Lite Training Strategies for Portuguese-English and English-Portuguese Translation

Alexandre Lopes[1] **Rodrigo Nogueira**[2,3,4] **Roberto Lotufo**[2,3] **Helio Pedrini**[1]

[1]Institute of Computing, University of Campinas, Brazil
[2]School of Electrical and Computer Engineering, University of Campinas, Brazil
[3]NeuralMind Inteligência Artificial, Brazil
[4]David R. Cheriton School of Computer Science, University of Waterloo, Canada

Abstract

Despite the widespread adoption of deep learning for machine translation, it is still expensive to develop high-quality translation models. In this work, we investigate the use of pre-trained models, such as T5 (Raffel et al., 2019) for Portuguese-English and English-Portuguese translation tasks using low-cost hardware. We explore the use of Portuguese and English pre-trained language models and propose an adaptation of the English tokenizer to represent Portuguese characters, such as diaeresis, acute and grave accents. We compare our models to the Google Translate API and MarianMT on a subset of the ParaCrawl dataset, as well as to the winning submission to the WMT19 Biomedical Translation Shared Task. We also describe our submission to the WMT20 Biomedical Translation Shared Task. Our results show that our models have a competitive performance to state-of-the-art models while being trained on modest hardware (a single 8GB gaming GPU for nine days). Our data, models and code are available at `https://github.com/unicamp-dl/Lite-T5-Translation`.

1 Introduction

With the advent of deep neural networks, results in machine translation have recently improved over classical statistical strategies (Wu et al., 2016; Artetxe et al., 2018). For instance, in the Third and Fourth Conference on Machine Translation (WMT18 (Edunov et al., 2018) and WMT19 (Ng et al., 2019)), the top-performing systems for English-German and German-English competitions were based on transformers (Vaswani et al., 2017).

Transformer models are state-of-the-art architectures for MT tasks and are capable of translating the same word to different words based on the context. For instance, the word 'bank' in Portuguese can be translated to 'bench' or 'bank' depending on the context.

This work explores translation strategies using language models pre-trained on Portuguese and English corpora. More specifically, we investigate the use of **T**ext-to-**T**ext **T**ransfer **T**ransformer (T5) pre-trained model for these tasks. An illustration of T5 for the English-Portuguese translation task is shown in Figure 1. The main contributions of this work are:

- We show that it is possible to train translation models that are competitive with the state of the art using few computational resources. We trained our models on a gaming desktop with an Nvidia RTX2070 GPU, i5 CPU, and 32GB RAM. In comparison, the winning submission of the WMT19 Biomedical Translation Shared Task used four NVIDIA V100 GPUs, each being approximately ten times more expensive than an RTX2070.
- We created and made public ParaCrawl 99k, a dataset of 99k sentence pairs extracted from ParaCrawl's English-Portuguese parallel corpus[1]. This large test corpus allows researchers to evaluate their models on a general-domain translation task.
- We evaluated Google Translate on ParaCrawl 99k, allowing other researchers to compare their results to a high-quality commercial system.
- We developed an adaptation for the English pre-trained tokenizer and achieved better results on English-Portuguese translation tasks than using the tokenizer without any changes. This allows us to efficiently adapt language models to a vocabulary that was not seen during pre-training.

[1] `https://paracrawl.eu/`

2 Related Work

Two widely adopted types of MT systems are Statistical Machine Translation (SMT) systems and Neural Machine Translation (NMT) systems (Sutskever et al., 2014; Bahdanau et al., 2014). The first one relies on statistical techniques to perform translation, such as counting the number of times a word occurs in the context of other words. A popular example of such system is Moses (Koehn et al., 2007).

The winning system of WMT'19 Biomedical competition for en-pt and pt-en translation tasks (Soares and Krallinger, 2019a) is an NMT system. They used OpenNMT-py to train a transformer model on seven parallel corpora. However, differently from our models, their model was trained from scratch.

Recent works (Peters et al., 2018; Devlin et al., 2018) have shown the advantages of using pre-trained models for tasks such as question-answering and text classification. The intuition is to allow the network to use information from pre-training language representations to increase the performance on specific tasks.

Edunov et al. (2019) evaluated the use of a pre-trained encoder-decoder model for translation. Both encoder and decoder weights were tied, but they were pre-trained on different languages. This is an expensive strategy for techniques that use a trainable tokenizer, such as SentencePiece (Kudo and Richardson, 2018), because it is necessary to re-train the entire model if the vocabulary changes, as new token embeddings need to be learned.

Many commercial systems, such as Google Translate (GT) and Amazon Translate (AT), have an excellent performance on MT, but they are expensive if one needs to translate vast amounts of text. For example, we estimate that it would cost 50,000 USD to translate the 20 million sentences of ParaCrawl using GT. Unfortunately, no commercial system that we are aware of provides metric scores on public datasets that would allow us to compare their systems to ours.

3 Methods

We proposed two main strategies for translating: using a T5 model pre-trained on a Portuguese corpus and adapting the original T5 tokenizer to work with Portuguese texts.

3.1 Pre-trained Language Model

We evaluated four different scenarios: English-Portuguese translation with T5 pre-trained in a Portuguese corpus; English-Portuguese translation with T5 pre-trained in an English corpus; Portuguese-English translation with T5 pre-trained in an English corpus; and Portuguese-English translation with T5 pre-trained in a Portuguese corpus.

These variations allow us to evaluate how the language used during pre-training affects the translation's performance.

3.2 Adaptation of the English tokenizer to Portuguese

Here we investigate if we can adapt to the English-Portuguese translation task a model already pre-trained on languages other than Portuguese.

We observe that using a non-Portuguese tokenizer can cause translation problems, since some Portuguese characters cannot be represented, such as letters with the tilde accent (e.g. 'ã'). To fix this issue, we propose an adaptation of the original T5 tokenizer using a pre-processing and post-processing strategy. The tokenizer's adaptation allows the tokenizer to represent all possible characters in the Portuguese language.

We can divide this adaptation into two stages: Token Completion and Word Regrouping. The first stage allows the use of Portuguese special characters, such as accented vowels, whereas the second stage merges these extra tokens back to form correct words.

3.2.1 Token Completion Stage

In this step, we start adding to the tokenizer Portuguese accented vowels that were not present in it. We ended up adding fourteen of those characters, as well as the word 'não', which is the most common word in the ParaCrawl pt-en dataset.

A list of all added tokens is available in Table 1. The addition of these tokens allowed the model to learn and generate them in en-pt translation.

This is also an inexpensive method for increasing the number of words that can be represented, since only the embeddings of the new tokens have to be learned from scratch. The existing token embeddings, which represent the majority of the non-Portuguese tokens, were already learned during the pre-training phase and can be reused in the fine-tuning phase.

We show in Table 2 some encoding and decoding examples after adding tokens to the tokenizer.

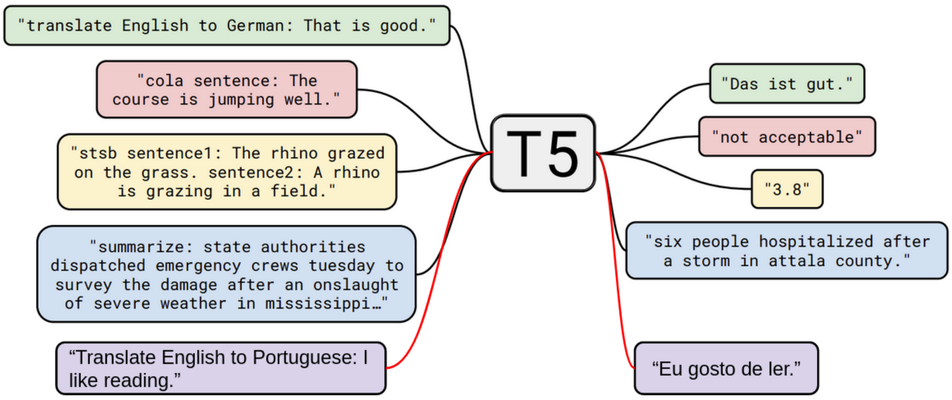
"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
"Translate English to Portuguese: I like reading."
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."
"Eu gosto de ler."

Figure 1: The text-to-text framework used by T5. The purple boxes and red connections represent the task used in this work. Figure adapted from (Raffel et al., 2019).

ì ò Á Í Ó Ú í ú Â Ê Ã Õ ã õ não

Table 1: List of tokens added to the T5 tokenizer by our adaption method.

Tokenizer without additional Port. tokens
original → after encoding/decoding
eu gosto de arroz → eu gosto de arroz
eu não como → eu n ? o como
indignação completa → indignaç ? o completa
Tokenizer with additional Port. tokens
original → after encoding/decoding
eu gosto de arroz → eu gosto de arroz
eu não como → eu não como
indignação completa → indignaç ã o completa

Table 2: Comparing tokenizer results before and after adding the Portuguese tokens.

3.3 Word Regrouping Stage

When adding tokens directly to the tokenizer, the HuggingFace's (Wolf et al., 2019) SentencePiece implementation used in our work interprets the result as a new complete token, i.e., not part of a word. For example, the word 'pão' is broken into three different tokens 'p' 'ã' 'o'. This is fixed in a post-processing step called Word Regrouping.

In this step, we regroup the added tokens of vowels with accents separated erroneously by the tokenizer. We find in the translated text all tokens added in the Token Completion step, and merge them with their neighboring words.

In Figure 2, we illustrate our algorithm.

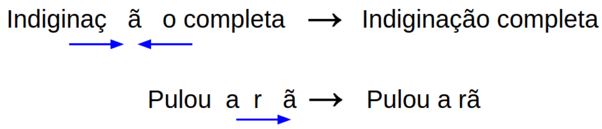

Figure 2: An example of separated tokens merged back into a single word. Our algorithm searches for an isolated special token (in this case, 'ã') and merges it with its neighbors. It can be merged at the beginning, middle, or end of a sentence.

4 Datasets

We trained our models using six different datasets, and we evaluate our system on two datasets: the WMT19 Biomedical Translation Task dataset and a subset of 99,000 sentence pairs of the ParaCrawl dataset. We also present the results of our submission to the WMT20 Biomedical Translation Task competition.

4.1 Training Datasets

We have two different strategies for training our models depending on the test datasets. For the evaluation on the ParaCrawl dataset, we only trained the models on ParaCrawl data. ParaCrawl is a public parallel corpus of many European languages available online. It contains approximately 20M English-Portuguese sentence pairs. Due to our small computational budget, we randomly selected approximately 5M pairs for training.

For WMT19 and WMT20 Biomedical Translation Tasks, we train our models on the ParaCrawl dataset as well as on the following datasets, which are of the same domain as WMT's Biomedical data:

- EMEA Corpus (Tiedemann, 2012): A parallel corpus of European Medicines Agency documents.

- CAPES Parallel Dataset (Soares et al., 2018b): A parallel corpus of theses and dissertations abstracts collected from the CAPES website.

- Scielo Dataset (Soares et al., 2018a): A parallel corpus of scientific articles collected from SciELO.

- JRC-Acquis (Steinberger et al., 2006): A parallel corpus of European Union (EU) documents in all official EU languages.

- Biomedical Domain Parallel Corpora (Névéol et al., 2018): A repository of the challenge that contains links to different parallel corpora. We used the Medline, Scielo, and ReBEC training datasets.

Besides being of the same domain of WMT's Biomedical task, an advantage of these datasets over ParaCrawl is that they are in Brazilian Portuguese, such as most of WMT's Biomedical data. The number of sentence pairs used for training from each dataset is shown in Table 3.

Corpus	**Sent. Pairs**
EMEA	1,082,144
CAPES	1,157,610
Scielo	2,828,916
JRC-Acquis	1,236,846
Biomedical Domain Corpora	331,937
Total	**6,637,453**

Table 3: Number of sentence pairs of each domain-specific dataset used to train our models for the WMT19 and WMT20 Biomedical tasks.

4.2 Testing Datasets

We created a general-domain test set from the ParaCrawl dataset. We begin by randomly selecting 128,000 sentence pairs from its 20M pairs. ParaCrawl is originally deduplicated, but similar sentences still might exist in our split of the training and test sets. Thus, we apply as stricter deduplication process to increase the quality of our test set. To increase the speed in verifying similarity of sentence pairs, we used MinHash and Locality-Sensitive Hashing (LSH) (Rajaraman and Ullman, 2011) to compare sentences of training and test datasets. We set a Jaccard similarity threshold to 0.7, i.e., all sentences with similarity greater than 0.7 were discarded from the test set. LSH found 28,913 sentences in the test set with a similarity score above 0.7 of sentences in the training set. The final test set ended up having 99,087 sentence pairs, which we called ParaCrawl 99k test set. This dataset and its corresponding translations using GT are available in our Github.

We also evaluated our system on the WMT19 Biomedical Shared Task test set. This is a dataset composed of approximately 500 parallel sentences of Medline abstracts.

Finally, we submitted our results to the WMT20 Biomedical Shared Task competition. The WMT20 test set has 544 parallel sentences for the English-Portuguese translation task and 498 sentences for the Portuguese to English task.

5 Experiments

We conducted several experiments using different configurations of T5. We divided the experiments into two groups: model hyperparameter optimization and different pre-training studies. All experiments were performed on a desktop computer with an Nvidia 8GB RTX 2070 Super, 32 Gb RAM memory, and a 4-core Intel processor running on Ubuntu 18.04. We used PyTorch (Paszke et al., 2017), HuggingFace Transformer, and Pytorch-Lightning (Falcon, 2019) frameworks to train and evaluate our models.

5.1 Model Hyperparameter Optimization

We tuned the hyperparameters using the original T5 checkpoint available in the HuggingFace library. This model was pre-trained on a corpus whose majority of documents were in English with a small proportion of German, French, and Romanian documents. We first conducted a small training using 100k sentence pairs and evaluated on another 50k sentence pairs to determine some hyperparameters of the T5 model, such as batch size and maximum length of tokens in the source and target sentences. We also evaluated the optimizer and found the best convergence with the AdamW Optimizer (Loshchilov and Hutter, 2017). All hyperparameters used are in Table 4. With this configuration, we evaluated the performance of adding Portuguese-only characters to the tokenizer in comparison to using the original T5 tokenizer. The results are available in Table 5. Our proposed tokenizer adaption resulted in an improvement of almost 5 BLEU points over the original tokenizer in the en-pt translation task.

Hyperparameters	Values
Batch Size	256
Source Sequence Length (SSL)	96
Target Sequence Length (TSL)	160
Learning Rate	$5 \cdot 10^{-3}$
eps	$1 \cdot 10^{-5}$

Table 4: Hyperparameters used for training the models.

After finding these hyperparameters, we analyzed the trade-off between model sizes in a subset of the ParaCrawl dataset of 1M sentence pairs and evaluated them in a 150k sentence subset. We did not use any sentence from the test set. The results of this analysis are reported in Table 6. We trained the T5-small and T5-base models with different epoch sizes. Training 3 epochs of T5-small takes almost the same time as training one epoch with a T5-base model.

The performance would possibly increase if we used large models such as T5-large, T5-3B, or T5-11B. However, we could fit only the T5-base model in our 8GB GPU. We used batch accumulation to achieve batches of size 256 as the T5-small can only handle batch size 4 in 8GB. Thus, one of the contributions of this work is to show that it is possible to train translation models that are close to the state of the art on a relatively inexpensive hardware.

All experiments in the following sections using Tokenizer's Adaptation Steps (3.2) were performed using the best pre-processing and post-processing strategies presented in Table 6.

5.1.1 Pre-training Studies

We also evaluated the effects of pre-training the model in a corpus of the same language of the target language. The intuition here is that it would be easier for the model to learn the target language than having previous knowledge of the source language.

Translation Type	SacreBLEU
Original T5 tokenizer	31.15
+ Portuguese characters	35.95 (+4.8)

Table 5: Effects in performance of using our adaption of the original T5 tokenizer to the English-Portuguese translation task. Numbers are from ParaCrawl's 99k en-pt test set.

Translation Type	Sacre BLEU
Adding Top 25 words in Port. + T5-small + 3 epochs	43.03
Adding tokens of Table 1 in Port. + T5-small + 3 epochs	43.48
Adding tokens of Table 1 in Port. + T5-base + 1 epoch	**44.52**

Table 6: Effects in performance of different strategies for adapting the original T5 tokenizer to Portuguese. Numbers are from our dev set of ParaCrawl.

Since the tokenizer mainly has tokens of one of the two languages, it is better to have a smaller quantity of tokens to learn. This is because, if the SentencePiece tokenizer does not have the word in its vocabulary, it will use subtokens to form the original word. For example, the sentence 'They like to drink coconut water' is represented by six tokens in English SentencePiece and thirteen tokens in Portuguese SentencePiece. We are not evaluating here the possibility to train the pre-training model from scratch with both languages together, as it is not possible with our modest hardware setup.

For the Portuguese pre-trained model, we used PTT5-base model (Carmo et al., 2020) with Portuguese tokenizer. PTT5 was pre-trained on BrWAC, a large corpus of Brazilian Portuguese webpages. PTT5 started training using T5's official published weights as initial weights, so it also uses English learning to its model. For the English pre-trained model, we used the Huggingface implementation of T5 with its default tokenizer, which is based on SentencePiece.

In Table 7, we compare both models with Google Translate in the ParaCrawl 99k test set. Both models perform similarly in the Portuguese-English translation task, but the Portuguese pre-trained model gives a better result than the English pre-trained model in the English-Portuguese translation task. We are on par with Google Translate on en-pt, but a few BLEU points below on pt-en.

	pt-en	en-pt
Google Translate API	51.20	45.17
Ours - English pre-training	46.49	44.56
Ours - Portuguese pre-training	46.35	45.44

Table 7: SacreBLEU comparison between GT and our approach in Paracrawl 99k test set.

6 WMT19 and WMT20 Results

We now evaluate our models on the WMT19 Biomedical Translation Task and our show the results of our official submission to the WMT20 Biomedical Translation Task.

In Table 8, we show WMT19 results of our models as well as the winning submission of WMT19 Biomedical tasks (Soares and Krallinger, 2019b) and the MarianMT (Junczys-Dowmunt et al., 2018) implementation available on the HuggingFace's Transformer Library.[2] Models pre-trained on Portuguese obtained the best performance in both translation tasks. Notably, we achieved an improvement of +6.31 BLEU points in the English to Portuguese translation task by using the Portuguese pre-trained model and +9.75 with an increase of target and source sequence lengths. We also obtained an improvement of +0.62 in the Portuguese to English translation task using the Portuguese pre-trained model and +2.27 when increasing target and source sequence lengths.

We believe that the improvement of Portuguese pre-training models is associated with PTT5's training strategy that uses English pre-trained weights as initial weights. The intuition is that PTT5 carries information from the English model too.

	pt-en	**en-pt**
MarianMT	27.91	47.44
BSC (Soares and Krallinger, 2019b)	39.90	48.18
Ours - English pre-training	45.89	39.31
Ours - Portuguese pre-training	46.51	45.62
+ TSL=256 and SSL=256	–	**49.06**
+ TSL=140 and SSL=160	**48.16**	–

Table 8: BLEU scores on the test set of WMT19 Biomedical Shared Task. Portuguese pre-training was tested in three different scenarios: one with default hyperparameters available in Table 8 and two with different Target Sequence Length (TSL) and Source SEquence Length (SSL).

The results for WMT20's challenge are in Table 9. Our submission is 2.17 BLEU points below the winning team in Portuguese-English, but it is 4.48 BLEU points higher than the baseline. For the English-Portuguese task, our results are below the baseline. That can be attributed to not using the Portuguese pre-trained model, which was not available at the time of our submission. As noted above, we achieved a large improvement on WMT19 when we switched from the English to the Portuguese pre-trained model. Therefore, we assume that a Portuguese pre-trained model would obtain superior results to the baseline on WMT20.

Team Names	**pt-en**	**en-pt**
Sheffield	48.16	44.57
Unicamp_DL	45.99	38.08*
baseline	41.51	39.77

Table 9: BLEU scores on WMT20's automatic evaluation. *Since the Portuguese T5 model was not available at the time of our submission, we used the original (English) T5. Hence, results for en-pt can now be improved by switching to the Portuguese pre-trained model.

7 Conclusions and Future Work

We show that it is possible to develop English-Portuguese translation models close to the state of the art using modest hardware. Despite not reaching the same level of performance of Google Translate on pt-en, the fact that our system was developed mostly by the first author on its personal computer shows that implementing high-quality machine translation systems has become possible for anyone, including small companies and research labs.

We also presented our submission strategies for the WMT20 Biomedical Translation Shared Task using a T5 model. We show that a simple adaption of the original T5 tokenizer to the Portuguese language largely improves the translation quality and does not require any further pre-training, which is expensive. However, we achieve the best results with models pre-trained on Portuguese.

As directions for future work, we plan to experiment with larger models and models pre-trained in both Portuguese and English languages simultaneously, as recent work showed that this a successful strategy (Wu et al., 2016; Arivazhagan et al., 2019). We believe that we could improve the translation results with larger and more complex models (Lepikhin et al., 2020).

8 Acknowledgements

We thank CNPq research funding, process number 310828/2018-0.

[2] https://huggingface.co/transformers/model_doc/marian.html

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, and Colin Cherry. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Diedre Carmo, Marcos Piau, Israel Campiotti, Rodrigo Nogueira, and Roberto Lotufo. 2020. Ptt5: Pre-training and validating the t5 transformer in brazilian portuguese data. `https://github.com/unicamp-dl/PTT5`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, pages 1–16.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4052–4059, Minneapolis, Minnesota. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

WA Falcon. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning Cited by*, 3.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, and Nikolay Bogoychev. 2018. Marian: Fast neural machine translation in C++. *arXiv preprint arXiv:1804.00344*, pages 1–6.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Aurélie Névéol, Antonio Jimeno Yepes, Mariana Neves, and Karin Verspoor. 2018. Parallel corpora for the biomedical domain. In *Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, pages 1–67.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of Massive Datasets*. Cambridge University Press.

Felipe Soares and Martin Krallinger. 2019a. BSC participation in the WMT translation of biomedical abstracts. In *Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 175–178, Florence, Italy. Association for Computational Linguistics.

Felipe Soares and Martin Krallinger. 2019b. BSC Participation in the WMT Translation of Biomedical Abstracts. In *Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 175–178.

Felipe Soares, Viviane Moreira, and Karin Becker. 2018a. A large parallel corpus of full-text scientific articles. In *Eleventh International Conference on Language Resources and Evaluation*.

Felipe Soares, Gabrielli Harumi Yamashita, and Michel Jose Anzanello. 2018b. A parallel corpus of theses and dissertations abstracts. In *International Conference on Computational Processing of the Portuguese Language*, pages 345–352. Springer.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Fifth International Conference on Language Resources and Evaluation*, Genoa, Italy. European Language Resources Association (ELRA).

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Eight International Conference on Language Resources and Evaluation*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.