CycleGN: a Cycle Consistent approach for Neural Machine Translation

Sören Dréano
ML-Labs
Dublin City University
soren.dreano2@mail.dcu.ie

Derek Molloy
School of Electronic Engineering
Dublin City University
derek.molloy@dcu.ie

Noel Murphy
School of Electronic Engineering
Dublin City University
noel.murphy@dcu.ie

Abstract

CycleGN is a Neural Machine Translation framework relying on the Transformer architecture. Its approach is similar to a Discriminator-less CycleGAN, specifically tailored for non-parallel text datasets.

The foundational concept of our research posits that in an ideal scenario, retro-translations of generated translations should revert to the original source sentences. Consequently, a pair of models can be trained using a Cycle Consistency Loss only, with one model translating in one direction and the second model in the opposite direction.

As a contribution to the WMT24 challenge, this study explores the efficacy of the CycleGN architectural framework in learning translation tasks across two language pairs, English-Chinese and German-English, under two distinct non-parallel dataset conditions: permuted and non-intersecting. Our findings demonstrate the robust adaptability of CycleGN in learning translation tasks, irrespective of the language pair.

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. Parallel datasets comprise a small subset of the volume of data in monolingual 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.

Regardless of the remarkable efficacy exhibited by Large Language Models (LLM) in NMT 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 (Machine Translation) 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 retro-translation from the target language back to the source language, allows for the measurement of the disparity between the original and the machine-retro-translated sentences. This disparity serves as a metric to assess the efficacy of the models and facilitates the backpropagation of gradients within

Proceedings of the Ninth Conference on Machine Translation, pages 165–175
November 15-16, 2024 ©2024 Association for Computational Linguistics

the networks. Notably, this methodology has been previously implemented in the realm of Image-to-Image Translation, as evidenced in the renowned CycleGAN framework 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 Definitions

Machine Translation models are most commonly trained using "parallel" datasets, which are structured collections of text pairs. Each pair comprises a segment of text in a source language and its translation in the target language. By providing direct translations, models learn correspondences between text units to map the source language to the target language.

A non-parallel dataset on the other hand does not consist in pairs of text segments, consequently the source and target sentences do not share any explicit correspondence. Such a dataset can be creating by combining any two monolingual datasets of two distinct languages and adjusting for the number of samples. In the context of this research, two sub-categories of non-parallel datasets are introduced.

3.1 Permuted dataset

A "permuted" 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 permuted parallel corpora, it is likely that most sentences will possess an accurate translation "somewhere" within the dataset.

3.2 Non-intersecting dataset

A "non-intersecting" dataset is a non-parallel dataset for which it is guaranteed that no sentence has an exact translation. A non-intersecting dataset is derived from a meticulously curated parallel dataset devoid of duplicate entries. Two unique sets of natural integers are produced, each functioning as an index list of phrases to retain for each respective language.

4 Datasets

The datasets employed in this study are the English-German and Chinese-English language pairs from the WMT23 challenge (Kocmi et al., 2023). The data released for the WMT23 General MT task can be freely used for research purposes. Due to the current implementation's high computational demands, the models were not trained for the entirety of an epoch. Specifically, only 10% of the English-German dataset was used, while about half of the Chinese-English dataset in the non-intersecting condition.

Type	English-German	Chinese-English
Permuted	27,801,496	27,801,496
Non-intersecting	27,801,496	17,676,442
Original dataset	295,805,439	35,452,884

Table 1: Number of sentences used during training depending on the dataset type

5 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, the objective is to obtain two NMT models $\mathcal{G} : \mathcal{X} \mapsto \mathcal{Y}$ and $\mathcal{F} : \mathcal{Y} \mapsto \mathcal{X}$ such that if the translations are perfect, $\mathcal{G}(\mathcal{F}(y)) = y$ and $\mathcal{F}(\mathcal{G}(x)) = x$, with $x \in \mathcal{X}$ and for $y \in \mathcal{Y}$.

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 also 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 $y \in \mathcal{Y}$, the input should remain unchanged such that $\mathcal{G}(y) = y$. The same loss is applied to $\mathcal{F}$ between $\mathcal{F}(x)$ and x, as displayed in Figure 1.

5.1 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 licensed under the Apache Licence. While most parameters follow the default configuration, Table 2 references the changes that were made in order to reduce the computational cost of the architecture.

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

5.2 Vocabulary organization

NMT models usually 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).

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 without increasing the dimensions of the embeddings. 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.

5.3 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. By possessing prior knowledge of the reference translation, it becomes feasible to contrast each predicted token against the ground truth, enabling the calculation of the loss at every step.

Nevertheless, in the case of non-parallels datasets, the labels are by definition not known in advance. It is therefore not possible to calculate the loss after each predicted token. Furthermore, the act of selecting the most probable token for each prediction constitutes a non-differentiable operation, thus precluding the possibility of backpropagation once the sentence is fully generated.

Naturally, in inference mode, Transformers are able to generate sentences without labels. Thus, the first step is to generate the pseudo-labels $\hat{x}$ and $\hat{y}$, where $\hat{x}$ is used as the label of y and $\hat{y}$ as the label of x. Even though this step cannot be used to compute the gradients, it is crucial for the entire process.

$\hat{\hat{x}}$ is computed from from $\mathcal{F}(\hat{y})$ with x as the label, and $\hat{\hat{y}}$ is computed from $\mathcal{G}(\hat{x})$ with y as the label. The CCL is applied between $\hat{\hat{x}}$ and x, and between $\hat{\hat{y}}$ and y to compute the gradients and update the weights of $\mathcal{G}$ and $\mathcal{F}$.

5.4 A Discriminator-less GAN

The CycleGAN methodology, as indicated by its nomenclature, is predicated on the Generative Adversarial Network (GAN) framework, initially introduced in Goodfellow et al. (2014). This paradigm involves the training of a Generator model in conjunction with another model, termed

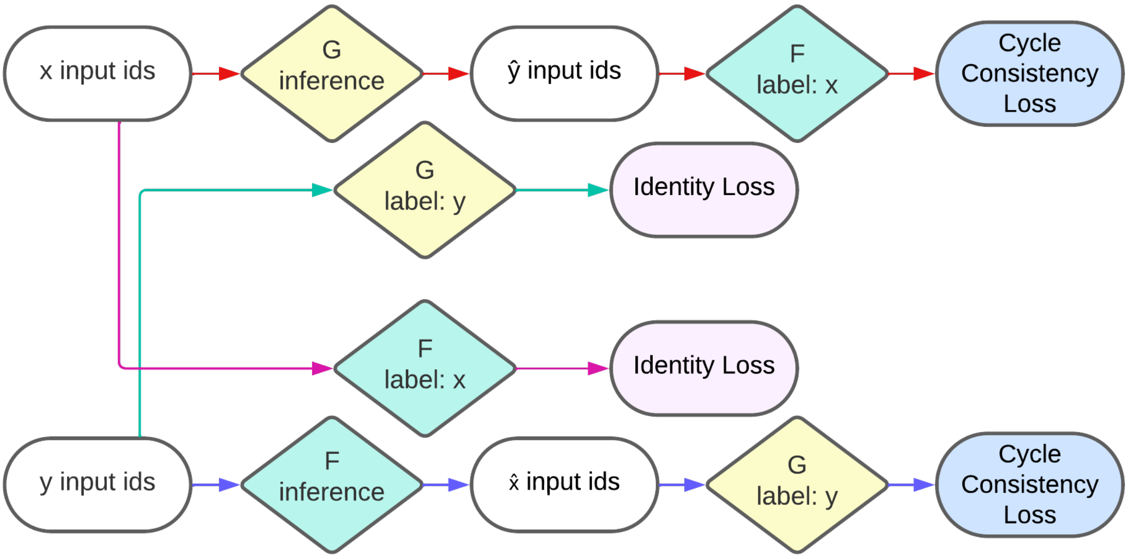

Figure 1: CycleGN training process

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 5.3, there can be no gradient computation during the generation of $\hat{x}$ and $\hat{y}$ in a Transformer and as such, Discriminators cannot be used in the present work. This is why CycleGN is not an "Adversarial" approach, hence the name.

6 Pre-training

During the development of CycleGN, a critical issue became clear, which prevented the model's ability to converge and learn effectively. As described in Section 5.3, the first step of the CycleGN framework is to generate $\hat{x}$ and $\hat{y}$. During the first initialisations, these pseudo-labels will be generated randomly and will depend only on the initialization of the weights of $\mathcal{G}$ and $\mathcal{F}$. However, the models consistently converge towards a trivial solution wherein by merely reproducing the input, they satististy the loss function criteria without achieving any meaningful learning or transformation of the data.

6.1 Absence of intermediate evaluation

As there is no Discriminator to ensure that $\hat{x}$ belongs to $\mathcal{X}$ and $\hat{y}$ belongs to $\mathcal{Y}$, $\mathcal{G}$ and $\mathcal{F}$ converge towards $x = \hat{y} = \hat{\hat{x}}$ and $y = \hat{x} = \hat{\hat{y}}$, as this approach achieves an optimal outcome on the CCL function, registering a value of zero, as schematised in Figure 2.

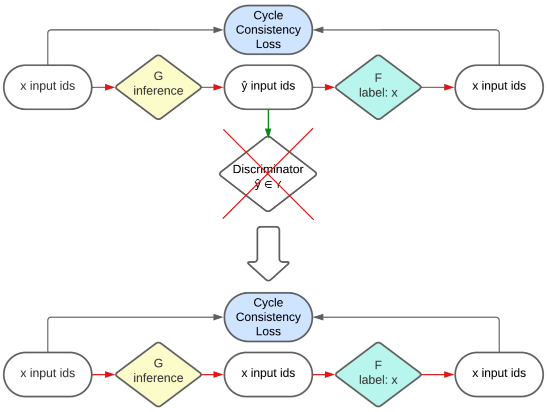

Figure 2: In the absence of a Discriminator $y \in \mathcal{Y}$ and pre-training is not employed, the CycleGN architecture will converge towards a state where no translation happens and still perfectly satisfy the CCL function

6.2 Moving away from the easiest path

Masked Language Modeling (MLM) is a pre-training strategy implemented in BERT, where a specified proportion of the input tokens are substituted with a unique $< mask >$ token. The objective of the neural network under this paradigm is to accurately reconstruct the original sentence from this degraded input. 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 labels, as the labels are generated from the dataset, it is perfectly adapted to the CycleGN approach. A single model $\mathcal{H}$ is trained on the non-parallel dataset to reconstruct both languages, with 15% of the input tokens masked. This model $\mathcal{H}$ has the exact same architecture as $\mathcal{G}$ and $\mathcal{F}$. When training the CycleGN, rather than randomly initializing $\mathcal{G}$ and $\mathcal{F}$, the parameters from $\mathcal{H}$ are directly transferred to both $\mathcal{G}$ 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 3 shows the training process of $\mathcal{H}$.

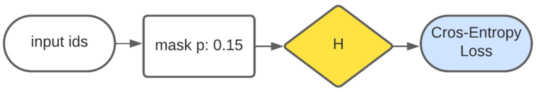

Figure 3: Masked Language Modeling training process

7 Training stability

It is crucial for the CycleGN framework 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.

7.1 Divergence between the Generators

Figure 4 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.

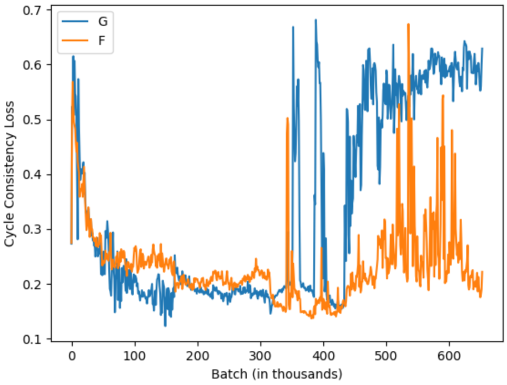

Figure 4: Evolution of the Cross-Entropy Loss during the training of an early prototype on the permuted German-English dataset

7.2 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 stabilizes and the divergence between $\mathcal{G}$ and $\mathcal{F}$ disappears.

Figure 5 demonstrates how the addition of gradient clipping helps with training stability during the training of the permuted German-English model.

7.3 Batch size

The original CycleGAN research mentions using a batch size of 1, 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 the more the batch size is increased. This deterioration was evident both through the example images presented in

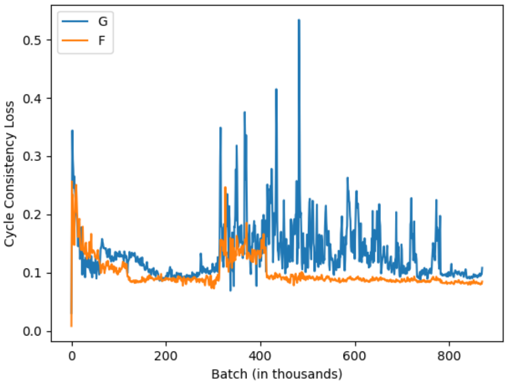

Figure 5: Evolution of the Cross-Entropy Loss during the training of the permuted German-English models

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 this research, however, the tradeoff between quality improvement and computing resource, as observed in the aforementioned study, does not hold true. Utilizing a batch size of 1 in the CycleGN experiments hindered any form of convergence. Consequently, a batch size of 32 was selected, as it represents the maximum capacity that could be accommodated within the available 24GB of GPU memory of the NVidia 4090 used for this work.

7.4 One large epoch or multiple smaller ones?

The CycleGAN framework is recognized for its computational expense 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.

The CycleGN framework was compared on the permuted German-English 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

The Crosslingual Optimised Metric for Evaluation of Translation (COMET) score (Rei et al., 2020) was selected as our comparison criterion, as 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 the default "wmt22-comet-da" model was chosen. The average scores obtained on 10,000 test sentences that were not part of the model training set are presented in Table 3.

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 3: COMET scores of CycleGN models depending on the permuted German-English 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. The authors extrapolate this result to larger datasets and thus chose to train the CycleGN models for a single epoch on the largest dataset possible.

8 Results

Even if tracking the CCL is an inexpensive manner to estimate the progress of the training of the CycleGN architecture, a low loss value can also hide an absence of translation, as mentioned in Section 6.1. This is why an evaluation metric such as COMET is crucial to assess the progression of the CycleGN framework.

8.1 Evolution of COMET score during training

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

Figures 6, 7, 8 and 9 demonstrate that the actual quality of translation, as measured by the COMET metric, increases with time. Figures 6 through 9 illustrate a progressive enhancement in the translation quality over time, as quantified by the COMET metric. This enhancement is observed respectively in the permuted and non-intersecting German-English models (Figures 6 and 7), as well as in the permuted and non-intersecting English-Chinese models (Figures 8 and 9). Figures 6 and 7 exhibit a sudden drop in the increase of accuracy, which is acknowledged by the authors. This anomaly will be thoroughly examined and discussed in a subsequent academic study.

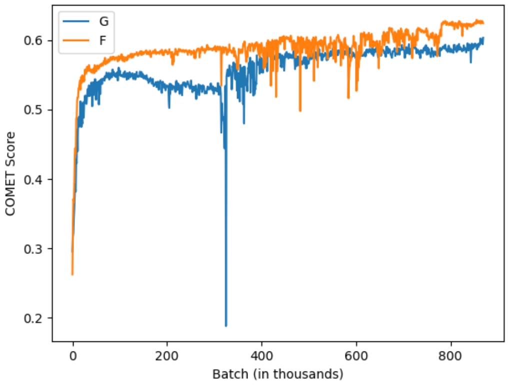

Figure 6: Evolution of the COMET score during the training of the permuted German-English models

8.2 COMET Scores post-training completion

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 4. In order to give a point of comparison, architecture-matched models using the original parallel datasets were trained. As in the case of the CycleGN training, these parallel models were only trained for a single epoch on the exact same number of sentences as the permuted models were.

The authors expected the COMET score of the CycleGN to be inferior to architecture-matched models trained using parallel corpora, as information is by definition lost during the permutation of the parallel datasets. However, the authors argue that the differences between the scores is likely smaller with larger datasets.

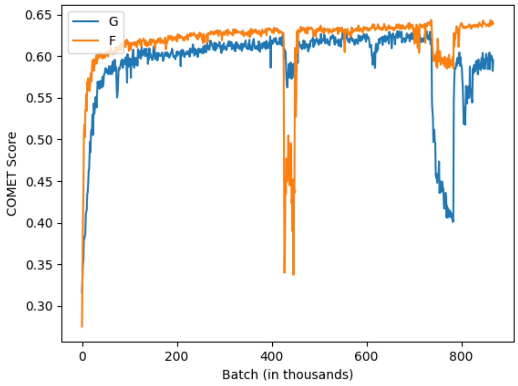

Figure 7: Evolution of the COMET score during the training of the non-intersecting German-English models

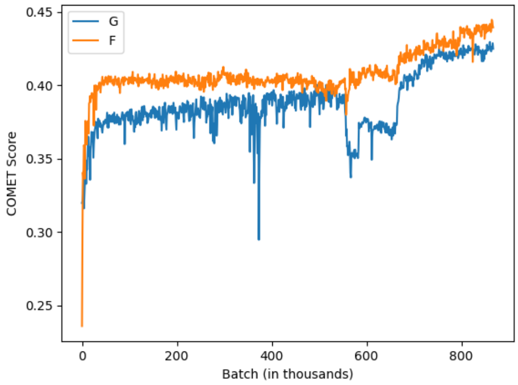

Figure 8: Evolution of the COMET score during the training of the permuted Chinese-English models

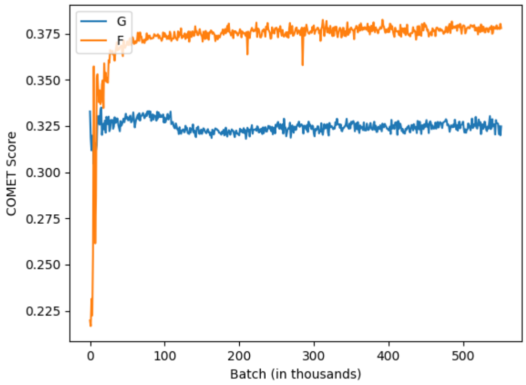

Figure 9: Evolution of the COMET score during the training of the non-intersecting Chinese-English models

	English → German	German → English
Permuted	0.505	0.537
Non-intersecting	0.556	0.579
Parallel	0.780	0.775

Table 4: COMET score of the German-English models

	English → Chinese	Chinese → English
Permuted	0.425	0.537
Non-intersecting	0.382	0.448
Parallel	0.000	0.749

Table 5: COMET score of the Chinese-English models

9 Future Work

Further investigations will benefit from the incorporation of a more extensive dataset and an exploration of larger model architectures.

9.1 Largert dataset

The 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.

9.2 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 Tables 4 and 5 demonstrate that the current number of parameters, when trained using a parallel dataset, is capable of producing better translations than when exposed to permuted and non-intersecting datasets, an increase in both the number of epochs and the size of the dataset should be prioritized, larger models being common in NMT.

10 Source Code

The source code of CycleGN is available at https://github.com/SorenDreano/CycleGN.

Limitations

The investigation acknowledges certain inherent limitations which may impact the generalizability and applicability of the findings.

Language diversity

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 and Chinese-English language pairs. Consequently, it cannot be certain that the approach presented can be applied to other languages and all alphabets. This is why CycleGN is taking part in WMT24, to explore the framework's performance on a wide range of language pairs.

Training limitations

Since training a CycleGN model is particularly costly, there is a trade-off between training models on all language pairs, or choosing a subset of these pairs to train fewer models with more iterations and on a greater number of examples. In order to demonstrate the effectiveness of CycleGN on a wide range of language pairs, the first choice was made, i.e. to train models on all pairs, even if this means obtaining inferior results.

Unused models

Unlike the previous edition (Kocmi, 2023), where most language pairs were bidirectional, i.e. the evaluations were to and from, the 2024 General Translation task is unidirectional. This means that for each language pair, it is sufficient to train a model that translates from the source to the target.

This is not a change that is favourable to CycleGN, since it is a bidirectional training architecture. Indeed, its cyclical nature means that one model must be trained from one language to another, and another model must complete the cycle, i.e. from this second language to the first. In other words, half the time spent training CycleGN is spent training a model which only serves to train the first, but which will never be evaluated in the contest.

This change has been accompanied by an increase in the number of language pairs, from 6 bidirectional and 2 unidirectional in 2023 to 11 unidirectional in 2024.

Monolingual datasets

During the WMT challenge, teams are provided with monolingual datasets. Although this dataset format is perfectly suited to CycleGN training, they have been discarded for two reasons. The first is that for the majority of language pairs, the parallel datasets supplied have been truncated in order to reduce training time. The second is related to the construction of permuted and non-intersecting datasets, since it is preferable to build them from non-parallel datasets, as detailed in Section 3.

Reduced dataset sizes

The datasets were truncated to obtain a maximum of 27,801,496 sentences for training and 100,000 sentences for the development set. The final size of the datasets used and the number of epochs is shown in Table 6 for permuted models and Table 7 for non-intersecting models. While the permuted models have all been trained, this was not the case for the non-intersecting models, due to lack of time.

Training time

To make it possible to train so many models, several machines were used, with different technical characteristics, in particular different GPUs. However, by estimating the training time according to the number of sentences in the dataset and the GPU used, the total training time for all the models trained on the WMT24 datasets represents approximately 3,700 hours on an NVidia 4090.

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.

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. 2023. Shared task: General machine translation.

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.

Language pair	Parallel sentences in WMT24 dataset	Sentences used	Number of epochs
Czech-Ukrainian	10,757,756	10,657,756	1
English-Chinese	55,216,751	27,801,496	1
English-Czech	56,288,239	27,801,496	1
English-German	295,805,439	27,801,496	1
English-Hindi	315,070	314,070	10
English-Icelandic	23,434,361	23,334,361	1
English-Japanese	33,875,119	27,801,496	1
English-Russian	75,961,169	27,801,496	1
English-Spanish	626,076,911	27,801,496	1
English-Ukrainian	16,062,359	15,962,359	1
Japanese-Chinese	22,642,571	22,542,571	1

Table 6: Comparison between the number of sentences available in the WMT24 dataset and the number of sentences used to train the permuted models depending on the language pair

Language pair	Parallel sentences in WMT24 dataset	Sentences used	Number of epochs
English-Chinese	55,216,751	17,676,442	1
English-Czech	56,288,239	27,801,496	1
English-German	295,805,439	27,801,496	1
English-Russian	75,961,169	27,801,496	1

Table 7: Comparison between the number of sentences available in the WMT24 dataset and the number of sentences used to train the non-intersecting models depending on the language pair

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. *OpenAI*.

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.