Stealing Brains: From English to Czech Language Model

Petr Hyner[1,2], Petr Marek[3], David Adamczyk[1,2], Jan Hůla[1,3] and Jan Šedivý[3]
[1]*Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, Ostrava, Czech Republic*
[2]*Department of Informatics and Computers, Faculty of Science, University of Ostrava, Ostrava, Czech Republic*
[3]*Czech Technical University in Prague, Prague, Czech Republic*

Keywords: Language Models, Neural Networks, Transfer Learning, Vocabulary Swap.

Abstract: We present a simple approach for efficiently adapting pre-trained English language models to generate text in lower-resource language, specifically Czech. We propose a vocabulary swap method that leverages parallel corpora to map tokens between languages, allowing the model to retain much of its learned capabilities. Experiments conducted on a Czech translation of the TinyStories dataset demonstrate that our approach significantly outperforms baseline methods, especially when using small amounts of training data. With only 10% of the data, our method achieves a perplexity of 17.89, compared to 34.19 for the next best baseline. We aim to contribute to work in the field of cross-lingual transfer in natural language processing and we propose a simple to implement, computationally efficient method tested in a controlled environment.

1 INTRODUCTION

State-of-the-art large language models (LLMs) demonstrate proficiency in English text generation (OpenAI et al., 2024; Dubey et al., 2024; Jiang et al., 2023; Anil et al., 2024), while exhibiting comparatively limited capabilities in lower-resource languages (Jin et al., 2023; Wendler et al., 2024). This imbalance can be attributed to a higher representation of English in training corpora (mostly data scraped from the Internet) relative to other languages.

Currently, there are two primary approaches that exist for developing language-specific large language models: training from scratch and fine-tuning existing models. However, the computational resources required to train large language models capable of generating high-quality text from scratch are prohibitively expensive for most academic institutions, research centers, or other subjects. Recent estimates suggest that training costs for state-of-the-art LLMs can range from tens to over a hundred million dollars, primarily due to hardware requirements and energy consumption.

This contribution proposes an approach that leverages a pre-trained English language model to develop an adapted model capable of generating coherent text in Czech. Our aim is not to develop a Czech language model that would be competitive with other Czech models, but to explore the efficiency of different methods on small sample regimes, where we can conduct experiments quickly and with a small computational budget. Specifically, to demonstrate the efficacy of the presented method, we adapt a *GPT Neo* model, trained on an English corpus, to a parallel corpus translated into Czech.

Our research focuses on various techniques for model adaptation that avoid the need for training from scratch or full fine-tuning. In doing so, we aim to significantly reduce the computational costs associated with developing language-specific models.

We show that if the target corpus is very small compared to the original English corpus, the presented method leads to drastic improvements compared to training from scratch and other baselines.

The text is structured as follows. Section 2 will discuss the method for fine-tuning a language model using vocabulary swap. The following Section 3 will discuss how we obtained a parallel corpus by translating the original corpus and how having parallel corpora is useful for training a language model. Section 4 discusses the specific models and data with which we worked, along with the baselines against which we tested, and we present experimental results by comparing the perplexity of trained models on a valida-

Hyner, P., Marek, P., Adamczyk, D., Hůla, J. and Šedivý, J.
Stealing Brains: From English to Czech Language Model.
DOI: 10.5220/0013064500003837
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 16th International Joint Conference on Computational Intelligence (IJCCI 2024)*, pages 606-612
ISBN: 978-989-758-721-4; ISSN: 2184-3236
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

tion set across models and data splits. In Section 5 we discuss relevant work. Section 6 discusses limitations to our approach, and we conclude in Section 7.

2 THE PROPOSED METHOD

In this section, we describe the method that produced the best results in our experiments. We use the name *source model* for the model trained on English texts and *target model* for the model adapted to Czech language.

2.1 Motivation

The motivation for the method is based on the fact that if the two languages would be used to speak about the same topics and would have the same grammar, then the main obstacle for understanding the text in the second language would be the different vocabulary. If further the tokenizer would tokenize the sentences to whole words and we would have a mapping from English to Czech words, then we could just replace the Czech words with their English counterparts and use the source model to generate text in English, and finally replace the English words in the generated text with their Czech counterparts.

The assumption that texts in Czech and English would be very similar in meaning is probably benign, and therefore we conduct our experiments on Czech translation of the English corpus on which the source model was trained. Unfortunately, Czech and English do not have the same grammar and word order, and therefore the source model would need to be fine-tuned for the new grammar. It could be expected that during fine-tuning the model would just need to adapt to a new grammar, and all its other capabilities (reasoning, world knowledge, etc.) would transfer from the source model, and therefore not many training samples would be needed for fine-tuning.

The final obstacle to overcome is the fact that the tokenizer of the source model does not tokenize sentences to whole words and often is also not suitable for the tokenization of the target language (Czech in our case). Therefore, in practice, we have two different tokenizers, which split words into subwords, in each language in a different way. This makes remapping tokens between the two languages nontrivial.

The naive solution would be to let the source model adapt to a new vocabulary together with the new grammar. We could hope that the model would still transfer many capabilities from the source language. This would mean that the subcircuits which are not dealing with grammar and word meaning would be left intact and the model would just adapt the low-level circuits dealing with word embeddings and grammar (this is of course a simplified rationale, as sometimes high-level reasoning is needed to correctly parse a sentence and understand the meaning of a given word). Nevertheless, it is possible that the model would be too “confused” with the new vocabulary and grammar and would start to reorganize all its subcircuits during fine-tuning, and therefore would need a lot of training data to rediscover the capabilities such as reasoning, etc. It is even possible that such fine-tuning would work even worse than training the model from scratch (as can be seen in our experiments).

We therefore want to keep as much of the form of the original English sentences as possible so that the model does not start to fully reorganize itself and the old subcircuits useful for the new task are kept intact. We attempt to achieve this by identifying Czech words which are tokenized to one token using the Czech tokenizer and whose translation to English is also tokenized to one token using the English tokenizer. Such words could be swapped, and if there is enough such words, the model could have an easier time to “reorient” itself to the new language.

2.2 High-Level Summary

On the high-level, the presented method is straightforward. We assume that we have a well-trained model for English language and a tokenizer optimized for the Czech training corpus. Our goal is to fine-tune the original model with the new tokenizer, but we also want to make the transition to the new language as smooth as possible. Therefore, we try to remap the tokens in the Czech language to the tokens in the English language as closely as possible and then fine-tune the original model without any additional tricks.

The efficiency of this method will depend on the semantic overlap of the content in the training data for the source and target model. If the training data for the source model contain texts about biology and the training data for the target model contain texts about mathematics, there is probably not much space for the transfer of knowledge and capabilities. In order to avoid uncertainties about the semantic similarity of texts in Czech and English training data, we conduct our experiments on a Czech corpus which is obtained by translating the original English training corpus into a Czech language. We also leverage the fact that we have such parallel corpus in order to find the mapping between Czech and English tokens which we describe in the next subsection.

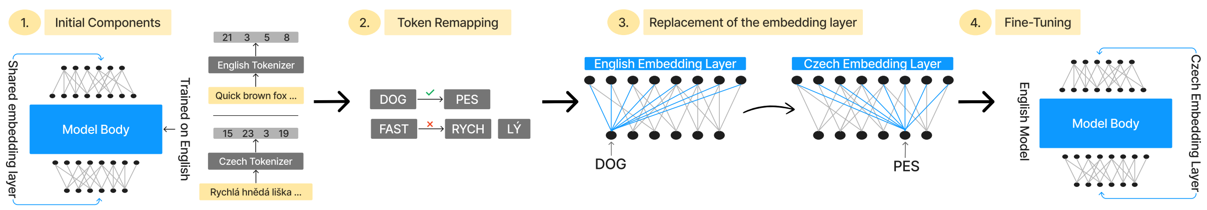

Figure 1: High-level diagram of the proposed method. The diagram shows the process of the application of the proposed method, where we begin with a pre-trained model trained on English, and a tokenizer trained on Czech. It shows the process of remapping the source language tokens to the target counterparts and the replacement of the source embedding layer with a target embedding layer, where the indices of Czech tokens correspond to the semantically equivalent tokens of the source language. Finally, we fine-tune the model with now a new Czech embedding layer after we apply these steps.

2.3 Vocabulary Swap

In this section, we use the term *embedding matrix* for the layer that maps the indices of the tokens to their corresponding token embeddings (columns of the matrix). This matrix is reused in the final layer, where its transpose multiplies the final token embeddings to obtain the logits for individual tokens that should be predicted in the next step. We use the term *model body* for all other layers.

As mentioned, our aim is to find a partial mapping, which will map tokens of the Czech tokenizer to tokens of the English tokenizer. Once we have such mapping, we can initialize the columns of the embedding matrix for the Czech tokens with the corresponding columns of the embedding matrix for the English tokens. This is depicted in Figure 1. If we manage to remap a large portion of the whole vocabulary, we can expect that the inputs to the model body will be very similar for both languages and therefore the subcircuits for the high-level capabilities and knowledge would be retained during fine-tuning.

Mapping Between Czech and English Tokens. The mapping between Czech and English tokens can be obtained in multiple ways. The most basic approach would be to translate the Czech tokens which correspond to whole words to English and if the translation of the Czech token corresponds to a token from the English tokenizer, then this pair would be added to the mapping. This naive approach could miss a lot of pairs due to the ambiguity in translation.

To have a high-quality mapping, we leverage the fact that we have a parallel corpus in which we have a corresponding English sentence for each Czech sentence. To create the mapping, we iterate over all pairs of corresponding sentences and update the pair counter, which counts how many times a given Czech token was mapped to a given English token. Concretely, this is achieved with the following steps:

1. Obtain word embeddings for each Czech token in the Czech sentence[1].
2. Obtain word embeddings for each English token in the English sentence.
3. For each Czech token, find the best matching English token by using the cosine similarity between the corresponding embeddings.
4. For each mapped pair update the pair counter.

Once we iterate over all pairs of sentences, we select the pairs for which the count exceeded a given threshold (20). We choose this threshold arbitrarily, and the reason we introduce the threshold is to limit noise. Finally, we use each selected pair to initialize the Czech embedding matrix, i.e., if there is a pair with Czech token with ID 5 and English token with ID 9, then we copy the ninth column of the English embedding matrix to the fifth column of the Czech embedding matrix. Using this procedure, we managed to map **14,272 tokens**, which is approximately **47.57%** of the Czech vocabulary.

3 CREATION OF THE PARALLEL CORPUS

To translate the English *TinyStories* (Eldan and Li, 2023) dataset into Czech, we used the advanced capabilities of the *WMT 21 En-X* (Tran et al., 2021)[2]. This model, part of the WMT 21 En-X suite, is a state-of-the-art multilingual encoder-decoder (seq-to-seq) model with 4.7 billion parameters, trained to perform one-to-many multilingual translation, including Czech.

To manage the size of the TinyStories dataset and

[1]We first translate the Czech tokens to English and than use the word embeddings for English words in both sentences. Word embeddings are obtained using the FastText library.

[2]facebook/wmt21-dense-24-wide-en-x Hugging Face model

address memory limitations of GPU [3] we employed, we utilized the sentence tokenizer from the *spaCy* library[4] (Honnibal et al., 2020) to split the dataset into individual sentences. Each sentence was then translated separately using the WMT 21 En-X model. This sentence-level approach not only mitigated memory constraints but also allowed for efficient parallel processing of the dataset.

Once translated, the sentences were meticulously recombined to recreate complete records that mirrored the original dataset structure and narrative coherence. This method ensured that each translated story retained its narrative flow and logical consistency, making the Czech dataset structurally equivalent to the original English version.

4 EXPERIMENTS

We will be using the TinyStories (Eldan and Li, 2023) dataset along with a pre-trained TinyStories GPT Neo 33 million parameter model for fine-tuning. For training from scratch, we will use the same model architecture without any pre-training. Since we are using a custom tokenizer, we change the vocabulary size of the model to 30,000 tokens, which corresponds to the number of tokens our tokenizer learned.

4.1 Data

The TinyStories dataset contains simple stories written in simple English. The authors describe the available vocabulary from which these stories are generated as containing words that a 3–4-year-old child would typically understand or use; therefore, the vocabulary only contains 1500 unique, simple words. These data were generated using *GPT 3.5* or *GPT 4* and there are approximately **2.1 million samples** in the dataset. It is important to mention that the authors have also published other models of various sizes, so for our initial experiments, we used the smallest, 1 million parameter model to not waste unnecessary resources.

As discussed in Section 3, we translated the original TinyStories corpus to obtain a parallel corpus in Czech. We trained or fine-tuned all models on data splits, the same for each model. We evaluated the models on a fixed test set, which is the same for each model.

We tested the three baselines on five different partitions of data: 10%, 25%, 50%, 75%, and 100%. Our goal is to show the viability of the proposed method when we use a small amount of data, obtaining significantly better results than those produced by the baselines.

[3]NVIDIA Tesla V100 GPU accelerators with 16 GB

[4]en_core_web_sm model

4.2 Baselines

TinyStories models use the GPT Neo architecture, which utilizes a decoder-only transformer structure.

The training process then proceeds as follows: We utilize the default GPT Neo hyperparameters, we set the batch size to 8 samples, and the gradient accumulation to 128 steps. We then train for 1 epoch on the target corpus on a single A100 GPU. We used the same hyperparameters for each baseline and data split, as well as for the proposed method. We also experimented with training the models for multiple epochs; see Figure 4.

We have evaluated a model trained using the proposed method of **Vocabulary Swap** against three baselines: a model trained from scratch, a model trained using naive fine-tuning, and a model trained using fine-tuning with embedding reset.

Training from Scratch. For this baseline, we train a 33 million parameter GPT Neo model from randomly initialized weights. As mentioned above, we use the same hyperparameters for all training regimes.

Naive Fine-Tuning. Here, we begin with the 33 million parameter GPT Neo source model pre-trained on the source language. We use the aforementioned tokenizer and target corpus. One could observe that this naive approach has a big flaw; token IDs of the target tokenizer will not correspond to the token IDs of the source tokenizer, and therefore a word “pes” (dog in English) could start with an embedding of a completely different word (e.g. “sun”) and this could cause that the model body will start change in order to adapt to the new embeddings and therefore can lose a lot of abilities that it previously learned on the source language.

Fine-Tuning with Embedding Reset. With this approach, we apply the same procedure as with Naive Fine-tuning, with the only difference that we reset the embeddings of tokens by sampling them from a Gaussian distribution with the mean and standard deviation calculated from the original embeddings for the source language.

4.3 Results

We use *perplexity* (PPL) as our evaluation metric, as it is a common metric to measure the quality of generative models. We evaluated the trained models on a test set and calculated the PPL for each sentence using the following Formula 1.

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\} \quad (1)$$

Where $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the i-th token conditioned on the preceding tokens $x_{<i}$ according to our model.

Formula 1 corresponds to an exponentiated negative log-likelihood of a sequence.

We can compute the perplexity for the entire test set as the mean of the perplexities for all individual sentences, using Formula 2. The test set should have a high probability according to the evaluated model.

$$\overline{PPL} = \frac{1}{N}\sum_{i=1}^{N} PPL(X_i) \quad (2)$$

Where $\overline{PPL}$ is the average perplexity of the entire test set and N is the total number of sentences.

Figure 2 shows the perplexity of each baseline, as well as the models trained using the proposed method. Each model was trained on four data splits that were sampled once, as well as on all data. We can see that using only 10% of the data, the proposed method shows a significant improvement compared to the baselines; we show the evaluation results in Table 1.

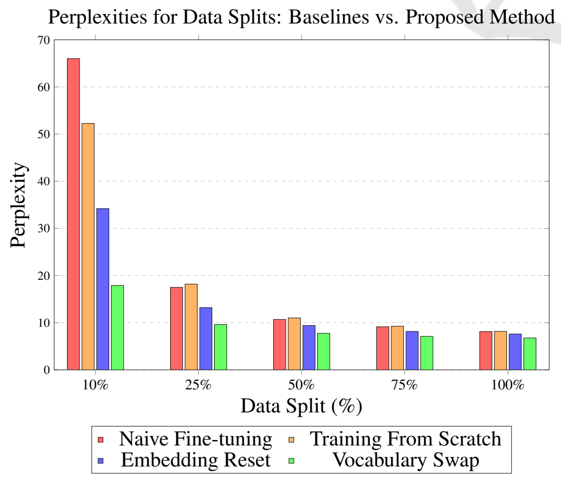

Figure 2: Perplexity comparison for different training approaches across various data splits. This serves as a visualization of Table 1, where each bar represents the mean perplexity of a given set of models. *Lower is better.*

Influence of the Number of Swapped Tokens. We also run an experiment to observe what happens if we only use part of the translated tokens to swap. The vocabulary size is 30 000 tokens. The full vocabulary swap method can map around half (47.57%, or 14,272 tokens out of the total of 30 000 tokens). We wanted to see whether using fewer tokens still yields a reasonable improvement. Figure 3 shows the perplexity vs. the percentage of swapped tokens. Keep in mind that not using any vocabulary swap is equivalent to the embedding reset method, while using all tokens is equivalent to full vocabulary swap, meaning that we swap 47.57% of tokens w.r.t. to the vocabulary size, which is all the tokens that we successfully mapped. As shown in Figure 3, we can see that the proportion of swapped tokens clearly impacts the quality of the model, although it seems that the model quality stops to improve if more than 19,03% tokens w.r.t. vocabulary size are swapped.

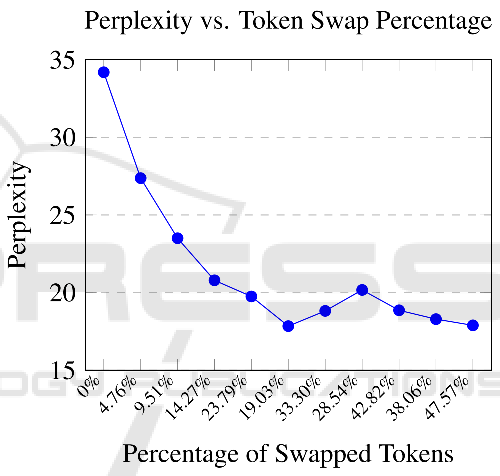

Figure 3: The figure shows perplexity values for different percentages of swapped tokens. The x-axis shows the percentages w.r.t. the total number of tokens (30,000). *Lower is better.*

Figure 4 then shows what happens if we train for more than one epoch. We trained the 4 types of models on 10%, 25%, and 50% splits for three epochs with otherwise the same hyperparameters as previously described. Although the proposed method is still showing the best results, with the increased amount of training, the gap between other methods is getting smaller. It is important to mention that even a marginal difference in perplexity is significant.

5 RELATED WORK

Several contributions have discussed transfer learning in natural language processing. (Hedderich et al., 2020) investigated the application of transfer learning

Table 1: Perplexity values for different training approaches using various data splits. We used the mean value of the perplexities evaluated from 5 independently trained models on 10% and 25% splits. For perplexity results on 50%, 75%, and 100% of data, we trained only 3 models for each split to save computational resources. *Lower is better.*

Method	**Data Splits**				
	10%	**25%**	**50%**	**75%**	**100%**
Naive Fine-Tuning	66.02 ± 0.00	17.50 ± 0.00	10.69 ± 0.00	9.13 ± 0.00	8.13 ± 0.00
Training From Scratch	52.27 ± 1.10	18.18 ± 0.22	11.01 ± 0.10	9.25 ± 0.18	8.18 ± 0.02
Embedding Reset	34.19 ± 0.40	13.19 ± 0.19	9.41 ± 0.02	8.15 ± 0.01	7.61 ± 0.03
Vocabulary Swap	**17.89** ± 0.54	**9.60** ± 0.03	**7.74** ± 0.04	**7.10** ± 0.03	**6.76** ± 0.01

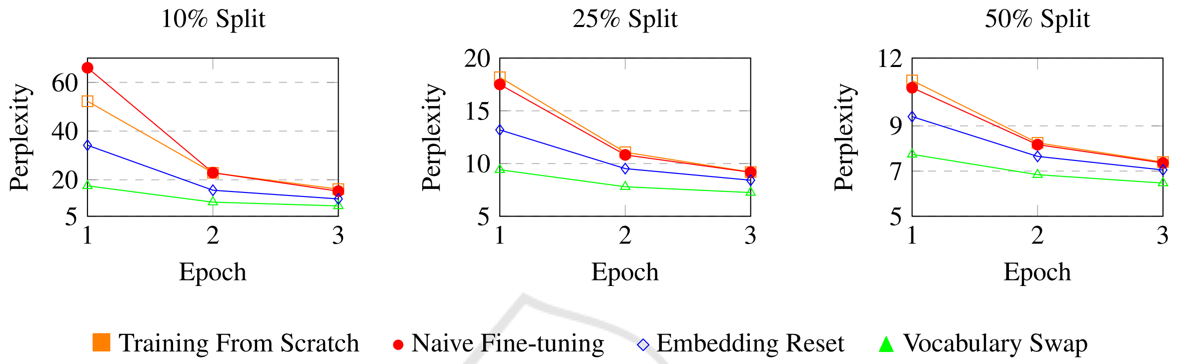

Figure 4: Perplexity across epochs for different data splits and approaches.

and distant supervision techniques to fine-tune multilingual transformer models based on BERT for low-resource African languages.

(Lauscher et al., 2020) shows empirically that linguistic distance in the source and target language plays a role in language transfer, obtaining better evaluation results on downstream tasks when the tested languages are linguistically close. The work shows that, while multilingual models can capture some universal linguistic properties, they are affected by factors such as language similarity, corpus size, and specific model design choices.

(Csaki et al., 2023) proposed novel methods for efficiently adapting pre-trained language models to new languages. A key contribution was their technique for improving tokenizer efficiency by selectively replacing infrequently used tokens from the original vocabulary with tokens from the target language. Importantly, they concluded that adapting an existing pre-trained model was more effective and resource-efficient than training a new model from scratch for low-resource languages.

(Li et al., 2024), demonstrates the advantages of using a parallel corpus, where they employ transfer learning to adapt a Llama 2 model from English to Chinese. Their approach yields superior results on downstream tasks when compared to source models evaluated on standard English and Chinese benchmarks. A significant finding was that their approach of leveraging both corpora showed impressive efficiency, since the previous state-of-the-art model used more than twice the amount of data.

6 LIMITATIONS

There are several limitations known to us, as we developed this method with some favorable conditions that are not usually present in real-world data.

Controlled Environment. TinyStories is a synthetic dataset which contains simple language. Language in the real world is much more complex, and different fields vary in the use of terminology, therefore one could say that one word can have many different semantic meanings. Since we know how the data was created and what it contains, it is much simpler for us to devise functional experiments with simple evaluation on the testing set.

Having a Parallel Corpus. Since the data are synthetic and have a simple language, we could have easily obtained a high-quality translation, thus obtaining a parallel corpus. In the real world, we would not have this benefit. Our method of mapping tokens is directly tied to the availability of a parallel corpus.

Evaluation Metric. Since we use the same tokenizer for each of our models, we can compare the perplexity between models. Comparing different models along with different tokenizers is non-trivial since we would not be able to use the perplexity metric as is. For us, this essentially invalidated any way to compare the proposed method with a method that would use a different tokenizer.

Amount of Training. With increased amount of training, baselines close the gap between each other and the proposed method. Our method works best with a limited amount of training. Training for multiple epochs shows diminishing returns for all baselines considered, as well as the proposed method.

7 CONCLUSION

In this study, we have demonstrated the potential of leveraging pre-trained English language models to effectively adapt and generate coherent text in Czech, a lower-resource language. Our approach, which utilizes a method of vocabulary swap, significantly reduces the computational costs associated with training language-specific models from scratch. Through our experiments, we have shown that even with a small parallel corpus, the adapted model can outperform traditional training methods, highlighting the efficiency of transfer learning in natural language processing. Our method is trivial to implement. It only requires to find a partial mapping between the Czech and English tokenizer and then to initialize the embeddings of Czech tokens to the corresponding embeddings of English tokens. Future experiments will test this method on more realistic datasets.

ACKNOWLEDGEMENTS

This article is from the project "Research of Excellence on Digital Technologies and Wellbeing CZ.02.01.01/00/22_008/0004583" which is co-financed by the European Union. The translation of the parallel corpus, model training and model evaluation was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254). This article has been produced with the financial support of the European Union under the REFRESH – Research Excellence For REgion Sustainability and High-tech Industries project number CZ.10.03.01/00/22_003/0000048 via the Operational Programme Just Transition.

REFERENCES

Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., et al. (2024). Gemini: A family of highly capable multimodal models.

Csaki, Z., Pawakapan, P., Thakker, U., and Xu, Q. (2023). Efficiently adapting pretrained language models to new languages.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, et al. (2024). The llama 3 herd of models.

Eldan, R. and Li, Y. (2023). TinyStories: How small can language models be and still speak coherent english?

Hedderich, M. A., Adelani, D., Zhu, D., Alabi, J., Markus, U., and Klakow, D. (2020). Transfer learning and distant supervision for multilingual transformer models: A study on african languages.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spacy: Industrial-strength natural language processing in python.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, et al. (2023). Mistral 7b.

Jin, Y., Chandra, M., Verma, G., Hu, Y., De Choudhury, M., and Kumar, S. (2023). Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries.

Lauscher, A., Ravishankar, V., Vulić, I., and Glavaš, G. (2020). From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499. Association for Computational Linguistics.

Li, Y., Feng, Y., Zhou, W., Zhao, Z., Shen, L., Hou, C., and Hou, X. (2024). Dynamic data sampler for cross-language transfer learning in large language models.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., et al. (2024). GPT-4 technical report.

Tran, C., Bhosale, S., Cross, J., Koehn, P., Edunov, S., and Fan, A. (2021). Facebook ai wmt21 news translation task submission. *arXiv preprint arXiv:2108.03265*.

Wendler, C., Veselovsky, V., Monea, G., and West, R. (2024). Do llamas work in english? on the latent language of multilingual transformers.