CycleGN: a Cycle Consistent approach for Neural Machine Translation training using the Transformer model in a shuffled dataset

Anonymous ACL submission

Abstract

CycleGN is a Transformer architecture using a Discriminator-less CycleGAN approach, specifically tailored for training Machine Translation models utilizing non-parallel datasets. Despite the widespread availability of large parallel corpora for numerous language pairs, the capacity to employ solely monolingual datasets would substantially expand the pool of training data. This approach is particularly beneficial for languages with scarce parallel text corpora.

The foundational concept of our research posits that in an ideal scenario, translations of translations should revert to the original source sentences. Consequently, we can simultaneously train a pair of models using a Cycle Consistency Loss framework. This method bears resemblance to the technique of back-translation, prevalently employed in Machine Translation, where a pre-trained translation model is used to generate new examples from a monolingual corpus, thereby artificially creating a parallel dataset for further training and refinement.

1 Introduction

The introduction of the Transformer architecture (Vaswani et al., 2017) marked a significant advancement in the field of Machine Translation, witnessing widespread adoption since its inception. Although self-attention mechanisms were not novel and had been investigated in prior studies (Bahdanau et al., 2016), the Transformer model demonstrated its formidable capabilities within Natural Language Processing (NLP). Characterized by its parallelized structure, the Transformer architecture facilitated computational efficiency, enabling the incorporation of a larger number of parameters. This enhancement has been exemplified in NLP systems like Charles University Block-Backtranslation-Improved Transformer Translation (cubbitt) (Popel et al., 2020), which have surpassed the performance levels of human professionals in certain contexts.

Neural Machine Translation (NMT) datasets necessitate substantial text corpora, structured as aligned pairs. This alignment implies the requirement for sentences with equivalent meaning to be present in a minimum of two distinct languages, enabling the initiation of model training to forge linguistic linkages. Ongoing initiatives, including OPUS (Tiedemann and Thottingal, 2020) and Tatoeba (Tiedemann, 2012), are committed to facilitating public access to these datasets. Clearly, parallel datasets comprise a small subset of the volume of data in monolingual datasets.

Despite the remarkable efficacy exhibited by Large Language Models (LLMs) in MT (Machine Translation) without the necessity of exclusive training on parallel data (Zhu et al., 2023), their considerable magnitude renders them costly in terms of both training and operation. This economic burden consequently restricts their widespread availability.

Back-translation (Sennrich et al., 2016) is a technique leveraging a trained MT model to translate sentences from a monolingual dataset to produce corresponding pairs, thereby synthetically augmenting the training data. Our research is founded on the premise that the process of translating a sentence from a source language to a target language, followed by its retranslation from the target language back to the source language, allows for the measurement of the disparity between the original and the machine-translated sentences. This disparity serves as a metric to assess the efficacy of the models and facilitates the backpropagation of gradients within the networks. Notably, this methodology has been previously implemented in the realm of Image-to-Image Translation, as evidenced in the renowned CycleGAN study from Zhu et al. (2017).

2 Previous work

The TextCycleGAN model (Lorandi et al., 2023), while not utilizing the Transformer architecture nor operating within the MT field, introduced an innovative strategy for text style transfer. This approach employed a CycleGAN on the Yelp dataset to facilitate the learning of mappings between positive and negative textual styles, notably in the absence of paired examples.

Shen et al. (2017) exemplified the feasibility of training two encoder-decoder networks in an unsupervised manner that enables the sharing of a latent space, thereby permitting style transfer. Lample et al. (2018), adopting a similar technique within the MT context, substantiated that the use of parallel datasets is not a prerequisite for effective translation.

3 Dataset

In the context of the current study, a "shuffled" dataset is defined as a parallel dataset wherein the sentences of one language have been systematically rearranged. Consequently, this results in a non-parallel corpus where it is guaranteed that each sentence has a corresponding translation located at an unspecified index within the dataset. The authors postulate that when employing sufficiently large monolingual datasets, which are not derived from shuffled parallel corpora, it is likely that most sentences will possess an accurate translation "somewhere" within the dataset.

For the purposes of this research, a shuffled dataset was utilized in lieu of a monolingual dataset. This choice was made to facilitate a direct comparison of our approach with conventional NMT training, employing an identical non-shuffled parallel dataset and the same model architecture.

The dataset employed in this study is the English-German language pair from the WMT23 challenge (Kocmi et al., 2023). Specifically, only the first half of this dataset was used for training, due to the current implementation's high computational demands. This amounts to a total of approximately 27 million sentences. The data released for the WMT23 General MT task can be freely used for research purposes.

4 Training

For greater clarity, the mathematical notations from the original CycleGAN work will be employed in the present study. Given two languages $\mathcal{X}$ and $\mathcal{Y}$ with appropriate datasets, our objective is to obtain two NMT models $\mathcal{G} : \mathcal{X} \mapsto \mathcal{Y}$ and $\mathcal{F} : \mathcal{Y} \mapsto \mathcal{X}$ such that for $x \in \mathcal{X}$, $\mathcal{G}(x) = \hat{y}$, for $\hat{y} \in \mathcal{Y}$ and that for $y \in \mathcal{Y}$, $\mathcal{F}(y) = \hat{x}$, for $\hat{x} \in \mathcal{X}$. If the translations are perfect, $\mathcal{G}(\mathcal{F}(y)) = y$ and $\mathcal{F}(\mathcal{G}(x)) = x$. By using the Cross-Entropy Loss (CEL) (Zhang and Sabuncu, 2018) in the role of the Cycle Consistency Loss (CCL), we can determine the distance between the original sentence and its double translation in order to compute the gradients.

As in the original CycleGAN work, our current study also implements an Identity Loss (IL), which relies on the CEL, to help with the training stability. As $\mathcal{G}$ consists in a mapping $\mathcal{X} \mapsto \mathcal{Y}$, if given an input $y \in \mathcal{Y}$, we want to obtain an unchanged output such that $\mathcal{G}(y) = y$. The same is applied to $\mathcal{F}$, where we also compute the IL between $\mathcal{F}(x)$ and x. See Figure 1.

4.1 Obtaining labels

In the training process of a Transformer model, it is imperative to have prior knowledge of the labels, as the decoder predicts tokens sequentially. Each token prediction, barring the initial one, is contingent upon all preceding predictions. The act of selecting the most probable token constitutes a non-differentiable operation, thus precluding the possibility of backpropagation. By possessing prior knowledge of the reference translation, it becomes feasible to contrast each predicted token against the ground truth, enabling the calculation of loss at every step.

Teacher Forcing (Gers et al., 2002) is a technique that involves substituting the predicted token with the actual ground truth at each stage of the decoding process. This approach is designed to mitigate the cascading impact of early erroneous predictions in the sequence.

The CycleGN training process used here consists in a cooperation between $\mathcal{G}$ and $\mathcal{F}$. The first step is to generate $\hat{x}$ and $\hat{y}$, since labels are not required during inference, as backpropagation is unnecessary. Even though this step cannot be used to compute the gradients, it is crucial for the entire process. From $\mathcal{G}(\mathcal{F}(y)) = y$ and $\mathcal{F}(\mathcal{G}(x)) = x$, it follows that the label for $\hat{y}$ is x and the label for $\hat{x}$ is y. We can compute $\hat{\hat{x}}$ from $\mathcal{F}(\hat{y})$ with x as the label, and $\hat{\hat{y}}$ from $\mathcal{G}(\hat{x})$ with y as the label, and use the CCL between $\hat{\hat{x}}$ and x, and between $\hat{\hat{y}}$ and y to compute the gradients and backpropagate.

4.2 A Discriminator-less GAN

The CycleGAN methodology, as indicated by its nomenclature, is predicated on the Generative Adversarial Network (GAN) framework, initially

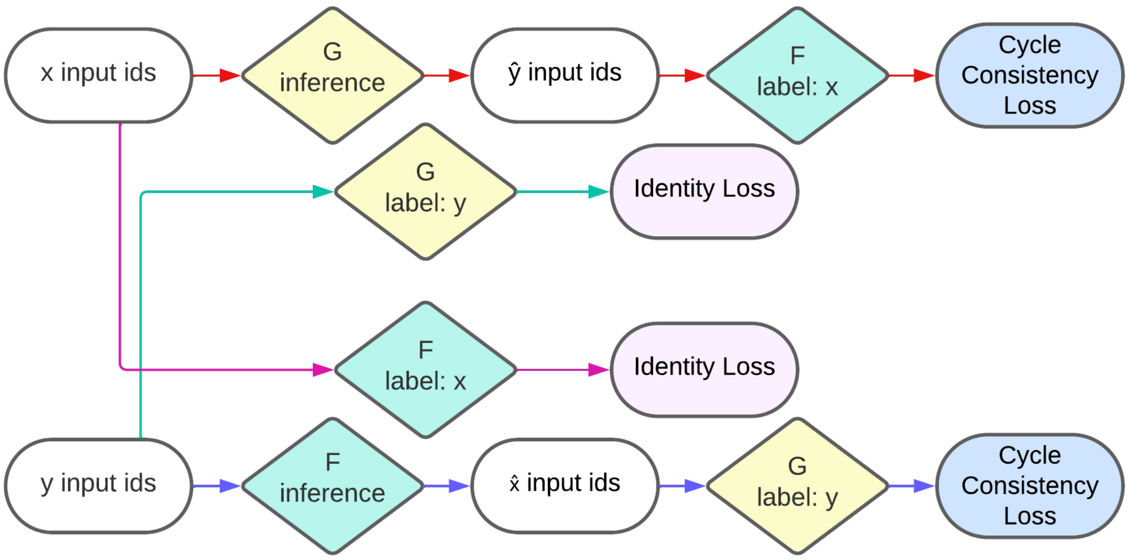

Figure 1: CycleGN training process.

introduced in Goodfellow et al. (2014). This paradigm involves the training of a Generator model in conjunction with another model, termed the Discriminator. The Discriminator is specifically trained to distinguish between authentic samples drawn from the dataset and synthetic samples produced by the Generator. In the CycleGAN training process, the Discriminators intervene after the generation of $\hat{x}$ and $\hat{y}$, helping the training of the Generators. However, as mentioned in Section 4.1, there can be no gradient computation during the generation of $\hat{x}$ and $\hat{y}$ in a transformer model and as such, Discriminators cannot be used in the present work. This is why CycleGN is not an "Adversarial" approach, hence the name.

5 Model architecture

The architecture used for both models, $\mathcal{G}$ and $\mathcal{F}$, is the Marian framework (Junczys-Dowmunt et al., 2018) implemented by Huggingface's Transformers library (Wolf et al., 2020), which is licenced under the Apache Licence. While most parameters follow the default configuration, Table 1 references the changes that were made in order to reduce the computational cost of the architecture.

6 Vocabulary organization

Sequence2Sequence models employ either a unified tokenizer or two distinct tokenizers. In the case of a single tokenizer, it is trained using sentences from both the source and target distributions, avoiding any duplicates. This approach facilitates the sharing of the encoder and decoder embedding layers, thereby diminishing computational demands and enhancing model accuracy (Press and Wolf, 2017).

Parameter	Huggingface	Current work
Vocabulary size	58,101	32,000
Encoder layers	12	6
Decoder layers	12	6
Encoder attention heads	16	8
Decoder attention heads	16	8
Encoder feed-forward	4096	2048
Decoder feed-forward	4096	2048
Position embeddings	1024	128
Activation function	GELU	ReLU

Table 1: Non-default parameters in the configuration of Marian Transformer models

Conversely, the alternative approach entails training one tokenizer on the source distribution and another one on the target distribution. While this method restricts the possibility of tying embeddings, it can potentially double the vocabulary size. The overall vocabulary size of the model in this scenario, is the cumulative total of the two individual vocabularies, barring shared tokens like punctuation symbols.

While contemporary Transformer models like Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) and Generative Pre-trained Transformers (GPT) (Radford et al.,

2018) typically utilize a single tokenizer, this study introduces a novel vocabulary methodology that amalgamates the aforementioned approaches. This method involves training two tokenizers, each for a respective language and with half the vocabulary size. Subsequently, the identifiers of one tokenizer are adjusted to prevent overlap, yielding a result analogous to a single tokenizer that includes duplicates across languages. It is important to note that special tokens such as $< eos >$ (End of Sentence) and $< pad >$ (Padding) are shared and not duplicated. This strategy is designed to simplify model analysis during development, albeit at the expense of a reduced vocabulary.

7 Pretraining

The CycleGN approach requires a pre-training step, as it will not converge at all without it. Indeed, as there is no Discriminator to ensure that $\hat{x}$ belongs to $\mathcal{X}$ and $\hat{y}$ belongs to $\mathcal{Y}$, $\mathcal{G}$ and $\mathcal{F}$ can converge towards identity matrices. That is, if both $\mathcal{G}$ and $\mathcal{F}$ do not apply any change to their input, they can still achieve $\mathcal{G}(\mathcal{F}(y)) = y$ and $\mathcal{F}(\mathcal{G}(x)) = x$ without learning how to translate.

Masked Language Modeling (MLM) is a pre-training strategy first implemented in BERT, wherein a specified proportion of tokens within the input text are substituted with a unique $< mask >$ token. The objective of the neural network under this paradigm is to accurately reconstruct the original sentence. This process enables the model to discern intricate relationships between words and to develop a profound representation of the language. This pre-training has revealed excellent performances in diverse NLP application such as sentiment analysis (Alaparthi and Mishra, 2021), text classification (Sun et al., 2020), Named Entity Recognition (NER) (Souza et al., 2020) (Chang et al., 2021) (Akhtyamova, 2020) and paraphrase detection (Khairova et al., 2022).

As MLM does not require any labeling, it is perfectly adapted to the CycleGN approach. A single model $\mathcal{H}$ is trained on the entire dataset for a single epoch to reconstruct both languages, with 15% of the input tokens masked. When training the CycleGN, rather than randomly initializing $\mathcal{G}$ and $\mathcal{F}$, the parameters from $\mathcal{H}$ are directly copied to $\mathcal{G}$ and $\mathcal{F}$. Indeed, as $\mathcal{H}$ learns to reconstruct both language $\mathcal{X}$ and $\mathcal{Y}$, it can be used to initialize both networks. Figure 2 shows the training process of $\mathcal{H}$.

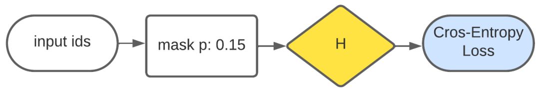

Figure 2: Masked Language Modeling training process.

8 Batch size

The original CycleGAN research mentions using a batch size of one, and while they did not state the reason in the research paper, one of the authors explained it in a GitHub issue (Junyanz, 2017) as a lack of GPU memory.

Rajput et al. (2021) examined the impact of batch size within the CycleGAN architecture, observing a significant decline in performance with its increase. This deterioration was evident both through the example images presented in that study and through the calculated cosine dissimilarity, indicating inferior model performance with larger batch sizes. However, quality was achieved at the expense of computational efficiency, as the training duration to achieve 200 epochs was 8 hours with a batch size of 1, but this was reduced to just 2 hours with a batch size of 64.

In the context of our research, however, the trade-off between quality improvement and computing resource, as observed in the aforementioned study, does not hold true. Utilizing a batch size of 1 in our experiments hindered any form of convergence. Consequently, a batch size of 16 was selected, as it represented the maximum capacity that could be accommodated within the available 24GB of GPU memory of the Nvidia 4090 used for this work.

9 Training stability

It is crucial for a CycleGAN architecture that the two models exhibit approximately equivalent levels of performance. Given the interdependent nature of these models, where the output of one serves as the input for the other, maintaining consistency between them during training is imperative. Without a strategy in place to prevent the performance of the models from diverging, it is possible for one model to gain the "upper hand" over the other.

9.1 Divergence between the Generators

Figure 3 presents the evolution of the CCL of an early prototype of CycleGN and it can clearly be seen that one of the two generators, $\mathcal{F}$, ends up performing much better than its counterpart $\mathcal{G}$, which blocks any future training.

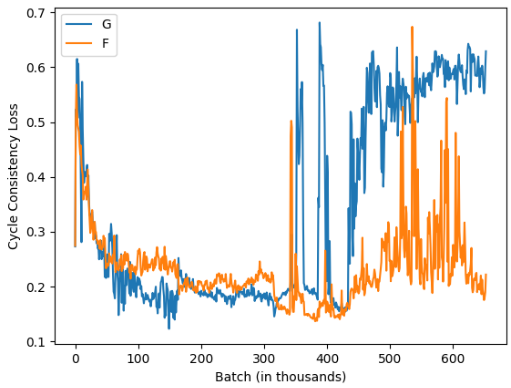

Figure 3: Evolution of the Cross-Entropy Loss during the training of an early prototype.

9.2 Gradient Clipping

Gradient clipping is a technique utilized in the training of Deep Learning (DL) models, to address the problem of 'exploding' gradients. This issue occurs when gradients escalate to excessively high values during training, leading to numerical instability and impeding the model's convergence to an optimal solution.

Gradient clipping can be implemented through two primary methods: norm clipping and value clipping. Norm clipping involves establishing a threshold on the overall magnitude of the gradient vector. On the other hand, value clipping involves individually adjusting elements of the gradient vector that exceed the specified threshold.

By clipping the gradients by norm, with a threshold of 1.0, as advised by the Huggingface library, the training stabilized and the divergence between $\mathcal{G}$ and $\mathcal{F}$ was observed to disappear.

10 One large epoch or multiple smaller ones?

The CycleGAN framework is recognized for its computational intensity due to several inherent factors. Primarily, as CycleGAN operates on the principle of cycle consistency, it necessitates the training of two GANs simultaneously – one for each direction of the transformation. This structure requires substantial computational resources, as each GAN consists of both a Generator and a Discriminator.

The resource-intensiveness of the CycleGAN process, thus limits the size of the dataset that can be used in a reasonable time. This necessitated a decision between training for a single epoch on a large dataset, or training for multiple epochs on a smaller corpus arose.

We compared the CycleGN model on the entire dataset under four different conditions:

1. One epoch containing 1% of the dataset
2. Five epochs containing 0.2% of the dataset
3. One epoch containing 2% of the dataset
4. Five epochs containing 0.4% of the dataset

We have selected the Crosslingual Optimised Metric for Evaluation of Translation (COMET) score, as proposed by Rei et al. (2020), as our comparison criterion. This metric has proven to be one of the most effective in recent WMT competitions, according to Kocmi et al. (2022), due to its strong correlation with human judgment, aligning well with our goal of mirroring human evaluative standards. Multiple COMET models have been made available and we chose the default "wmt22-comet-da" model. The average scores obtained on 10,000 sentences that were not part of the model training set are presented in Table 2.

Condition	English->German	German->English
1	0.2727	0.2715
2	0.2411	0.2635
3	0.2741	0.2665
4	0.2258	0.2658

Table 2: COMET scores of CycleGN models depending on the dataset condition.

Models exposed to a larger portion of the total dataset demonstrate superior performance compared to those limited to a smaller, repetitive subset, especially when the dataset encompasses over half a million to a million sentences. We extrapolate this result to larger datasets and thus chose to train our model for a single epoch on the largest dataset possible.

11 Results

To measure the performances of CycleGN, every 1000^{th} batch the CCL was averaged and 1,000 sentences from the test set were translated to compute the COMET score.

Figure 4 demonstrates how the addition of gradient clipping helps with training stability.

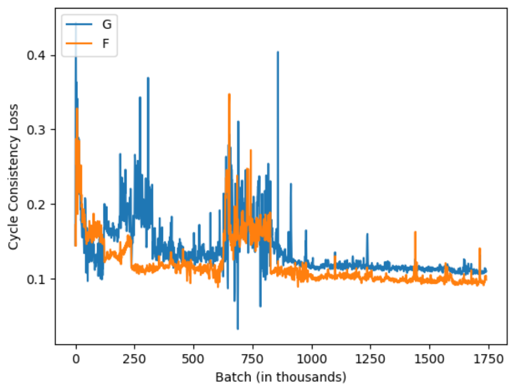

Figure 4: Evolution of the Cross-Entropy Loss during the training.

11.1 Translation quality

Even if tracking the CCL is an inexpensive manner to estimate the progress of the training of the CycleGN architecture, as mentioned in Section 7, it can also hide an absence of translation. Figure 5 demonstrates that the actual quality of translation, as measured by the COMET metric, increases with time. Note that the sudden drop is discussed in the next section.

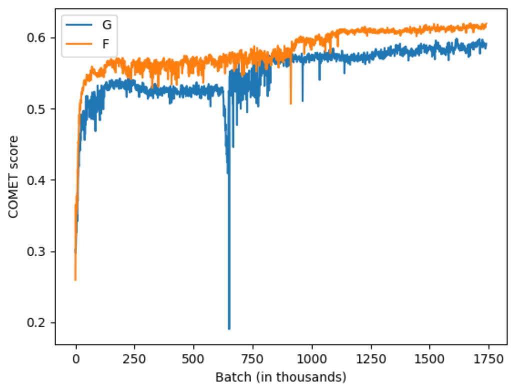

Figure 5: Evolution of the COMET score during the CycleGN training.

After the end of the training, a test set of 10,000 sentences per language were translated and the COMET scores are displayed in Table 3.

	English->German	German->English
Score	0.505	0.537

Table 3: COMET score of CycleGN models.

As mentioned in Section 3, in order to give a point of comparison, we trained a couple of architecture-matched models using the parallel dataset. As in the case of the CycleGN training, these models were only trained for a single epoch on the first half of the WMT23 English-German language pair. Results are displayed in Table 4. We fully expected the COMET score of the CycleGN to be inferior to architectures using parallel corpora, but we believe the differences between the scores will reduce with larger datasets.

	English->German	German->English
Score	0.780	0.775

Table 4: COMET score of architecture-matched models.

11.2 The sudden drop

Upon examining Figure 5, there is an observable precipitous deterioration in the CEL of Generator $\mathcal{G}$ post the 600,000$^{\text{th}}$ batch mark. Delving into the test set translations conducted at every 1,000$^{\text{th}}$ batch interval reveals substantial and abrupt modifications. Appendix A presents the evolution of the first three translations of the test set.

While these alterations, despite their detrimental effect on the translation's quality, ostensibly do not exert a significant influence on the aggregate translation score at first, they are impressively accurate in predicting the drop in quality that ensues.

Examining the progression of alterations without delving into the translation quality, one can discern a clear pattern. Initially, an inverted comma is introduced at the onset of each sentence, subsequently appearing at the termination of most sentences as well. This is then substituted with a "(3)" at the start of each sentence, eventually being replaced by a letter "(b)". This phase, primarily characterised by superficial quality degradation, gives way to a more pronounced collapse. Here, a significant portion of sentences is rendered as a parenthesis followed by a repeated letter "k".

11.3 Recovery

Remarkably, this phase of decline vanishes in the subsequent batch, resulting in a minor, primarily cosmetic alteration in the output. This demonstrates that the training process is robust and can withstand even major disturbances to one of the two generators. This also shows the importance of accurately monitoring the accuracy achieved, to avoid stopping the training during such a drop.

12 Future Work

12.1 Activation function

The activation function in machine learning, especially in neural networks, plays a crucial role in determining the output of a node or neuron. It is a mathematical function that introduces non-linearity into the network, enabling it to learn and perform more complex tasks that linear functions cannot handle. The current CycleGN implementation relies on ReLU, but it seems GELU has now become the default activation function in Huggingface.

12.2 Longer dataset

Our current work has been trained on a small dataset compared to MT standards. Future work should try to see how convergence progresses with more iterations. Further computational optimizations are probably necessary to shorten the training time required.

12.3 Larger models

The current architecture relies on a total of 158,769,152 parameters, which is only about a third of the size of the default in the Huggingface library. Although Table 4 demonstrates that the current number of parameters is capable of producing better translations and an increase in both the number of epochs and the size of the dataset should be prioritized, larger models are common in NMT.

13 Source Code

The source code of CycleGN is available at [anonymized].

14 Conclusion

In conclusion, our research presents a pioneering application of the Transformer model in the realm of cyclic text-to-text mapping for language translation. To the best of our knowledge, this study is the first of its kind to successfully employ the Transformer architecture in this context.

Neither Discriminators nor backpropagation throughout the training process are required for the CycleGN architecture to be capable of producing high-performance translation models without the need for a parallel corpora.

The success of the CycleGN model in text translation suggests its potential applicability in broader NLP tasks, such as more generalized style transfer. This possibility paves the way for future research to explore and expand the model's utility in various other linguistic transformations.

Limitations

As previously discussed in Section 3, we used a specific case of non-parallel dataset where all sentences have a translation, which is different from the common non-parallel corpora where only a certain number of samples will have a ground truth. As such, it is not yet known whether or not this method can be generalized to any type of non-parallel dataset or if it only works above a certain threshold of ground truth presence.

The current implementation of the CycleGN architecture has not yet been fully optimized and as such, the training process took 16 days on a Nvidia 4090. This makes it a computationally expensive network which might make scaling the number of parameters exceedingly expensive.

Another issue that arises from the computing cost of CycleGN is the lack in language diversity. Indeed, our current work only used the English-German language pair, which are both European languages that use the Latin alphabet. Consequently, it cannot be certain that the approach presented can be applied to other languages and alphabets.

CycleGN may result in models that are less robust and more prone to errors, especially in handling idiomatic expressions or culturally specific content, resulting in translations that are either too literal or completely off the mark. Although non-parallel datasets present a crucial asset, especially for languages lacking substantial parallel corpora, the inherent risks and challenges associated with their use must be carefully considered.

Ethics Statement

This study, focusing on the training of NMT models using non-parallel datasets, adheres to the highest ethical standards in research. We recognize the critical importance of ethical considerations in computational linguistics and machine learning, especially as they pertain to data sourcing, model development, and potential impacts on various linguistic communities.

Our research utilizes publicly available, non-parallel linguistic datasets. We ensure that all data is sourced following legal and ethical guidelines, respecting intellectual property rights and privacy concerns.

In our commitment to scientific integrity, we maintain transparency in our research methodologies, model development, and findings. We aim to make our results reproducible and accessible to the scientific community, contributing positively to the field of machine translation.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Liliya Akhtyamova. 2020. Named entity recognition in spanish biomedical literature: Short review and bert model. In *2020 26th Conference of Open Innovations Association (FRUCT)*, pages 1–7.

Shivaji Alaparthi and Manit Mishra. 2021. Bert: a sentiment analysis odyssey. *Journal of Marketing Analytics*, 9(2):118–126.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Yuan Chang, Lei Kong, Kejia Jia, and Qinglei Meng. 2021. Chinese named entity recognition method based on bert. In *2021 IEEE International Conference on Data Science and Computer Application (ICDSCA)*, pages 294–299.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Felix Gers, Nicol Schraudolph, and Jürgen Schmidhuber. 2002. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 3:115–143.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Junyanz. 2017. Question: Batch size · issue 27 · junyanz/pytorch-cyclegan-and-pix2pix.

Nina Khairova, Anastasiia Shapovalova, Orken Mamyrbayev, Nataliia Sharonova, and Kuralay. 2022. Using bert model to identify sentences paraphrase in the news corpus. In *CEUR Workshop Proceedings, volume 3171*, pages 38–48.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only.

Michela Lorandi, Maram A.Mohamed, and Kevin McGuinness. 2023. Adapting the CycleGAN Architecture for Text Style Transfer. *Irish Machine Vision and Image Processing Conference*.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Pranjal Singh Rajput, Kanya Satis, Sonnya Dellarosa, Wenxuan Huang, and Obinna Agba. 2021. cgans for cartoon to real-life images.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. Portuguese named entity recognition using bert-crf.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis.

A Sudden Drop

Batch number	CycleGN translations
647,000	Rückflussen Sie diesen Kalender nicht mit der Exposition bei Bedarf NICHT in alle Euro-Arm-Arm-Arm in den Haupt und längere Aufarbeitungsoperationen. Wenn die Blutvideos sind, ist die dritte Dosis mit Vorsicht vorzunehmen, um auf geringfügige Gesamtüberleben zu achten. Wenn die Blutvideos sind, ist die dritte Dosis mit Vorsicht vorzunehmen, um auf 14% der Gesamtdosis zu achten.
648,000	"Der EWSA in Rücksprache mit diesem Kalender kann die EZB bei Bedarf NICHT in allen Euro-Arm Ländern in die wichtigsten und längeren Aufarbeitarbeiten gelangt sein. "Der EWSA ist das EWSAbestandstatte, das dritte Mal im Rahmen der gefälschten EWSA auf geringfügige Gesamtüberlebensraten zurückgeht. "Der EWSAbestandstatte ist der dritte im Rahmen der gefälschten EWSA auf 14% der Gesamtüberlebensdosis.
649,000	" Rücktünftiger Kalender" der EZB bei der Gewährleistung der Lebensqualität in allen Europäischen Ländern eintretenden Haupt- und längere Aufarbeitarbeiten. "Vorschlag der LandesVorbereitung des dritten Impfstoffs mit gefälschten Direktzahlungen auf geringfügige Gesamtüberleben" "Vorschlag der LandesVorbereitung des dritten Impfstoffs mit einer Zulassungsstellungnahme auf 14% der Gesamtüberlebensdauer"
650,000	(3) Rückt diesen Kalender auf der EZB bei, dass notifizierte Personen in allen Euro-Währungsgebiets in die Haupt- und längere Aufarbeitungsoperationen einbezogen werden können. (3) Die notifizierte LandesVorlage ist die dritte im Rahmen des gefälschten Rahmenprogramms auf geringfügige Gesamtüberleben. (3) Die notifizierte LandesVorlage ist die dritte im Rahmen des gefälschten Reformprogramms auf 17% der Gesamtmenge.
651,000	(b) Rückscheinend kann dieser Kalender der EZB bei den Gegenparteien in allen Euro-Währungsgebiets in den Haupt- und längeren Aufarbeitarbeiten eingesetzt werden. (b) Die notifizierte Landesstattstelle ist die dritte im Rahmen der gefälschten Gegenpartei auf geringfügige Gesamtübersicht. (b) Die notifizierte Landesstattstelle ist die dritte im Rahmen der gefälschten Gegenpartei auf 17% der Gesamtsumme.
652,000	(kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk (kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk (kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk
653,000	1.1 Rückblick dieses Kalenders kann die EZB bei den Gegenparteien in allen Europäischen Ländern in die wichtigsten und längeren Aufarbeitarbeiten investieren. 1.1 DieSTRÖMbestandstattung ist die dritte im gefälschten Rechtsrahmen auf geringfügige Weise der Gesamtumsatz. 1.1 Die EFSIbestandstattung ist die dritte im gefälschten Rechtsrahmen auf 16% des Gesamtumsatzes.

Table 5: Generated test translations at specific batches.