Unintentional Bilingualism in Large Language Models

Raimundo Becerra Parra
raimundo.becerra@tum.de

***Abstract*—The recent success of large language models (LLMs) in natural language processing (NLP) tasks has raised questions about the source of their cross-lingual capabilities, particularly in machine translation (MT). *Unintentional bilingualism* is the hypothesis that Briakou et al. [1] propose for explaining LLMs' translation abilities. In order to test this hypothesis, they design a methodology and apply it to their proprietary dataset and language model. In this research project, we aim to reproduce their method and release an open-source implementation, including source code, datasets and models, in the hope that future research can build on our work and further explore this hypothesis.**

1. Introduction

With the release of OpenAI's GPT-2, Radford et al. [2] demonstrated that large language models (LLMs) trained on a diverse corpus are capable of performing non-trivially on a variety of natural language processing (NLP) tasks, including machine translation (MT), in the few-shot and even zero-shot setting. With OpenAI's next iteration, GPT-3 [3], it was demonstrated that by increasing the capacity of LLMs one could achieve SOTA performance on a variety of tasks in the few-shot and zero-shot setting.

One of the most surprising results from these two papers is the ability of the GPT models to translate text to and from non-English languages, even when their training datasets are English-centric, without any *intentional* translation examples, and are trained on a language modeling objective instead of a parallel data alignment objective. The training data for GPT-2 was actively filtered so that it contained only English, while the training data for GPT-3 was more diverse: 92.65% of words were in English (en), 1.82% in French (fr), 1.48% in German (de), 0.77% in Spanish (es), 0.61% in Italian (it), 0.52% in Portuguese (pt), 0.34% in Dutch (nl), etc. While the composition of non-English text was relatively well known, what was not clear was where exactly the translation ability came from.

To the best of our knowledge, the first paper to try to tackle this question was [1]. They propose a methodology for finding *unintentional* translation examples in a NLP training dataset. In order to study the effect of these unintentional translation examples, and show that they have an outsized impact on the model's translation ability, they train a series of models with different levels of ablation. They show that the performance of their model on MT tasks is significantly impacted when unintentional translation examples are removed from the training data. They also show that bilingual and non-English monolingual training signals have a positive impact on the model's translation ability. They conclude that the translation ability of LLMs is due to the presence of unintentional translation, bilingual and non-English monolingual examples in the training data. We will refer to this phenomena as *unintentional* or *incidental bilingualism*, or as *bilingual contamination*

In this research project, we aim to reproduce the methodology of [1] using publicly available tools, datasets and language models, and to provide an open-source implementation of their method, publicly releasing the resulting datasets and models. We hope that this will allow other researchers to build on our work and to further investigate the impact of unintentional bilingualism on the translation ability of LLMs.

2. Related Works

Blevins et al. [4] were the first to propose that language contamination was responsible for the cross-lingual capabilities of English language models. They showed that English monolingual corpora commonly used in the literature contain non-negligible amounts of non-English tokens. Their experiments consisted on evaluating English language models on masked language modeling (MLM) and part-of-speech (POS) tagging, where they find that the quantity of the unintended non-English tokens present in their training dataset has a positive correlation to the performance of the downstream tasks on the corresponding non-English target language.

Briakou et al. [1] continue this line of research by evaluating the performance of their PaLM model on MT tasks for different levels of ablation. They ablation experiment consists of the following steps:

1) Train a smaller PaLM model with their unfiltered original dataset.
2) Remove all training data containing translations between English and 44 other languages. Train a second small PaLM model with the resulting dataset.
3) Remove all training data containing text in both English and any of the aforementioned 44 languages. Train a third model with the resulting dataset.
4) Remove all training data containing text in any of the 44 languages. Train a fourth model with the resulting dataset.

This effectively allows them to control for the unintended translation, bilingual and non-English training signals, and evaluate their individual impact on PaLM's ability to translate from and to English.

Li and Flanigan [5] investigate the issue of task contamination beyond the scope of MT. They perform a training data inspection and find that some LLMs present instances of task contamination in their training dataset, arguing that they should not be considered few-shot or zero-shot in those tasks anymore.

3. Methods

In this section we present the models, dataset and language identification tool used in this work. We make our code, models and datasets publicly available at HuggingFace [1] and GitHub [2].

3.1. Model

In this research project, we use the smallest GPT-2 model, with 124 million parameters, and its tokenizer pretrained on WebText [2]. The model and tokenizer are available in the HuggingFace's *Transformers* library [6].

3.2. Dataset

The dataset used in this project is the OpenWebText2 dataset, released as part of The Pile corpus [7]. It consists of scrapped websites taken from URLs posted on Reddit, similar to WebText [2] and OpenWebTextCorpus [8]. Although it is English-centric, no filter was used to remove non-English content. In order to analyse the dataset, we define the following units of text:

- **Document:** A document is a single website.
- **Example:** An example is a collection of one or more documents up to a maximum number of tokens. [1] uses a maximum length of 2048 tokens, this being PaLM's context length. Correspondingly, we use GPT-2's context length of 1,024 tokens as our maximum length. While [1] uses a special token for separating the documents, we simply use a newline token i.e. `\n`.
- **Instance:** An instance is either an entire document or a part of a document up to a maximum number of tokens. Again, we use 1,024 tokens as our maximum length.

3.3. Language Identification

Briakou et al. [1] use CMX [9], a language identification model for capable of identifying 100 languages at the token level. Since this tool is not publicly available, we use the word-level language identification model CoSwID[3] [10], trained to detect words in en, de, es, fr, it, pt and nl. It is also capable of identifying Corsican, but since it tends to produce false positives in this language, we do not include it into our analysis. In addition to the predicted language label, CoSwID outputs the confidence score or probability of such prediction, ranging from 0 to 1. Our language identification pipeline can be summarized as follows:

1) Feed an instance to CoSwID. Receive language labels and confidence scores for each word.
2) Group adjacent words with the same language label into a single block.
3) Calculate the average confidence score for each block. If the average confidence score of a block is below a certain threshold μ, then consider the block as an *ambiguous block*. We choose $\mu = 0.6$ after manually inspecting CoSwID's predictions on a random sample of instances.
4) Disambiguate the ambiguous blocks by using the following heuristic:
    a) Merge adjacent ambiguous blocks into a single block, even if their language label is different.
    b) Run each merged ambiguous block through fastText[4], a sentence-level language identification model [11][12].
    c) Replace the language label of the merged ambiguous block with the one predicted by fastText.

    Note that this may produce language labels outside of CoSwID's scope.

The result of this pipeline is that each instance in our dataset is divided into a sequence of blocks, each with a language label.

3.4. Bilingual Instance Classification

In order to analyse the degree of bilingual contamination in the dataset, the authors of [1] define bilingual and monolingual instances. In our experiments, an instance is considered **bilingual** if it contains at least two blocks with length greater than N and different language labels. Otherwise it is considered as **monolingual**. We choose $N = 10$, following [1].

3.5. Translation Instance Classification

The authors also classify some bilingual instances as **translation** instances. We use the same approach as in [1], which can be summarized as follows:

1) For each bilingual instance, run a sentence breaker. [1] do not mention which sentence breaker they

[1] https://huggingface.co/RaiBP

[2] https://github.com/RaiBP/translation-detection

[3] CoSwID is an open source tool: https://github.com/lkevers/coswid

[4] fastText Python wrapper: https://pypi.org/project/fasttext-langdetect/

use. In our experiments, we use NLTK's sentence breaker[5].

2) Label each sentence with the most frequent language label among the sentence's words. The language with the most sentences becomes the primary language, while the one with the second most sentences becomes the embedded language.
3) Get the embedding vectors of the primary language sentences and the embedded language sentences. Following [1], we use the LABSE[6] [13] model and tokenizer to get the embeddings.
4) Calculate the cosine similarity between each primary language sentence and each embedded language sentence. If the similarity is above a certain threshold τ, we consider the embedded language sentence as a possible translation of the primary language sentence, and we call this a translation pair. We choose $\tau = 0.6$ following [1].
5) Filter the resulting translation pairs using Alibaba's WMT Data Filtering submissions [14, 15]. This is the same approach [1] use to filter translation pairs. We additionally filter all translation pairs where either of the sentences contains no letters. Whereas [1] use the sentence-level detection tool from [16] to confirm that the sentences in the translation pair are in different languages, we use fastText instead.

Any instance with at least one translation pair is considered a translation instance.

4. Experiments

In this research project, we perform two experiments. First, we analyse the language distribution of the OpenWebText2 dataset, using instances as the unit of analysis. Second, we perform an ablation study to investigate the impact of incidental bilingualism on the text generation performance of a language model in different languages.

4.1. Instance Extraction and Classification

We select the first 31 chunks of OpenWebText2's train set[7], totaling approximately 3B tokens. We select this amount of tokens to comply with the Chinchilla scaling laws [17] for training a 124M parameter model. We ran the language identification pipeline on each chunk and obtained their instance composition. The total amount of instances of each class across all chunks are shown in Table 1. The *Other* category under *Monolingual* includes languages such as Corsican and Romanian that might have arisen during detection, but that we do not explicitly analyse in this report. Under *Bilingual* we explicitly report bilingual instances where two languages are present, one of them being English and the other one being among de, fr, es, it, pt or nl. Bilingual instances where more than two languages are present, or where two languages are present but none of them are English, or where two languages are present, one of them being English and the other one being not one of the aforementioned languages, are included in the *Other* category. Under the *Translation* category we explicitly report instances where English is paired with de, fr, es, it, pt or nl; all other translation pairs are reported under the *Other* category. Note that we do not differentiate between English being the primary or the embedded language in a translation pair. If an instance has multiple translation pairs in different pairs of languages, we count it as belonging to the majority pair. We also report the percentage of each class in the dataset, considering that 5,919,324 instances were extracted from the dataset in total.

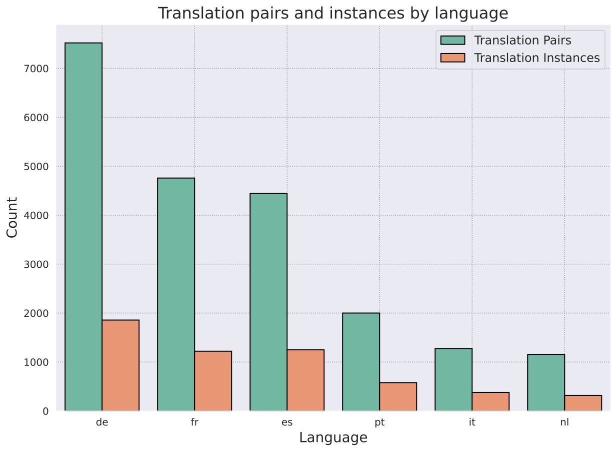


Figure 1: Translation pair and instance counts for English paired with de, fr, es, pt, it or nl.

Figure 1 shows the number of translation pairs and translation instances for English paired with de, fr, es, pt, it or nl, while Figure 2 presents the number of translation, bilingual and monolingual instances for these languages. Figure 3 shows the Pearson correlation between monolingual instances and bilingual instances, and between monolingual instances and translation instances. We find a strong correlation between monolingual and bilingual instances ($r = 0.96$) and a slighly weaker correlation between monolingual and translation instances ($r = 0.81$). The linear relation found by [1] in their own dataset is also shown for comparison.

The raw output dataset of our language detection program containing all instance labels, alongside the scripts used to obtain the instance counts, is publicly available at HuggingFace[8].

4.2. Ablating Incidental Bilingualism

Data In order to study the impact of incidental bilingualism on downstream tasks, it is necessary to first ablate it from the dataset. We take the instance labels obtained in

[5] Python Natural Language Toolkit: https://www.nltk.org/

[6] Available at https://huggingface.co/setu4993/LaBSE

[7] Available at https://huggingface.co/datasets/ArmelR/the-pile-splitted

[8] Available at https://hf.co/datasets/RaiBP/openwebtext2-first-30-chunks-lang-detect-raw-output

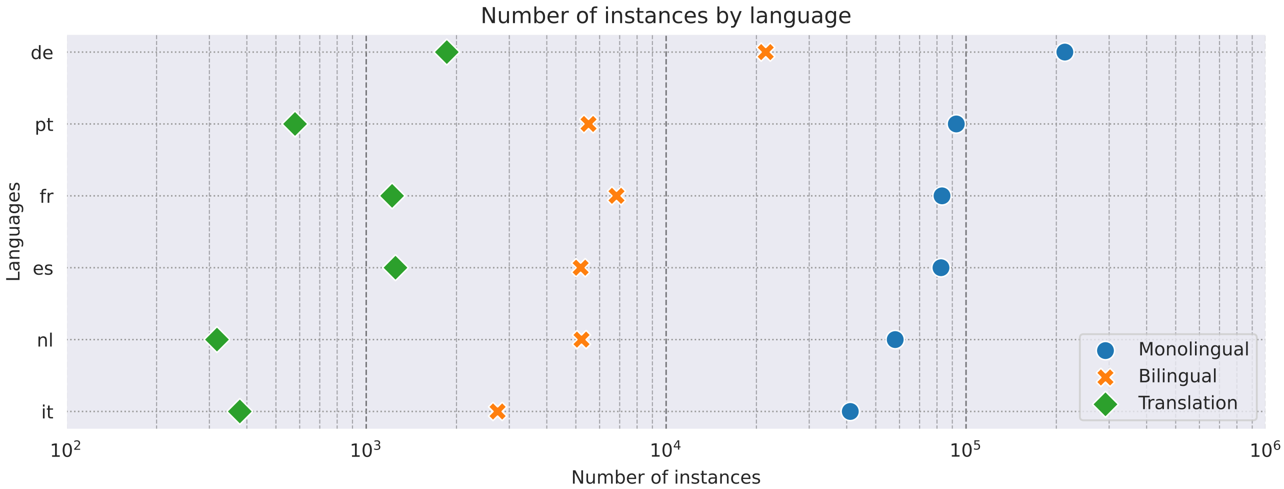


Figure 2: Number of monolingual, bilingual, and translation instances detected for each language on the first 31 chunks of OpenWebText2's train set. Bilingual and translation instances are paired with English.

Table 1: Instance composition of the first 31 chunks of OpenWebText2's train set.

Type	Language	# of instances	Percentage [%]
Monolingual	en	5,047,722	85.28
	de	213,539	3.61
	fr	83,158	1.40
	es	82,499	1.39
	it	41,119	0.69
	pt	92,789	1.57
	nl	58,049	0.98
	Other	225,636	3.81
	Total	5,844,511	98.74
Bilingual	de	21,485	0.36
	fr	6,832	0.12
	es	5,188	0.09
	it	2,742	0.05
	pt	5,512	0.09
	nl	5,223	0.09
	Other	27,831	0.47
	Total	74,813	1.26
Translation	de	1,857	0.03
	fr	1,220	0.02
	es	1,252	0.02
	it	379	0.01
	pt	579	0.01
	nl	318	0.01
	Other	1,895	0.03
	Total	7,500	0.13

the previous experiment, and create four non-overlapping groups: English (**ENG**) instances, non-English (**NEN**) instances (excluding bilingual and translation), bilingual (**BIL**) instances (excluding translation) and translation (**TRA**) instances. This is the same partition used in [1] for their ablation experiment. Equivalently to their method, we merge instances within their groups into 1,024 token examples using the following heuristic:

Table 2: Data statistics for the ablation experiment in number of examples.

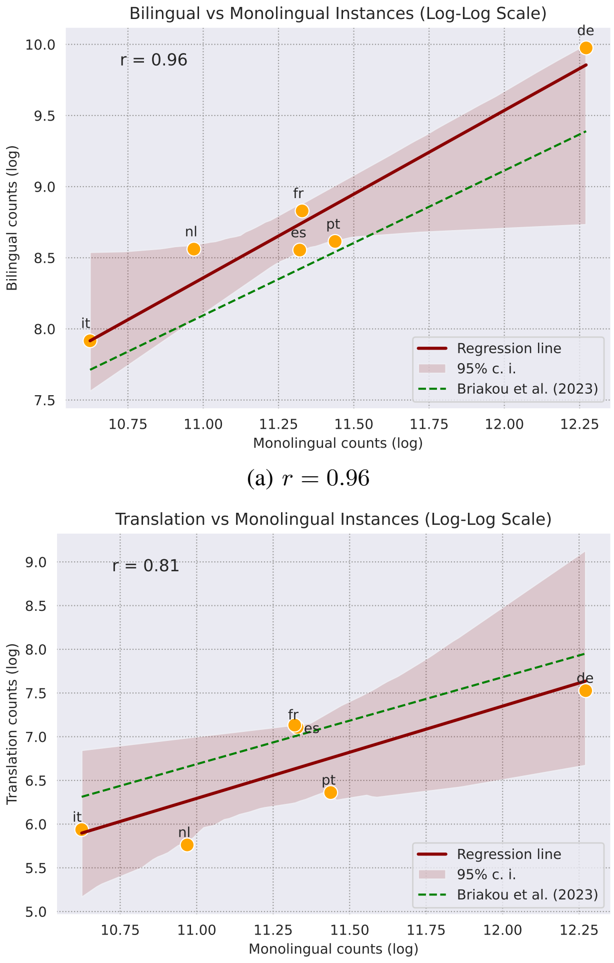
	ENG	**NEN**	**BIL**	**TRA**
FULL	3,020,621	481,215	32,302	6,521
-TRA	3,020,621	481,215	38,823	✗
-BIL	3,020,621	520,038	✗	✗
-NEN	3,540,659	✗	✗	✗

- All instances with 1,024 tokens are left as they are, as a single-instance example.
- All instances with less than 1,024 tokens are merged into an example of up to a maximum of 1,024 tokens using a greedy approach. We use the newline token `\n` between instances.

Note that this approach may result in examples of less than 1,024 tokens. Counting examples from each group gives us the following distribution: **ENG**: 85.3%; **NEN**: 13.6%; **BIL**: 0.9%; **TRA**: 0.2%. This is comparable to the distribution found by [1] in PaLM's dataset. We then proceed to ablate each group in the following order: **TRA**, **BIL** and **NEN**. The example counts for each ablation are shown in Table 2. The ablation datasets are publicly available at HuggingFace[9].

Training Four GPT-2 models (small version with 124M parameters) are trained from scratch, one per ablation dataset. Each model is trained using all examples in the respective dataset, on two NVIDIA Quadro RTX 6000 GPUs with an effective batch size of 64, learning rate of 0.005 and Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and

[9] **FULL** dataset: https://hf.co/datasets/RaiBP/openwebtext2-first-30-chunks-ablation-full; **-TRA** dataset: https://hf.co/datasets/RaiBP/openwebtext2-first-30-chunks-ablation-translation; **-BIL** dataset: https://hf.co/datasets/RaiBP/openwebtext2-first-30-chunks-ablation-bilingual; **-NEN** dataset: https://hf.co/datasets/RaiBP/openwebtext2-first-30-chunks-english-only-examples



(a) $r = 0.96$

(b) $r = 0.81$

Figure 3: Pearson correlation between counts of monolingual instances with (a) bilingual instances and (b) translation instances. A linear regression applied to the data is shown in red. The linear relation found by [1] is shown in green.

$\epsilon = 10^{-8}$. Training takes about 2 days per model. The trained models and all training details are publicly available at HuggingFace[10].

Evaluation While it would be ideal to evaluate our ablation models on their translation abilities, similarly to [1], it is practically impossible do so for such small models without further fine-tuning. Instead, we evaluate their text generation ability. For this, we use two datasets: LAMBADA[11] [18] and Wikipedia[12] [19].

LAMBADA is a widely used English-only dataset for evaluating pre-trained language models. OpenAI uses it to evaluate their GPT-2 [2] and GPT-3 [3] models, for example. LAMBADA tests whether the language model is capable of predicting the last word of a passage, given the entire passage. We perform the evaluation using the LM Evaluation Harness library [20], where machine-translated versions of LAMBADA for de, fr, es and it are available alongside the original English dataset. The perplexity and accuracy evaluation results for the four ablation models are presented in Table 3. We include the results for small GPT-2 as a baseline.

Testing the language generation capability of a language model on Wikipedia articles is common practice, with WikiText [21] arguably being the most popular. Since WikiText is only available in English, we use instead the Wikipedia dataset, a dump of Wikipedia in many languages. Since there is no predefined test dataset, we design our own experiment. For each language and each model, we:

1) Sample 2,000 articles at random.
2) Combine the 2,000 articles into a single string of text, using three newline tokens (`\n\n\n`) between each article.
3) Calculate the perplexity of the model on the combined string of text, using a stride of 1,024 tokens.
4) Repeat the process 5 times and average the perplexity scores.

The results are also presented in Table 3, including small GPT-2 as a baseline.

5. Discussion

5.1. Instance Composition

We find that the instance composition of OpenWebText2 is not too different from PaLM's dataset [1]. They find that 1.4% of PaLM's training instances are bilingual instances, while we find that 1.26% of OpenWebText2's instances are bilingual instances. Similarly, they find that 0.34% of PaLM's training instances contain at least one translation pair, while we find that 0.13% of OpenWebText2's instances do. When considering de, fr, es, it, pt and nl paired with English, translation instances are about one order of magnitude less frequent than bilingual instances, and bilingual instances are about one order of magnitude less frequent than monolingual instances, as Figure 2 shows. This was also found by [1]. Additionally, we also find that there is a strong correlation between the monolingual counts and translation counts, and monolingual counts and bilingual counts.

5.2. Ablation Experiment

From Table 3 it is clear that removing all non-English instances has the greatest effect on the language model's generation ability in non-English languages. The effect of

[10] **FULL** model: https://hf.co/RaiBP/gpt2-openwebtext2-first-30-chunks-ablation-full; **-TRA** model: https://hf.co/RaiBP/gpt2-openwebtext2-first-30-chunks-ablation-translation; **-BIL** model: https://hf.co/RaiBP/gpt2-openwebtext2-first-30-chunks-ablation-bilingual; **-NEN** model: https://hf.co/RaiBP/gpt2-openwebtext2-first-30-chunks-ablation-non-english

[11] Available at https://hf.co/datasets/EleutherAI/lambada_openai

[12] Available at https://hf.co/datasets/wikimedia/wikipedia

Table 3: Results of language generation experiments. The best result for each language is in bold. ↑ means that higher is better, ↓ means that lower is better. Metrics across languages are not comparable.

Language	Model	Dataset (Metric)					
		LAMBADA (PPL) ↓	LAMBADA STD (PPL)	LAMBADA (ACC) ↑	LAMBADA STD (ACC)	Wikipedia (PPL) ↓	Wikipedia STD (PPL)
en	**FULL**	294	14	**16.65**	0.52	42.89	0.92
	-TRA	273	12	16.17	0.51	**42.24**	1.00
	-BIL	303	14	15.80	0.51	42.76	0.95
	-NEN	**235**	11	18.30	0.54	43.32	0.88
	GPT-2	40	1	32.56	0.65	28.42	1.18
de	**FULL**	6,359	448	6.70	0.35	27.16	0.56
	-TRA	6,591	461	6.29	0.34	**26.85**	0.52
	-BIL	**5,733**	401	**6.71**	0.35	27.89	0.97
	-NEN	1,843,331	199,462	3.43	0.25	257.12	5.36
	GPT-2	81,093	6,633	4.25	0.28	64.49	1.15
fr	**FULL**	7,070	464	8.21	0.38	**27.66**	0.19
	-TRA	6,872	447	8.11	0.38	27.70	0.18
	-BIL	**5,873**	382	**8.54**	0.39	28.69	0.24
	-NEN	169,906	12,935	5.18	0.31	188.19	4.49
	GPT-2	19,714	1,360	8.36	0.39	56.85	0.96
es	**FULL**	**18,725**	1,244	5.08	0.31	23.31	0.33
	-TRA	22,110	1,431	4.74	0.30	**23.17**	0.35
	-BIL	19,580	1278	**5.14**	0.31	24.01	0.34
	-NEN	1,101,939	94,426	3.30	0.25	215.25	3.66
	GPT-2	112,397	8,322	5.41	0.32	64.63	0.93
it	**FULL**	14,151	979	5.49	0.32	29.77	0.51
	-TRA	13,450	915	5.76	0.32	**29.31**	0.56
	-BIL	**11,449**	779	**5.84**	0.33	30.20	0.52
	-NEN	410,357	35,029	5.18	0.31	265.61	10.62
	GPT-2	68,936	5,092	6.44	0.34	96.60	3.51
pt	**FULL**	-	-	-	-	21.96	0.55
	-TRA	-	-	-	-	**21.57**	0.52
	-BIL	-	-	-	-	22.57	0.53
	-NEN	-	-	-	-	216.44	4.63
	GPT-2	-	-	-	-	84.87	1.94
nl	**FULL**	-	-	-	-	35.09	0.37
	-TRA	-	-	-	-	**34.63**	0.41
	-BIL	-	-	-	-	35.56	0.44
	-NEN	-	-	-	-	280.28	15.35
	GPT-2	-	-	-	-	107.33	5.74

removing translation and bilingual instances has no distinguishable effect on the results. This is not entirely unexpected, as we are testing monolingual ability instead of bilingual ability (as one would do in a machine translation task, for example). An interesting result is that our baseline, small GPT-2, consistently performs between **-NEN** and the rest of the ablation models, implying that GPT-2's dataset has some non-English monolingual training instances, but not as many as those found on OpenWebText2. This is expected, since GPT-2 was trained on WebText, where non-English webpages were filtered out, while OpenWebText2 keeps them.

6. Conclusion

In this research project, we have presented an open-source version of Briakou et al. [1]'s method for identifying unintentional bilingual and translation instances in a language model's training dataset. We have also presented an ablation experiment, where we remove different types of instances from OpenWebText2 and evaluate the resulting models on their text generation ability. We find that removing all non-English instances has the greatest effect on the language model's generation ability in non-English languages. The effect of removing translation and bilingual instances has no distinguishable effect on the results.

In order to truly test and potentially confirm the hypothesis that translation and bilingual instances on the training dataset have a significant impact on a language model's translation ability, it is necessary to directly evaluate the ablation models on a translation task. This would require further fine-tuning, which is outside the scope of this work. However, we believe that the methods and models presented in this research project and made publicly available will make it easier for future researchers to perform such experiments.

References

[1] Eleftheria Briakou, Colin Cherry, and George Foster. Searching for Needles in a Haystack: On the Role of Incidental Bilingualism in PaLM's Translation Capability, May 2023. URL http://arxiv.org/abs/2305.10266. arXiv:2305.10266 [cs].

[2] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI, February 2019.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Terra Blevins and Luke Zettlemoyer. Language Contamination Helps Explain the Cross-lingual Capabilities of English Pretrained Models, November 2022. URL http://arxiv.org/abs/2204.08110. arXiv:2204.08110 [cs].

[5] Changmao Li and Jeffrey Flanigan. Task contamination: Language models may not be few-shot anymore. *arXiv preprint arXiv:2312.16337*, 2023.

[6] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

[7] 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. The Pile: An 800GB Dataset of Diverse Text for Language Modeling, December 2020. URL http://arxiv.org/abs/2101.00027. arXiv:2101.00027 [cs].

[8] Aaron Gokaslan and Vanya Cohen. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

[9] Yuan Zhang, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. A Fast, Compact, Accurate Model for Language Identification of Codemixed Text, October 2018. URL http://arxiv.org/abs/1810.04142. arXiv:1810.04142 [cs].

[10] Laurent Kevers. CoSwID, a Code Switching Identification Method Suitable for Under-Resourced Languages. In Maite Melero, Sakriani Sakti, and Claudia Soria, editors, *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 112–121, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.sigul-1.15.

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

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

[13] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.62. URL https://aclanthology.org/2022.acl-long.62.

[14] Jun Lu, Xiaoyu Lv, Yangbin Shi, and Boxing Chen. Alibaba submission to the WMT18 parallel corpus filtering task. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 917–922, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6481. URL https://aclanthology.org/W18-6482.

[15] Jun Lu, Xin Ge, Yangbin Shi, and Yuqi Zhang. Alibaba submission to the WMT20 parallel corpus filtering task. In Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors, *Proceedings of the Fifth Conference on Machine Translation*, pages 979–984, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.111.

[16] Jan A. Botha, Emily Pitler, Ji Ma, Anton Bakalov, Alex Salcianu, David Weiss, Ryan McDonald, and Slav Petrov. Natural language processing with small feed-forward networks. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2879–2885, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1309. URL

https://aclanthology.org/D17-1309.

[17] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[18] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

[19] Wikimedia Foundation. Wikimedia downloads. URL https://dumps.wikimedia.org.

[20] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

[21] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.