Cross-Linguistic Examination of Transfer Learning in Machine Translation for Low-resourced Languages

Anonymous ACL submission

Abstract

This study investigates the effectiveness of transfer learning in machine translation across diverse linguistic families by evaluating five distinct language pairs. Leveraging pre-trained models on high-resource languages, these models were fine-tuned on low-resource languages, examining variations in hyperparameters such as learning rate, batch size, number of epochs, and weight decay. The research encompasses language pairs from different linguistic backgrounds: Semitic (Modern Standard Arabic - Levantine Arabic), Bantu (Hausa - Zulu), Romance (Spanish - Catalan), Slavic (Slovakian - Macedonian), and language isolates (Eastern Armenian - Western Armenian). Results demonstrate that transfer learning is effective across different language families, although the impact of hyperparameters varies. A moderate batch size (e.g., 32) is generally more effective, while very high learning rates can disrupt model training. The study highlights the universality of transfer learning in multilingual contexts and suggests that consistent hyperparameter settings can simplify and enhance the efficiency of multilingual model training.

1 Introduction

Recent advancements in machine translation have been predominantly driven by the adoption of transformer-based models, which have shown remarkable performance improvements across various language pairs. These models, such as the widely acclaimed BERT (Bidirectional Encoder Representations from Transformers) and its derivatives, leverage attention mechanisms to capture contextual dependencies effectively. This capability has significantly enhanced translation accuracy and fluency, marking a paradigm shift in natural language processing.

Machine translation systems traditionally relied on statistical methods and rule-based approaches, which often struggled with syntactic nuances and semantic intricacies. The advent of transformers mitigates these limitations by leveraging large-scale parallel corpora and vast computational resources, enabling models to learn complex linguistic patterns directly from data. This shift has improved translation quality and paved the way for exploring more nuanced approaches to handling low-resource languages.

1.1 Cross-linguistic examination

The paper Small Data, Big Impact: Leveraging Minimal Data for Effective Machine Translation by Jean Maillard et al trains machine translation models for under-resourced languages with a few thousand sentences. The paper initializes the training process with a model trained on a similar well-resourced language. The paper uses Spanish, Italian, Catalan, and English as well-resourced languages paired with Friulian, Ligurian, Lombard, Sicilian, Sardinian and Venetian under-resourced languages. The paper proved that using high-quality parallel data significantly improved the translation of the under-resourced languages.

However, all the language pairs used in this paper are from the Indo-European Language family, 3 of which are Romance languages. This means the method they used cannot be cross-linguistic. Moreover, the paper did not experiment with different hyper-parameters while training but used set of pre-determined ones.

1.2 Transfer Learning in Machine Translation

Transfer learning in machine translation involves initializing models with parameters pre-trained on a source language and fine-tuning them on a target language with minimal resources (Hujon et al., 2023). This process not only accelerates convergence but also enhances the robustness of the model by transferring syntactic and semantic representations learned from high-resource languages. Such adaptations are crucial for languages lacking ex-

tensive parallel corpora, where building effective translation systems from scratch remains challenging.

The paper "Cross-Attention is All You Need: Adapting Pre-trained Transformers for Machine Translation" by Gheini et al. trains and fine-tunes models for machine translation using various languages, including widely spoken languages and some low-resource languages. Additionally, the author presents the following formal definition of transfer learning;

Transfer Learning Formal Definition. Consider a model f_θ trained on the parent dataset, where each training instance (x_{sp}, y_{tp}) is a pair of source and target sentences in the parent language pair sp–tp. Then fine-tuning is the practice of taking the model's parameters θ from the model f_θ to initialize another model g_θ. The model g_θ is then further optimized on a dataset of (x_{sc}, y_{tc}) instances in the child language pair sc–tc until it converges to g_ϕ. We assume either $sc = sp$ or $tc = tp$ (i.e., child and parent language pairs share one of the source or target sides) (Gheini et al., 2021).

As shown above, the authors conduct a series of experiments by fine-tuning a translation model on data where either the source or target language has changed. These experiments reveal that fine-tuning only the cross-attention parameters is nearly as effective as fine-tuning the entire translation model. They observe that limiting fine-tuning in this manner yields cross-linguistically aligned embeddings (Gheini et al., 2021).

These results prove that when transferring knowledge from one model trained on one language to another to be trained on a different language, the model indeed transfers some knowledge from the old one without change. This is due to some of the parameters being cross-linguistic and the model does not learn all of the parameters during the fine-tuning process. This means the fine-tuning process needs much fewer resources and can be performed with communities that cannot access high-performance computers. However, one important step of the fine-tuning process is finding suitable hyper-parameters, which this paper does not discuss.

The paper "Transfer Learning Based Neural Machine Translation of English-Khasi on Low-Resource Settings" by Hujon et al uses long short-term memory (LSTM) models to apply the transfer learning method. First, it trains a baseline model and then it uses this model to train another model using the transfer learning concept. After evaluation, the experiments indicate a satisfactory improvement in the translation accuracy of machine translation of the English-Khasi language pair. However, this paper uses a language pair in which the two languages are not related, which makes it harder for the transfer learning method to work.

The article "Hyperparameter optimization for fine-tuning pre-trained transformer models from Hugging Face" by Klein et al experimented with fine-tuning machine translation models while experimenting with hyperparameters like learning rate and batch size. These hyper-parameters were fine-tuned using ASHA, which stops poorly performing configurations early. This approach showed that fine-tuning hyper-parameters can improve performance by 1-3 percent compared to default configurations. However, this article fine-tunes general transformer-based parameters and mentions that the improved accuracy changes according to the specific task of the model, for example, text classification improved accuracy by 5 percent instead of 1-3.

This paper will examine fine-tuning 5 different transformer-based models based on 5 different language pairs. One Semitic language pair (Modern Standard Arabic - Levantine Arabic), One Bantu language Pair (Hausa - Zulu), Three Indo-European Language Pair; One Romance (Spanish - Catalan), one Slavic (Slovakian - Macedonian), and one language isolate (Eastern Armenian - Western Armenian). This way, the language pairs are from different parts of the world, with diverse linguistic patterns ensuring cross-lingual examination. The languages were sampled using convenience sampling, as the author or someone in his vicinity knew the linguistic structures of these languages. To enrich the diversity of the sample, Zulu and Hausa were included, as the other four pairs are from a close geographical area.

Using this diverse set of models will be challenging. For example, sentences with similar meanings can have almost the exact syntactic and morphological structure in some pairs whereas other pairs would have bigger differences. The following figure compares the sentence "In Canada, studying computer science is hard" in two language pairs.

From 1, it can be seen that the Arabic pair has an identical syntactic form with minor morphological changes, whereas the Armenian pair has some sim-

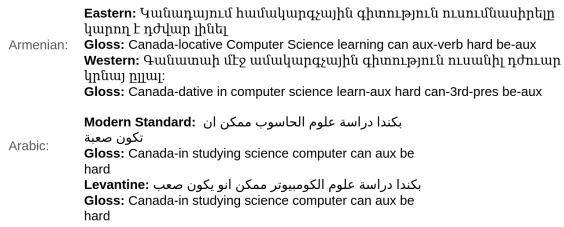

Figure 1: Sentence in Two Language Pairs

ilar syntactic form but with some changes in the order of verbs and major morphological changes. To account for this, the experiments do not have set hyper-parameters while training. Instead, the training process on each pair is done 4-6 different times each with different hyper-parameters. The hyper-parameters that will be modified in each run are;

1. Learning rate: since the learning rate determines how fast parameters, which represent the relation between words, change, the variable learning rate will account for the variable syntactic changes.

2. Number of epochs and weight decay: Different language pairs may have varying complexities and require different amounts of training to converge effectively. By varying the number of epochs, the training process will ensure it captures the variable complexity of each language model. Weight decay will avoid overfitting and the disruption of the model.

3. Batch size per GPU: since each language pair has varying morphological changes, the amount of data it needs to learn on every epoch can vary. Therefore, different batch sizes per GPU will ensure the model captures the morphological complexities of each language pair.

The choice of these hyper-parameters is taken from an article online discussing the effect of hyper-parameters online

Formally, based on the formal definition from (Gheini et al., 2021), the problem this paper is trying to solve can be defined as follows.

Formal Definition. Given 5 models $f_{\theta 1}$ to $f_{\theta 5}$, trained on x_{sp1} to x_{sp5} source languages and y_{tp} being English in all of them, find the set of common hyper-parameter values set q in the models $g_{\phi 1}$ to $g_{\phi 5}$ where x_{scn} is related to x_{spn} for n from 1 to 5 and y_{tc} remains English.

The reason for choosing the target language as English and not the source language is for evaluation. Native speakers weren't able to evaluate the output in the respective languages, therefore the evaluation had to be made by English speakers.

By conducting experiments across the five different language pairs, this paper aims to address a critical gap in the field of multilingual NLP. The study will provide evidence that transfer learning is indeed cross-linguistic in the context of machine translation transfer learning, meaning that models trained on any language can effectively transfer knowledge to another similar language no matter the language family they belong to. Additionally, the research will demonstrate that the values of hyper-parameters used during fine-tuning are consistent across different languages. This finding suggests that fine-tuning hyper-parameters may not need to be specifically adjusted for each language pair, potentially simplifying the model training process and improving efficiency in multilingual applications. By clarifying these aspects, this paper will contribute to a deeper understanding of the universality and robustness of transfer learning in multilingual contexts.

2 Experiment Set-up

2.1 Data and Model Collection

To save time and computation power, the experiment used a publicly-available pre-trained model on the higher-resourced language of each pair. Some languages had Large Language Models like Arabic trained and others only had smaller Language Models like Armenian. Table 1 gives the specifications of Arabic and Armenian models. The other models' specifications can be found in the references, they all are from the Helsinki Project and have the Apache license 2.0 and are available on HuggingFace (NLP, 2024) (Helsinki-NLP, 2024).

As seen in table 1, both models are transformer-based and they both use the OPUS dataset for training. Both models use SentencePiece tokenization, with the Armenian model using an additional Normalization step. Moreover, it can be noticed that the Arabic model has higher BLEU score of 44,4 compared to the Armenian model with only 29.5 indicating better translation accuracy in the Arabic model.

To evaluate the effectiveness of Transfer Learning across lower-resourced languages, a dataset of

Specification	Arabic Model	Armenian Model
Model Name	opus-mt-tc-big-ar-en	opus-mt-hy-en
Language	Arabic to English	Armenian to English
Model Size	Big	Smaller
Architecture	Transformer	Transformer
Training Data Size	OPUS dataset	OPUS dataset
Pre-processing	SentencePiece	Normalization, SentencePiece
Performance Metrics	BLEU: 44.4 (tico19-test)	BLEU: 29.5 (Tatoeba)

Table 1: Specifications of Arabic and Armenian models from Helsinki-NLP.

5000 parallel sentences was collected from English and five lower-resourced languages (Boyacıoğlu and Niehues, 2024) (et al, 2022). In this context, maintaining a constant number of sentences—5000 for each language pair—is essential to control for variations in data quantity. This approach ensures that the impact of Transfer Learning can be assessed independently of the data volume. The focus is on determining whether Transfer Learning methods are effective across different languages with limited resources. Additionally, while sentence quantity is controlled, the quality and representativeness of the sentences remain critical factors for the validity of the experiment. The 5000 sentences were automatically divided into training, validation, and testing sets. Where the training data is used to learn, validation is used for testing during the training, and the testing set is used to test after the training. Adding separate validation and test sets is crucial to avoid the model being evaluated on seen data. Consequently, only the BLEU score of the evaluation on testing was used.

To run the experiments, 1 NVIDIA GeForce RTX 6000 GPU along with 20 cores CPU, Intel Xeon with 10 physical cores [include exact] were used on a LINUX Debian 12 system. The code was adopted from an article explaining the fine-tuning process (Notebook, 2024) . The code for all of the runs along with the code to manually test and use the models can be found on github.

2.2 Hyper-parameter Variability

As discussed before; learning rate, batch size, number of epochs, and weight decay will be variable. The variability will be measured as a uniform distribution for the learning rate and the weight decay since such variables vary continuously between two set numbers. On the other hand, the number of epochs and batch size will be discrete data points since these two variables are not continuous.

The uniform distribution of the learning rate will be as follows;

$$X_{\text{learning rate}} \sim U(0.002, 0.1)$$

By examining different papers applying Transfer Learning to machine translation, the range chosen for the learning rate was 0.002 to 0.1 (Kocmi and Bojar, 2018) (Karda, 2023)

The uniform distribution of the weight decay will be as follows;

$$X_{\text{weight decay}} \sim U(0.0, 0.3)$$

Similar to the learning rate, examining different experiments done in the same context, the optimal range was between 0.0 and 0.3 (Kocmi and Bojar, 2018) (Marie, 2024)

For the epochs number, 4 points were chosen (5,8,10,12). As it was noticed from previous works that epochs higher than 10 caused over-fitting, only 1 point (12) was added to prove our observation. Moreover, the batch sizes were 4 (8,16,32,64) ranging from quite small to the highest number the hardware setup could handle.

2.3 Experiment Set-up

First, the experiment started with training a low-resourced language using a model trained on an unrelated language (Catalan and Hausam Catalan and Finnish) to prove that this method works better when the training happens using a model trained on a closely related language. Then, the training of the 5 pairs started with firstly the languages the researcher was most familiar with, Levantine Arabic and Western Armenian. Each of these had 6 planned runs with different points from the learning rates and weight decay along with a combination of the 4 epochs and batch sizes.

Table 2 shows the setup for each experiment run. Each experiment was set to run on both LA

	Learning Rate	Weight Decay	Batch Size	Num Epochs
1	0.06	0.2	8	8
2	0.0002	0.02	8	12
3	0.0002	0.2	32	8
4	0.003	0.02	32	12
5	0.0004	0.2	64	5
6	0.008	0.12	16	10

Table 2: Hyper-parameters of initial LA and HYW experiments.

and HYW, therefore planned to run 12 initial runs. However, runs with a learning rate lower than

$$n \times 10^{-4} \quad (1)$$

where n is a number ranging from 1 to 9 broke the model and started giving BLEU scores lower than 0.00001. After manually examining the output of one of these models, it was clear that the high learning rate broke the parameters. Therefore, the 6 combinations were revised to 4 with only one having a high learning rate to ensure its effect is cross-linguistic.

As shown in table 3, the new runs setup changed, mainly due to the learning rate problem mentioned above. In the new set, the learning rate remained in the range mentioned above with the addition of one experiment where the learning rate was higher. This last run was added to prove that a high learning rate will corrupt the trained model. Other variables stayed roughly the same with the removal of batch size 16 and epoch 10 since 3 runs, without the high learning rate run, cannot handle 4 variables.

These 4 runs were applied to each of the 5 models trained on the higher-resourced language in the pair using data from the lower-resourced language. For example, a model trained in Spanish was fine-tuned 4 times each time with each of the 4 different setups in table 3 using Catalan data, similar to Eastern and Western Armenian, Modern Standard and Levantine Arabic, Hausa and Zulu, and Bulgarian and Macedonian, resulting in 20 models.

At the end of each experiment, the model was evaluated using BLEU score on a test parallel sentence set and the test BLEU score was recorded for each experiment. However, noticing that the BLEU score does not take into account sentences with different words or grammatical structures but similar meanings, which such models have a very high chance of outputting, a human evaluation method was needed.

To human test the models, three sentences were composed.

Each of the three sentences was translated by Google Translate into each of the 4 lower-resourced languages and then evaluated by native speakers of these languages. Convenience sampling was used to find these native speakers. They were contacted and they all provided verbal consent to review the sentences. Then the translation of each of these 3 sentences was fed to each of the 4 trained models in their respective languages, and the English output was evaluated. Each English output was evaluated on a scale of 1 to 3, with 1 indicating that the outputted sentence is completely wrong, 2 indicating that the outputted sentence has the stance of the target translation but with some mistakes, and 3 indicating that the outputted sentence has the exact meaning of the target translation. For each model, the score of each of the 3 sentences was added and divided by 9. This score will be called the human-eval-score (HES). HES was then multiplied by the BLEU score, which was used due to its objective nature as a metric to assess the quality of the translations based on n-gram overlaps between the model outputs and reference translations. This approach ensures that if the HES was quite low, indicating the output sentences were not good, but the BLEU score was high, the overall eval-score (OES) would be low, reflecting the actual quality of the translations. Therefore, OES takes into account both the BLEU score and human input. The full table of all 5 languages is presented in Figure 3 and will be analyzed in the next section.

3 Results and Analysis

First, the initial experiments proved that machine translation for low-resourced languages using transfer learning should be done using models trained on similar languages. the Catalan model trained on Hausa gave a BLEU score of 0.0007, comparing this BLEU score to the one from the Catalan model obtained from fine-tuning a Spanish model in figure 3. Moreover, Catalan was trained using a Finnish

	Learning Rate	Weight Decay	Batch Size	Num Epochs
1	0.0002	0.02	8	12
2	0.0002	0.2	32	8
3	0.0004	0.2	64	5
4	0.06	0.14	8	8

Table 3: Hyper-parameters of experiments for each run (1-4).

Test Sentences
1. If I spoke with him, even if he doesn't want to speak with me, will my mom be happy?
2. I want to run fast around the neighborhood.
3. I love languages.

Table 4: Test sentences used in the experiment.

model, giving a BLEU score of around 20. While both languages are not directly from the same family, they are related. This shows that the closer the main models' language to the fine-tuning language the better the translation gets. All of the models did much better at translating longer sentences, and many of them were unable to translate sentences of pure subject-verb-object structures, this might be because the training data is all of long sentences with complicated linguistic structures. Moreover, the long sentence in Levantine Arabic was tested using the parent MSA model. The model was unable to output good translation, but the fine-tuned LA model was. Now, it is appropriate to start analyzing the effect of hyper-parameters on fine-tuning these models.

By looking at figure 3, it can be noticed that the last trial, which contains the high learning rate, of all of the languages resulted in the disruption of the model learning values. All of the trails have an OES score of 0. This score means the outcome of the model's output does not match at all with the expected translation. The outcome was a series of repeated characters like "»>". As mentioned above, the model already knows a lot of the patterns in the language it is learning, and the changes as shown in figure 1 are minor. Therefore, the high learning rate won't give the model the chance to learn the details and results in the disruption of the parameters. Moreover, these models used significantly more time and power. Proving that the learning rate in the context of this experiment should be in the range suggested in equation 1. Moreover, this result allows to safely ignore trial 4 in all of the 5 pairs since it disrupted the model.

In figure 2 The x-axis is chosen to be the batch size since each trial of the 3 initial trials in table

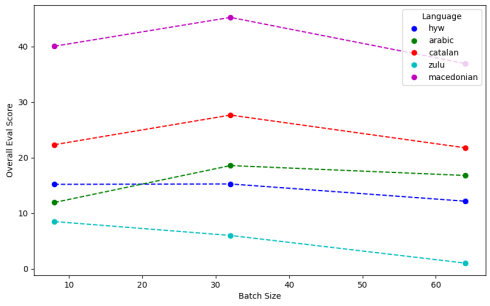

Figure 2: Overall Eval Score vs Batch Size by Language

3 have a distinct batch size, therefore the batch size will be treated as a label to each trial and the analysis will be over the OES and all of the hyper-parameters in each trial.

There is a positive correlation between the trial BLEU score and the OES difference between the trials. As seen in figure 2 when the OES is in the 40s range, the difference between the first and second trail is 5.1, and when it is in the high 20s, the difference is 5.3. When the range is in the low 20s the difference is 0.06 and when it is below 10, the difference is -2.5. Since the OES is composed of the BLEU score times the HES, the HES is a value between 0 and 1 and is only meant to modify the BLEU score. The main value of the OES comes from the BLEU score. Therefore, the higher the BLEU score the higher the difference is. However, there is an outlier in the Arabic data, the difference is the highest at 6.6. This outlier might have different reasons. For example, LA shares fewer words with MSA than other language pairs which might disrupt the learning process.

Since the BLEU score measures the accuracy of the output, this correlation can be proven as

causation relation between batch size and accuracy.

However, to prove this we need to isolate weight decay from the other variables. It was proven that the number of epochs should be between 6-7 that the exact number does not make big differences, and that the learning rate can be any number in the range shown in equation 1. The number of epochs and learning rate were isolated, and only weight decay was left to isolate.

Weight decay did not highly influence the learning process. As seen in table 3 weight decay is the same in trials 2 and 3 of all language pairs. In all 5 pairs, the 3rd trial is the worst, and in 4 of the 5 pairs, the 2nd trial is the best. From this, it was suspected the minor difference weight decay had in the context of this experiment. To further confirm this, an experiment on Catalan was run with 3 different weight decays (0.22, 0.02, 0.002) while keeping all other factors constant. Firstly, it was noticed that all of the models gave almost identical outputs. Moreover, the BLEU score after each epoch in each of the 3 trials was recorded in the table 5;

ANOVA test was conducted with the null hypothesis being "weight decay does not have a significant effect on this experiment". The p-value was 0.981 which is way bigger than the threshold of 0.05, therefore the null hypothesis is not rejected proving that weight decay is insignificant in this experiment. By extension and since all other languages had similar behaviour during training, we can say weight decay did not significantly affect the training process in this context.

Finally, after proving the insignificance of weight decay along with proving that the learning rate should be of any value in the equation 1 and that the number of epochs could be around 6-7, the relationship between the BLEU score, or the OES, and the batch size can be safely analyzed.

Medium and low batch sizes appear to be more effective for transferring learning from high-resourced to low-resourced languages. As shown in Figure 2, the medium batch size of 32 achieved the best OES in 4 out of 5 language pairs, while a batch size of 8 was most effective for Zulu. This suggests that medium batch sizes strike a balance between training efficiency and the model's ability to capture data details, whereas smaller batch sizes, such as 8, may be advantageous when the model needs to focus on fewer, more intricate details.

Changes in the similarity of language pairs cause changes in the required batch size. To measure this similarity, we translated the sentences in Table 4 into the respective languages and computed the Liechtenstein Distance as a similarity score. The results indicated that language pairs like Zulu and Hausa, with a Liechtenstein Distance of 0.55, were less similar compared to pairs with scores ranging from 0.74 to 0.85. This lower similarity suggests that Zulu and Hausa, being less mutually intelligible, require more granular learning, which is why a smaller batch size of 8 was found to be optimal. Conversely, languages with higher similarity did not need such detailed focus, making medium batch sizes effective for them.

In summary, A causal relationship between language similarity and the required batch size during transfer learning for low-resourced languages is suggested by the results of the experiments. Variables such as learning rate and weight decay were isolated and controlled, revealing that a higher similarity in language pairs is correlated with a need for larger batch sizes. This correlation is supported by empirical data indicating that better translation results are generally achieved with medium batch sizes, particularly when languages are more similar. Conversely, smaller batch sizes are found to be beneficial for less similar language pairs, accommodating the need for more detailed learning. The consistency of these findings across different language pairs emphasizes the importance of adjusting batch size based on language similarity to optimize performance in machine translation tasks.

4 Conclusion

In conclusion, this paper has provided a comprehensive examination of machine translation transfer learning across diverse linguistic pairs, spanning different language families and degrees of resource availability. By employing a robust experimental framework involving five distinct language pairs, we have demonstrated that transfer learning can be effectively applied across languages with varied linguistic characteristics, including Semitic, Bantu, Indo-European, and language isolates.

The experiments highlighted several critical findings, summarised below:

1. Transfer learning in machine translation should be done with similar languages; the more similar the languages are, the better.

2. During the fine-tuning process in this context, the learning rate must be in the range $n \times 10^{-4}$, but the exact value does not matter.

Weight Decay	1	2	3	4	5	6	7	8
0.2	37.4	39.5	39.8	40.3	40.4	40.7	40.7	41.2
0.02	37.2	39.3	39.6	40.3	40.5	40.5	40.8	40.9
0.002	37.2	39.1	39.9	40.5	40.4	40.5	40.7	41.0

Table 5: BLEU scores for different weight decays.

3. During the fine-tuning process in this context, the number of epochs does not need to be very high; epochs between 6-7 are cross-linguistically enough.

4. Weight decay does not have a significant impact on the fine-tuning process in this context.

5. Changes in the similarity of language pairs cause changes in the required batch size.

Hyper-parameter tuning emerged as a crucial factor influencing model performance. The results indicate that the learning rate has a substantial impact on the model's ability to generalize, while the number of epochs required for effective fine-tuning is relatively low due to the pre-existing knowledge in the source models. Weight decay, however, was found to have minimal effect, suggesting that its optimization may be less critical in the context of this study. Batch size, on the other hand, demonstrated varying effects depending on the language pair, with medium batch sizes generally proving more effective, except in cases where smaller sizes were beneficial due to the intricate details of less similar language pairs.

Overall, our findings contribute to a nuanced understanding of transfer learning in machine translation, providing evidence that, while general principles apply, specific parameter settings and language pair characteristics play a crucial role in achieving optimal results.

On a final note, it must be acknowledged that generative AI was used to write minimal parts of this paper and to review the coherence and grammatical issues, along with reviewing small parts of the code used in the experiment.

5 Limitations

While this study provides valuable insights into the application of transfer learning in machine translation, it is not without its limitations. One key limitation is the relatively narrow selection of language pairs. Although we covered a range of linguistic families, further investigation involving additional language pairs, particularly those with less similar characteristics, would help to generalize the findings.

Additionally, the study focused on relatively simple model architectures, and more advanced models might yield different results, particularly in terms of parameter optimization. The limited exploration of hyperparameter optimization techniques could also be expanded in future research to identify potentially more effective configurations.

Finally, although batch size was found to vary based on the language pair, the exploration of different batch sizes for a wider range of languages would help to further refine the conclusions. Future research could also investigate the impact of other hyperparameters, such as optimizer type or learning rate schedules, to provide a more comprehensive understanding of the factors influencing machine translation performance.

References

Ari Nubar Boyacıoğlu and Jan Niehues. 2024. The first parallel corpus and neural machine translation model of western armenian and english. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 345–356, Torino, Italia. ELRA and ICCL. ACL Anthology, https://aclanthology.org/2024.sigul-1.42.

NLLB Team et al. 2022. No language left behind: Scaling human-centered machine translation. Hugging Face, https://huggingface.co/datasets/allenai/nllb.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Helsinki-NLP. 2024. Helsinki-nlp/opus-mt-en-de. Hugging Face, https://huggingface.co/Helsinki-NLP/opus-mt-en-de.

Aiusha V. Hujon, Thoudam Doren Singh, and Khwairakpam Amitab. 2023. Transfer learning based neural machine translation of english-khasi on low-resource

settings. *Procedia Computer Science*, 00:000–000. Available online at www.sciencedirect.com, Accessed 1 Aug. 2024.

Vishal Karda. 2023. Fine-tuning a transformer model for neural machine translation. Medium, 17 Aug. 2023, shorturl.at/GbpWZ.

Tom Kocmi and Ondrej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*. Association for Computational Linguistics.

Benjamin Marie. 2024. The kaitchup – ai on a budget: A guide on hyperparameters and training arguments for fine-tuning llms. The Kaitchup, 1 Apr. 2024, Generated with DALL-E.

Helsinki NLP. 2024. Helsinki neural machine translation (nmt) project. University of Helsinki, https://opus.nlpl.eu.

Google Colab Notebook. 2024. Available at: https://colab.research.google.com/drive/1kC2XR2JttGBw_9ZV7nhH3l1vqZAVT5zj. Accessed 24 July 2024.

A Evaluation Summary

language	index	learning rate	weight decay	batch size	num_epochs	BLEU	sentence 1	sentence 2	sentence 3	human_eval_score	overall_eval_score
hyw	1	0.0002	0.02	8	12	22.8	2	1	3	0.6	15.2
	2	0.0002	0.2	32	8	22.9	2	1	3	0.6	15.2
	3	0.0004	0.2	64	5	21.9	2	1	3	0.5	12.1
	4	0.06	0.2	8	8	0.007	1	1	1	0.0	0.0
la	1	0.0002	0.02	8	12	17.9	2	2	2	0.6	11.9
	2	0.0002	0.2	32	8	20.9	3	2	3	0.8	18.5
	3	0.0004	0.2	64	5	18.9	2	3	3	0.8	16.8
	4	0.06	0.2	8	8	0.0007	1	1	1	0.0	0.0
ca	1	0.0002	0.02	8	12	40.2	2	2	1	0.5	22.3
	2	0.0002	0.2	32	8	41.5	2	1	3	0.6	27.6
	3	0.0008	0.2	64	5	39.2	2	2	1	0.5	21.7
	4	0.06	0.14	16	10	1e-05	1	1	1	0.0	0.0
zu	1	0.0002	0.02	8	12	15.3	2	1	2	0.5	8.5
	2	0.0002	0.2	32	8	10.8	2	1	2	0.5	6.0
	3	0.0008	0.2	64	5	2.3	2	1	1	0.4	1.02
	4	0.06	0.14	16	10	0.0007	1	1	1	0.0	0.0
mk	1	0.0002	0.02	8	12	51.5	3	2	2	0.7	40.05
	2	0.0002	0.2	32	8	50.9	3	3	2	0.8	45.2
	3	0.0008	0.2	64	5	47.5	3	3	1	0.7	36.9
	4	0.06	0.14	16	10	0.0001	1	1	1	0.0	0.0

Figure 3: Evaluation summary of the 16 models