale models. For this first release, we focus on the pre-training and scaling calculations underlining PAGnol. We fit a scaling law for compute for the French language, and compare it with its English counterpart. We find the pre-training dataset significantly conditions the quality of the outputs, with common datasets such as OSCAR leading to low-quality and offensive text. We evaluate our models on discriminative and generative tasks in French, comparing to other state-of-the-art French and multilingual models, and reaching the state of the art in the abstract summarization task. Our research was conducted on the public GENCI *Jean Zay* supercomputer, and our models up to the Large are publicly available.

Keywords: generative pretrained transformer, scaling laws, French language

1. Introduction

Large pre-trained language models are the workhorses of modern Natural Language Processing (NLP). The use of scalable and efficient attention-based Transformers (Vaswani et al., 2017), rather than recurrent neural networks, has enabled increasingly large and capable models. Through self-supervised learning, these models learn contextual word embeddings, building a general representation of language. After this *pre-training* they can be *fine-tuned* to target specific tasks (e.g. classification, parsing, summarization).

Three approaches dominate the field: (1) causal autoregressive decoder-only models, such as GPT (Radford et al., 2018), learning from a general language modelling tasks; (2) bidirectional encoder-only models, such as BERT (Devlin et al., 2018), learning from masked language modelling; (3) sequence-to-sequence models, such as BART (Lewis et al., 2019) or T5 (Raffel et al., 2020), combining both a bidirectional encoder and an autoregressive decoder, learning from a language denoising task. Encoder-only and sequence-to-sequence models excel in language understanding tasks, and have shadowed autoregressive models as a lesser option.

Autoregressive models have been shown to predictably benefit from increased size (Kaplan et al., 2020; Henighan et al., 2020). Scaling laws establish a direct relationship between model size and end-task performance, justifying the training of increasingly large models (Brown et al., 2020; Zeng et al., 2021; Kim et al., 2021; Wei et al., 2021). These laws can also inform design decisions, helping practitioners use their available compute budget optimally. Larger models are more sample and compute efficient: with a given compute budget, it is preferable to train a larger model significantly short of convergence than to train a smaller model to convergence. Furthermore, at extreme-scale, such as the 175 billion parameters of GPT-3 (Brown et al., 2020), autoregressive models exhibit unique few-shot abilities: they can learn from a few prompted examples, without weight updates. This capability questions the current fine-tuning paradigm. Recent forays into *prompt engineering/tuning* (Li and Liang, 2021; Lester et al., 2021) have even seemingly bridged the gap between few-shot performance and fine-tuning.

Encoder-only (CamemBERT (Martin et al., 2019) and FlauBERT (Le et al., 2019)) and sequence-to-sequence models (BARThez (Eddine et al., 2020)) exist for the French language, and recently a decoder-only model with 1 billions parameters has been made available (Simoulin and Crabbé, 2021). We introduce PAGnol in this family, a collection of four French GPT-like models, and make the following contributions:

- **Largest French model trained from scratch.**[1] We train on CCNet and publicly release four models, with up to 1.5B parameters for PAGnol-XL. At the time of this work, this is the largest non-sparse French language model available, that was trained from scratch, and we plan to explore increasingly large and powerful models in the future.
- **Optimal scaling.** We use scaling laws to inform our training setup, resulting in optimal use of our

[1] A larger model, *Boris* (Müller and Laurent, 2022), was recently made available by a Swiss team, however PAGnol is trained from scratch, while Boris is fine-tuned on French data starting from the English model GPT-J (Wang and Komatsuzaki, 2021).

compute budget. PAGnol-XL is trained with a budget of only 3 PF-days, just as much as the 13 times smaller CamemBERT. From our collection of models, we adjust scaling laws for the French language.

- **Dataset suitability.** We highlight the importance of proper dataset pre-processing when training generative autoregressive models. While OSCAR has been relied on for French encoder-only models, we find it is not suited to PAGnol, leading to low-quality offensive outputs.

- **End-task performance.** We evaluate on discriminative (FLUE) and generative tasks (question answering on FQuAD and summarization with OrangeSum) in the fine-tuning and prompt tuning regimes. We establish a new state of the art for summarization in French on OrangeSum.

2. Related work

Language models. The design and training of neural language models able to create and process word embeddings is the cornerstone of modern NLP. Early on, self-supervised learning was identified as an efficient and scalable way to train such models. The use of deeper and more complex neural architectures enabled going from static embeddings (*word2vec* (Mikolov et al., 2013), *GloVe* (Pennington et al., 2014)) to contextual embeddings, allowing models to deal with polysemy. Although approaches such as ELMo (Peters et al., 2018) and ULMFiT (Howard and Ruder, 2018) highlighted that learned representations can be transferred across downstream tasks, the poor scalability of RNNs prevented this vision from being fully realized.

By getting rid of the costly and delicate recurrent processing, attention-based Transformers (Vaswani et al., 2017) spurred a wide interest in NLP. GPT (Radford et al., 2018), a decoder-only variant of Transformers, demonstrated large-scale transfer learning from general language modelling to 12 NLU tasks. Along with the rise of easy-to-use libraries, encoder-only BERT (Devlin et al., 2018), relying on masked language modeling, made NLP a commodity – wherein every practitioner could rely on a pre-trained language model and fine-tune it cheaply to a task of interest. BERT models are limited by their ability to only "fill-in-the-gap" for a span of words: this forbids their use in generative tasks (e.g. summarization).

With sequence to sequence models and pre-training through denoising tasks, the original architecture of Transformers made a comeback with BART (Lewis et al., 2019), bridging the gap between the generative capabilities of decoder-only models and the downstream task performance of encoder-only models. Through gradually larger and more powerful architectures, state-of-the-art models are approaching human-level performance on many tasks.

Successive generations of GPT models have questioned the current fine-tuning paradigm. GPT-2 (Radford et al., 2019), with 1.5 billion parameters, demonstrated that large language models could tackle entirely new tasks through *few-shot learning*[2]. Without any fine-tuning, from just a few prompted examples, GPT-2 achieved fair performance on a number of complex downstream tasks. Furthering this endeavour, GPT-3 (Brown et al., 2020), with 175 billion parameters, achieved state-of-the-art performance on some tasks, without the need for fine-tuning. This opens new possibilities for low-resources tasks, as well as paths to more natural interactions with these models: recent research suggests the gap between few-shot learning and fine-tuning may even be bridged through so-called prompt programming/tuning (Li and Liang, 2021; Lester et al., 2021).

Scaling laws. More specifically to our setting, neural language models have been shown to predictably benefit from increased scale (Kaplan et al., 2020). Their training dynamics are size-invariant, allowing test loss, parameter count, and dataset size to be correlated through smooth scaling laws. This is in fact true of all GPT-like autoregressive models, even when applied to image, multimodal, or mathematics modeling (Henighan et al., 2020). Gains in autoregressive cross-entropy loss also directly translate to gains in end-task performance after fine-tuning. As they relate to compute budget, these predictions can be used to inform the training of large models.

Non-English generative models. BERT-like models are now available in a broad number of languages, either as specialized models or as multilingual ones. This is less so the case for generative models, perhaps because of issues in controlling the language used at generation time. For the French language, GPT_{fr} is an autoregressive generative model, and BARThez (Eddine et al., 2020) targets some generative abilities. Smaller-scale efforts exist, such as BelGPT (Louis, 2020), but they are limited to small models. GPT models have been trained for German (Schweter, 2020), Chinese (Zeng et al., 2021), Korean (Kim et al., 2021), Russian (Alexandr et al., 2021), and Arabic (Antoun et al., 2021a; Lakim et al., 2022), among others.

3. Efficient training with scaling laws

Scaling. We use scaling laws to inform the duration of the training of our largest models. Rather than training to convergence, which would be wasteful, we train to optimality, as predicted by the equations provided in (Kaplan et al., 2020). This is akin to what has been done for GPT-3, and this enables us to keep our computational budget in line with that of CamemBERT, a model 13x smaller than PAGnol-XL. We find that training all of our models for a single epoch on the 30 Gi-

[2] In other areas of machine learning, this has been referred to as *zero-shot learning*, as no weight updates are necessary.

gaTokens (GT) of CCNet enables us to reach optimality for the most expensive XL model. Table 3 presents the ratios between the compute budget effectively used and that to optimality (r^C_{opt}) or to convergence (r^C_{conv}). While our small model is trained to convergence, others are trained significantly short of it. We find that our training performance matches nicely with the estimated 2.6 PF-days for the training of GPT_{fr}-LARGE from (Simoulin and Crabbé, 2021).

4. PAGnol

In this section, we describe the data, model, and training specifications for PAGnol. In Table 1, we highlight some of the key differences and similarities with other French models, and in Table 2 we present two multilingual models that we consider in the following.

4.1. Pre-training data

Sourcing. The Common Crawl (CC) project browses and indexes all content available online. It generates 200-300 TiB of data per month (around 5% of which is in French), and constitutes the bulk of most NLP datasets nowadays. We consider in our experiments two datasets based on CommonCrawl data: CCNet (Wenzek et al., 2020) and OSCAR (Ortiz Suárez et al., 2020). Once tokenized, OSCAR contains 33GT and CCNet 32GT. We use CCNet for all our main experiments and released models, and compare with results obtained on OSCAR in Section 5. We validate on the fr-wiki dataset (0.5GT) and French TreeBank (650kT) (Abeillé et al., 2003).

CCNet. CCNet combines the usual fastText (Joulin et al., 2017) pre-processing of CC data with an additional filtering step to select high-quality documents. This filtering is done through a language model trained on Wikipedia, ensuring a text quality similar to that of its articles. We use a version of CCNet identical to the one considered in the CamemBERT paper.

OSCAR. OSCAR uses a fastText classifier to select documents and identify their languages, without any additional filtering. OSCAR is thus more varied, but more "noisy", than CCNet. OSCAR has been used to train other French language models such as CamemBERT.

Tokenization. We use byte-level Byte-Pair Encoding (BPE), with a vocabulary of 50,262 tokens: 256 bytes, 6 special tokens, and 50,000 merges. Paragraphs are separated by an $<$ EOS $>$ token and documents are separated by a $<$ SEP $>$ token. We add a prefix space before tokenization, so that the first word in a paragraph is tokenized in the same way as if it was at any other position. This is similar to the setup of FlauBERT and GPT-2. For the models trained on OSCAR, we use a slightly smaller vocabulary size of 50,000 tokens.

4.2. Model specification

PAGnol is a decoder-only autoregressive transformer. We evaluate four model sizes: small, medium, large, and extra-large, with architectures detailed in Table 3. We use a context size of 1,024 tokens for the S, M and L models. The XL uses a context size of 2,048, the largest at release for a French model. Additionally, we use Rotary Embeddings (Su et al., 2021) in place of Learned Positional Embeddings for the XL model, since they provide much better training stability at the billion parameters regime.

4.3. Pre-training

Training objective. We use an autoregressive language modelling objective, where the model learns to predict the next word in a sentence. To improve efficiency, we always fill the context with as much text at possible, and inform the model about separate documents through the ¡SEP¿ token.

Optimization. We use the Adam optimizer (Kingma and Ba, 2014) with a warmup followed by cosine decay learning rate schedule. We find that proper initialization is key to training stability, and reproduce the setup effectively implemented by Megatron-LM (Shoeybi et al., 2019). We initialize all weights with a normal distribution $\mathcal{N}(0, 0.02)$ and scale the weights under the residual layers by $1/\sqrt{2n_{\text{layers}}}$. We tune hyperparameters over 10k step first, and pick the set with the best train perplexity.

Distributed training. All training runs were performed on the public GENCI supercomputer *Jean Zay*, on nodes with 4x or 8x V100 32GB and a 100Gb/s interconnect. We built our own GPT implementation from scratch in PyTorch (Paszke et al., 2019), leveraging FairScale for distributed training (Baines et al., 2021).
Models up to PAGnol-L can be trained using simple distributed data parallelism (DDP). However, PAGnol-XL does not fit in 32GB of memory. We use optimizer and model state sharding, along with activation checkpointing and CPU offloading to fit the model in memory. This results in a setup similar to ZeRO-3 (Rajbhandari et al., 2021). It is beneficial to train even small models with this setup, as it allows for a larger batch size, and significantly higher GPU throughput.

Perplexity. We report final validation perplexity after 1 epoch over 30GT in Table 4. We use the official 2019 French Wikipedia dumps and the French TreeBank dataset (Abeillé et al., 2003) in its SPMRL instance (Seddah et al., 2013) as our validation sets. Because we are running all models for a single epoch on our data, there are limited risks of overfitting and memorization.

Scaling law for PAGnol models We fit a scaling law with the same functional form of (Kaplan et al., 2020), that is the following power law:

$$\mathcal{L} = \left(\frac{k}{C}\right)^{\alpha}$$

where $\mathcal{L}$ is the validation loss of PAGnol models trained on CCNet, k is a constant, C is the compute in PF-days,

	CamemBERT	FlauBERT	BARThez	GPT_{fr}	PAGnol (ours)
Language	French	French	French	French	French
Parameters	110/335M	138/373M	165M	124M/1B	124M/355M/773M/1.5B
Context	512	512	768	1024	1024/2048
Dataset	OSCAR 33GT/138GB	Custom[a] 13GT/71GB	Custom[a] 66GB	Filtered Common Crawl 1.6/3.11 GT	CCNet 32GT
Tokenization	SentencePiece 32k	BPE 50k	SentencePiece 50k	BPE 50k	BPE 50k
Compute [PF-days]	3/10	$\sim 7/26$[b]	~ 4[b]	?/2.6	0.3/0.7/2/3

Table 1: Model, data, and training setup for PAGnol and other French models. Data size is reported in gigatokens (GT), and compute in PF-days (8.64×10^{19} FLOP). PAGnol is the largest French model. Despite being significantly larger than existing models, its compute cost remains reasonable: as recommended by scaling laws, we train models to optimality, and not to convergence.

[a]FlauBERT and BARThez use a similar pre-training dataset assembling CommonCrawl, NewsCrawl, Wikipedia, and other smaller corpora.

[b]Insufficient data was provided by authors to infer compute budgets properly.

	mBERT	mBART
Language	104 languages	25 languages
Parameters	110M	610M
Context	512	768
Dataset	Wikipedia	CC25 180GT/1369GB (10GT/57GB French)
Tokenization	WordPiece 110k	SentencePiece 250k
Compute [PF-days]	2	~ 60[b]

Table 2: Model, data, and training setup for multilingual models including French that we consider. Data size is reported in gigatokens (GT), and compute in PF-days (8.64×10^{19} FLOP).

PAGnol	n_{params}	n_{layers}	d_{model}	n_{heads}	C [PF-days]	r^C_{conv}	r^C_{opt}
S	124M	12	768	12	0.3	1.3	9.0
M	355M	24	1024	16	0.7	0.5	3.0
L	773M	36	1280	20	2	0.4	2.5
XL	1.5B	48	1600	25	3	0.2	1.3

Table 3: Model and training budgets for PAGnol. All models are trained on a single epoch of our 32GT CCNet-curated data. C is the compute budget used for the training of the model. r^C_{opt} is the ratio between C and C_{opt}, the optimal compute budget derived from scaling laws. r^C_{conv} is the ratio between C and C_{conv}, the compute budget derived from scaling laws to train the model to convergence.

PAGnol	n_{params}	fr-wiki	FTB: whole	(train/val/test)
S	124M	43.38	23.87	(23.90, 24.38, 23.34)
M	355M	37.46	20.90	(20.92, 21.59, 20.46)
L	773M	34.81	19.97	(19.88, 21.19, 20.02)
XL	1.5B	28.85	16.18	(16.11, 16.67, 16.40)

Table 4: Validation perplexity on fr-wiki and on the whole French TreeBank (FTB) for PAGnol models after 1 epoch of training on 30GT of CCNet.

and α is the scaling exponent. The fit is performed in log-log, and constrained to remain under the efficient frontier, using *cvxpy* (Agrawal et al., 2018). We exclude the L and XL models from the fit: due to the HPC environment, the training was performed in multiple splits. At restart, the optimizer state is not necessarily available, generating artefacts in the training and validation curves. Additionally, the use of Rotary Embeddings for the XL model would affect the scaling, and make it incomparable with the English models. We therefore trained two smaller models, an XXS and an XS, following the same architectural decisions of the larger ones, on the same datasets, and used these to fit a scaling law. We find a scaling exponent $\alpha = -0.036$ for the French language, to compare to the -0.050 for the English language from (Kaplan et al., 2020). With the relatively important caveats that we are using different datasets, codebase, and hardware, it appears that French is less compute efficient than English, and that for a same improvement in validation loss, therefore we need to spend more compute for French than for English. The increased morphological complexity of French when compared to English (Seddah et al., 2010) and its, in average, longer sentences could be a factor explaining this discrepancy.

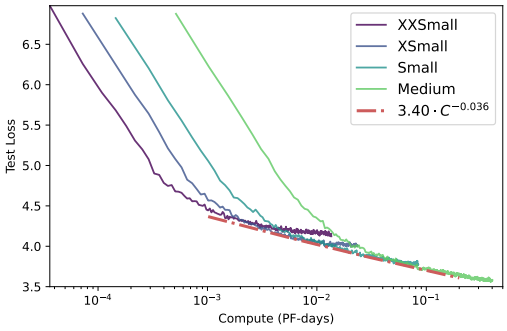

Figure 1: Scaling law relative to the compute for PAGnol models from XXS to M. We do not include L and XL: the interrupted nature of the training due to the HPC environment and the choice of Rotary Embeddings for the XL pollute validation curves with artefacts that negatively affect the quality of the fit.

5. Influence of the pre-training data

Existing French models (CamemBERT, FlauBERT, BARThez) have been trained on datasets based on a simple filtering pipeline. A fastText classifier is used to isolate content in French, deduplication is applied, and noisy content (phone numbers, code, etc.) is removed. While this has been sufficient for pre-training encoder-only models and sequence-to-sequence models, the lack of quality control may be an issue for free-form generation with a decoder-only model such as PAGnol. Moreover, recent work on the OSCAR dataset (used by CamemBERT) has found that it may contain up to 5% of non-linguistic content, or content in the wrong language, and 0.5% of explicit content for the French language (Caswell et al., 2021).

We initially pre-trained PAGnol on the OSCAR dataset, and while experimenting with the model, we observed the model generated offensive and explicit content, even when not prompted for it. For instance, the prompt *Bonjour je suis* (Hello I am) often resulted in pornographic content, despite being rather naive. This motivated our choice to switch to CCNet instead. For research purposes, we release the small and medium models trained on OSCAR. In future iterations of this document, we will provide a more detailed investigation of the content generated by PAGnol-CCNet and by PAGnol-OSCAR.

6. End-task performance

Discriminative tasks: FLUE We evaluate our models on the Sentence Classification task of the FLUE evaluation setup (Le et al., 2019). The task is a binary classification problem on reviews of Books, Music and DVD taken from the Amazon website. Each review is assigned a score from 1 to 5, and then labeled as "negative" if the score is lower than 3 and "positive" otherwise. We also evaluate our models on the paraphrasing and natural language inference tasks (PAWS-X and XNLI). PAWS-X consists in a binary classification task where the model has to identify whether two sentences are semantically equivalent or not. XNLI is instead a 3 class problem where we have to determine if a premise contradicts, entails or neither a given hypothesis. For the training, we add a CLS token at the end of the review (but before the EOS token). We then replace the projector at the end of the model with a linear layer and use the embedding of the CLS token to perform the classification.

Table 5 reports the test accuracy of the best along with a comparison with other French language models. All models are fine-tuned for 6 epochs, except the medium OSCAR and the extra-large CC-100 which were trained respectively for 4 and 5 epochs. For each model, we finetune the learning rate and weight decay in the interval $[10^{-6}, 10^{-4}]$ and $[0, 10^{-3}]$ respectively. For the classification task, we use a cosine annealing scheduler that decays down to $1/10$ of the original learning rate in 5 epochs (3 for the medium OSCAR and 4 for the extra-large CC-100). We additionally checked if adding dropout with $p = 0.1$ could improve the performance. For the PAWS-X and XNLI tasks, we finetune the learning rate in the interval $[10^{-6}, 10^{-4}]$. We use the cosine annealing scheduler down to a learning rate equal to $1/10$ of the original value ($1/5$ for the Small models) over $9/40$ Million tokens respectively. PAWS-X training is over 2 epochs while XNLI training over 1. PAGnol models slightly underperform smaller BERT models, while being better than multilingual alternatives, and their GPT_{fr} counterparts. For PAGnol, performance improves with size but seems to saturate with the XL model, possibly because we had to use a lower batch size to fit on the hardware for fine-tuning. Additionally, while the generation quality of models trained on OSCAR is noticeably worse, they perform as well or better than the corresponding models trained on CCNet on these discriminative tasks.

Generative task: FQuAD (d'Hoffschmidt et al., 2020) is a native French question answering dataset, comprising more than 25.000 questions fabricated by higher education students from a set of Wikipedia articles, following the same philosophy as the English dataset SQuAD (Rajpurkar et al., 2018). Given a document d_i, a question q_i and the corresponding answer a_i, the Question Answering task is casted into this format:

$$\text{"}\{d_i\}\ \textit{Question:}\ \{q_i\}\ \textit{Réponse:}\ \{a_i\}\text{"}$$

where *Réponse* corresponds to *Answer* in French.

Given this input format, in a setup similar to pre-training, the likelihood of the sequence corresponding to the answer is maximized using the cross entropy loss on the tokens corresponding to the answer. We use the Adam optimizer and finetune the learning rate and weight decay in the interval $[10^{-6}, 10^{-4}]$ and $[0, 10^{-3}]$. The different models were trained for 2 epochs. As noted by Radford et al. (2019), the performance of autoregressive models is still worse than question answering systems based on masked language models. Indeed, we evaluated the finetuning of OpenAI GPT-2 small and medium on SQuAD, and obtained exact match (EM) and F1 scores in the same range of PAGnol on FQuAD (Table 7).

Generative task: OrangeSum (Eddine et al., 2020) is a summarization dataset, considered to be the French equivalent of the XSum (Narayan et al., 2018). It is an abstractive dataset containing summary of news article from the "Orange Actu" website. Each article comes with a professionally-written title and abstract. Hence, the dataset includes two tasks: OrangeSum Title and OrangeSum Abstract. We evaluate PAGnol on the latter.

Similarly to our setup for question answering, given a news article n_i and an abstract a_i, we cast the summa-

Model	Parameters	Books	Music	DVD	PAWS-X	XNLI
MultiFiT	Not Specified	91.25	89.55	93.40	-	-
mBERT	110/340 M*	86.15	86.90	86.65	89.30	76.9
mBART	610 M	93.40	93.13	93.10	89.70	81.07
BARThez	216 M	94.47	94.97	93.17	88.90	80.73
CamemBERT-BASE	110 M	92.30	94.85	93.00	90.14	81.20
CamemBERT-LARGE	335 M	**95.47**	**96.00**	95.37	**91.83**	**85.33**
Flaubert-BASE	138 M	93.10	92.45	94.10	89.49	80.60
Flaubert-LARGE	373 M	95.00	94.10	**95.85**	89.34	83.40
GPT_{fr}-BASE	124 M	88.30	86.90	89.30	83.30	75.60
GPT_{fr}-LARGE	1 B	91.60	91.40	92.60	86.30	77.90
PAGnol-S $_{OSCAR}$	124 M	92.05	92.60	91.70	84.19	76.10
PAGnol-M $_{OSCAR}$	355 M	94.40	94.90	94.30	87.44	79.46
PAGnol-S $_{CC\text{-}100}$	124 M	92.00	93.00	91.65	87.19	75.67
PAGnol-M $_{CC\text{-}100}$	355 M	94.40	95.20	93.70	89.14	79.00
PAGnol-L $_{CC\text{-}100}$	773 M	94.65	95.25	94.00	90.70	81.48
PAGnol-XL $_{CC\text{-}100}$	1.5 B	94.65	95.35	94.18	89.47	81.83

Table 5: Results on the FLUE Benchmark including classification (Books, Music, DVD), paraphrasing (PAWS-X) and natural language inference (XNLI) tasks. The best overall results are highlighted in **bold**, and the best results for GPT models are underlined.

Model	EM	F1
CamemBERT-LARGE	**82.1**	**92.2**
CamemBERT-BASE	78.4	88.4
PAGnol-S $_{OSCAR}$	31.7	52.8
PAGnol-M $_{OSCAR}$	37.1	59.4
PAGnol-S $_{CC\text{-}100}$	33.7	56.0
PAGnol-M $_{CC\text{-}100}$	36.8	59.0
PAGnol-L $_{CC\text{-}100}$	42.8	66.3
PAGnol-XL $_{CC\text{-}100}$	44.4	68.5

Table 6: Question answering on FQuAD.

Model	Size	EM	F1
GPT	small	45.5	62.7
GPT	medium	50.8	68.1

Table 7: GPT-2 small and medium model performance on SQuaD

rization task in this format:

$$\text{"}\{n_i\}\ \textit{Résumé:}\ \{a_i\}\text{"}$$

We finetune our model on the crossentropy loss computed only on the tokens of the produced summary. We optimize the learning rate and weight decay in the same interval as FLUE, using the same scheduler, and train for 4 epochs. We add a dropout with $p = 0.1$ to improve the performance. We evaluate the fine-tuned model using greedy token generation and the ROUGE metrics (R-1, R-2, R-L for ROUGE-1, ROUGE-2, ROUGE-L) (Lin, 2004). This task, more geared towards generation, sees PAGnol-XL establish a new state of the art for summarization on OrangeSum, shown in Table 8.

Model	Parameters	R-1	R-2	R-L
BARThez	216 M	**31.44**	12.77	22.23
PAGnol-S $_{OSCAR}$	124 M	22.79	6.16	16.03
PAGnol-M $_{OSCAR}$	355 M	24.89	7.87	17.78
PAGnol-S $_{CC\text{-}100}$	124 M	26.47	9.49	17.12
PAGnol-M $_{CC\text{-}100}$	355 M	28.20	10.80	20.79
PAGnol-L $_{CC\text{-}100}$	773 M	28.12	11.05	20.81
PAGnol-XL $_{CC\text{-}100}$	1.5 B	31.17	**12.86**	**22.50**

Table 8: Text summarization on the OrangeSum Abstract task. Best results are highlighted in **bold**, and second best are underlined.

7. Prompt Tuning

Human prompt engineering to extract good zero- and few-shot performance for large language models has motivated research in *prompt tuning*: placing some random vectors in the input sequence and optimizing their values, while keeping the pre-trained model weights fixed. The advantage of this approach is that the model does not change, and only the prompt is optimized. We follow the approach in (Lester et al., 2021), and optimize a certain number k of tokens in our prompt for the three aforementioned tasks. The best hyperparameters per size per task have been selected by grid search for the value of k, learning rate and dropout. In particular we performed a grid search over $k = \{1, 5, 20, 35, 50\}$, learning rate in $\{0.3, 0.1, 0.01, 0.001, 0.0005\}$, and dropout in $\{0, 0.01, 0.1\}$. We show the results for FLUE, FQuAD, and OrangeSum in Table 9. We expected a smooth scaling in performance with size and to progressively close the gap with fine-tuning performance, as shown by (Lester et al., 2021), however this scaling slows significantly when we reach the XL model. We suspect a bug in our implementation of prompt tuning with Rotary Embeddings, causing the performance hit, therefore we temporarily show the results for the XL model in *italic* in this setting.

8. Discussion

Without fear of aligning an overused *cliché*, the release of large language neural models have not only revo-

	FLUE			FQuAD		OrangeSum		
PAGnol	Books	Music	DVD	EM	F1	R-1	R-2	R-L
S	88.50	87.95	88.24	0.243	0.427	24.54	8.98	18.45
M	91.60	92.65	90.69	0.320	0.561	27.80	10.56	20.29
L	92.60	93.10	91.69	0.365	0.526	28.25	11.05	21.03
XL	*92.50*	*93.25*	*92.14*	*0.403*	*0.450*	*28.72*	*11.08*	*20.89*

Table 9: Prompt tuning performance on FLUE CLS, FQuAD, and OrangeSum.

lutionized the NLP field by bringing a major leap in performance in almost every tasks they were applied to, they crucially changed the perception of the risks of their potential misuse. The point is that this dramatic boost of performance has led the field to rely on the capacity of those large models to transfer their, in layman's terms, "knowledge" to other tasks via various transfer learning modalities. Yet, with this transfer, the potential data biases inherent to large corpus collection used for pre-training are also susceptible to appear. Gehman et al. (2020) thoroughly demonstrated that all generative language models they tested (from GPT1 (Radford et al., 2018) trained on Book Corpus only to GPT3 (Brown et al., 2020) and CTRL (Keskar et al., 2019) trained on various corpora, including user-generated content and web-crawled data sets) were capable of producing toxic output in specific conditions and presented different ways of alleviating this behaviour. Having been pre-trained on Common Crawl-based corpora, our models are certainly not immune from toxic content generation. More generally, the question of knowing whether the pre-training data should be curated more or should the model output, depending on the downstream application in sight, be debiased, or filtered, directly is still the object of vivid debates among the community (Bender et al., 2021; Goldberg, 2021), while of course there is an agreement toward responsible use of such technology. In this aspect, the release of a GPT-generated text detector by Antoun et al. (2021b) along their Arabic GPT2 model is an interesting step toward this direction. Regarding the environmental aspects of this model, our model pretraining experiments consumed about 62k gpu hours from the Jean Zay HPC cluster. Being based in France, its energy mix is made of nuclear (65-75%), 20% renewable and the remaining with gas (or more rarely coal when imported from abroad) (S.Requena, Dir. of Jean Zay, P.C). Regarding the performance of our models which almost constantly outperform their closest French counterparts, the GPT_{fr} models (Simoulin and Crabbé, 2021), one explaining factor could be the size of our pretraining data set (30B token vs 3B). Given our computing limitation, we choose from the beginning to use an experimental protocol as comparable as possible to the one used for the CamemBERT model evaluation (Martin et al., 2019), it would of course be interesting to perform a head to head comparison with the GPT_{fr} model. In terms of raw performance, it has been constantly reported that GPT-based models on the billion parameter scale provide inferior performance when compared to their "regular" transformer counterparts in classic fine-tuning scenarios, our results confirm this for French as well while highlighting the interest of our models in generation-oriented scenarios (such as text summarization where PagnolXL establishes a new state of the art for French). As for English (Lester et al., 2021), our preliminary prompt-tuning results suggest that this approach is promising and could be a way to close this performance gap.

9. Conclusion

We presented the Pagnol model collection, the first released large scale generative model for French[3], to date the largest neural language model trained from scratch for French. Trained on the CCnet corpus, we used scaling laws to inform our training setup, resulting in an optimal use of our training budget. The evaluation of our models on various end-tasks demonstrated first, that the CCnet corpus was a better choice than the Oscar French instance when used for generation; second, they showed that our models provide the same range of performance than their English counterparts and established a new state of the art for summarization of French on OrangeSum. Pagnol-XL and our smaller models are available.

10. Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocation 2020-AD011012024 made by GENCI, enabling us to use the *Jean Zay* supercomputer. We thank Stéphane Réquena and the support team for their valuable help. We also acknowledge Louis Martin for helping us reproducing the CamemBERT settings for CCNet. Djamé Seddah was partly funded by the French Research National Agency via the ANR project ParSiTi (ANR-16-CE33-0021).

[3]Released on May 4th, 2021. https://twitter.com/LightOnIO/status/1389579858754293761?s=20

11. Bibliographical References

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. (2018). A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60.

Alexandr, N., Irina, O., Tatyana, K., Inessa, K., and Arina, P. (2021). Fine-tuning gpt-3 for russian text summarization. In *Proceedings of the Computational Methods in Systems and Software*, pages 748–757. Springer.

Antoun, W., Baly, F., and Hajj, H. (2021a). AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual), April. Association for Computational Linguistics.

Antoun, W., Baly, F., and Hajj, H. M. (2021b). Aragpt2: Pre-trained transformer for arabic language generation. *ArXiv*, abs/2012.15520.

Baines, M., Bhosale, S., Caggiano, V., Goyal, N., Goyal, S., Ott, M., Lefaudeux, B., Liptchinsky, V., Rabbat, M., Sheiffer, S., Sridhar, A., and Xu, M. (2021). Fairscale: A general purpose modular pytorch library for high performance and large scale training. https://github.com/facebookresearch/fairscale.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Caswell, I., Kreutzer, J., Wang, L., Wahab, A., van Esch, D., Ulzii-Orshikh, N., Tapo, A., Subramani, N., Sokolov, A., Sikasote, C., et al. (2021). Quality at a glance: An audit of web-crawled multilingual datasets. *arXiv preprint arXiv:2103.12028*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Eddine, M. K., Tixier, A. J.-P., and Vazirgiannis, M. (2020). Barthez: a skilled pretrained french sequence-to-sequence model. *arXiv preprint arXiv:2010.12321*.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. (2020). RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online, November. Association for Computational Linguistics.

Goldberg, Y. (2021). A criticism of "on the dangers of stochastic parrots: Can language models be too big", January.

Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T. B., Dhariwal, P., Gray, S., et al. (2020). Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, April.

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

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.

Kim, B., Kim, H., Lee, S.-W., Lee, G., Kwak, D., Jeon, D. H., Park, S., Kim, S., Kim, S., Seo, D., et al. (2021). What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers. *arXiv preprint arXiv:2109.04650*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lakim, I., Almazrouei, E., Alhaol, I. A., Debbah, M., and Launay, J. (2022). A holistic assessment of the carbon footprint of noor, a very large arabic language model. In *Challenges & Perspectives in Creating Large Language Models*.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., and Schwab, D. (2019). Flaubert: Unsupervised language model pre-training for french. *arXiv preprint arXiv:1912.05372*.

Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Louis, A. (2020). BelGPT-2: a GPT-2 model pre-trained on French corpora. https://github.com/antoiloui/belgpt2.

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y.,

Romary, L., de la Clergerie, É. V., Seddah, D., and Sagot, B. (2019). Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Müller, M. and Laurent, F. (2022). Cedille: A large autoregressive french language model. *arXiv preprint arXiv:2202.03371*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rajbhandari, S., Ruwase, O., Rasley, J., Smith, S., and He, Y. (2021). Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. *arXiv preprint arXiv:2104.07857*.

Schweter, S. (2020). German gpt-2 model, November.

Seddah, D., Chrupała, G., Çetinoğlu, Ö., van Genabith, J., and Candito, M. (2010). Lemmatization and lexicalized statistical parsing of morphologically-rich languages: the case of French. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 85–93, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Seddah, D., Tsarfaty, R., Kübler, S., Candito, M., Choi, J. D., Farkas, R., Foster, J., Goenaga, I., Gojenola Galletebeitia, K., Goldberg, Y., Green, S., Habash, N., Kuhlmann, M., Maier, W., Nivre, J., Przepiórkowski, A., Roth, R., Seeker, W., Versley, Y., Vincze, V., Woliński, M., Wróblewska, A., and Villemonte de la Clergerie, E. (2013). Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA, October. Association for Computational Linguistics.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. (2019). Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Simoulin, A. and Crabbé, B. (2021). Un modèle transformer génératif pré-entrainé pour le _ français. In *Traitement Automatique des Langues Naturelles*, pages 245–254. ATALA.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. (2021). Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Wang, B. and Komatsuzaki, A. (2021). GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2021). Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Zeng, W., Ren, X., Su, T., Wang, H., Liao, Y., Wang, Z., Jiang, X., Yang, Z., Wang, K., Zhang, X., et al. (2021). Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*.

12. Language Resource References

Abeillé, A., Clément, L., and Toussenel, F. (2003). Building a treebank for french. In *Treebanks*, pages 165–187. Springer.

d'Hoffschmidt, M., Belblidia, W., Brendlé, T., Heinrich, Q., and Vidal, M. (2020). Fquad: French question answering dataset. *arXiv preprint arXiv:2002.06071*.

Eddine, M. K., Tixier, A. J.-P., and Vazirgiannis, M. (2020). Barthez: a skilled pretrained french sequence-to-sequence model. *arXiv preprint arXiv:2010.12321*.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., and Schwab, D. (2019). Flaubert: Unsupervised language model pre-training for french. *arXiv preprint arXiv:1912.05372*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme

summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Ortiz Suárez, P. J., Romary, L., and Sagot, B. (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, July. Association for Computational Linguistics.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, É. (2020). Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012.