Bidirectional Hindi-Punjabi Machine Translation

Mukund K Roy [1], Karunesh K Arora [1] and Sunita Arora [1]

[1] *SNLP Lab, CDAC, Noida, Uttar Pradesh, India*

Abstract

This paper presents the development and comprehensive assessment of a Hindi-Punjabi machine translation system tailored specifically for the MTIL (Machine Translation for Indian Languages) track of FIRE 2023. Leveraging neural machine translation techniques, we developed a robust translation model to facilitate seamless communication between Hindi and Punjabi, two prominent Indian languages despite low resource availability. The methodology involved fine-tuning a pretrained NLLB-1.3B to adapt to the Hindi-Punjabi translation task. To evaluate the efficacy of the translation system, we conducted comprehensive experiments using standard evaluation metrics on FLORES, as well as, on our own testset. Our results demonstrate promising performance of Punjabi-Hindi language pair, showcasing highest score in terms of BLEU, chrF and TER metrics across all domain specific translations in the track. Similarly, our Hindi-Punjabi pair also scored the highest in all domains except Governance domain where our chrF and COMET scores marginally second highest though BLEU and TER were still the highest. The findings underscore the viability and potential of our developed machine translation system, contributing to the advancement of translation technology for Indian languages in diverse applications.

Keywords

Machine Translation, Hindi-Punjabi, Transformer based NMT, NLLB-200, Finetuning

1. Introduction

India, renowned for its linguistic diversity, houses languages like Hindi and Punjabi that wield significant cultural and regional importance. However, despite their prevalence, the effective translation between Hindi and Punjabi remains a considerable challenge due to inherent linguistic intricacies and lack of good quality and large parallel resources. Though Hindi and Punjabi belongs to same Indo-Aryan family, they diverge significantly in script and vocabulary. Hindi uses Devanagari script while Punjabi is predominantly written in the Gurmukhi script. The dissimilarities in script and lack of parallel resources pose challenge for machine translation between the language pair. This necessitates careful handling during the translation process to ensure accurate conversion without loss of semantic or contextual meaning.

The vocabulary also presents another hurdle, with both languages exhibiting distinct lexical items, idiomatic expressions, and dialectical variations. The challenge lies in accurately capturing the essence of these linguistic intricacies during translation to ensure natural and contextually relevant output. In light of these complexities, our work aims to address these challenges by leveraging advanced Neural Machine Translation techniques to facilitate accurate, contextually relevant, and fluent translations between Hindi and Punjabi.

Neural machine translation (NMT) has reformed the area of machine translation (MT) in recent years, achieving significant improvements in translation quality compared to traditional statistical MT (SMT) approaches. NMT is based on artificial neural networks (ANNs), which are capable of learning complex relationships between languages from large amounts of training data.

Forum for Information Retrieval and Evaluation, 15-18 December 2023, Panjim, India

EMAIL: mukundkumarroy@cdac.in (M. Roy); karunesharora@cdac.in (K. Arora); sunitaarora@cdac.in (S. Arora)

One of the key advancements in NMT was the development of the encoder-decoder architecture (Sutskever et al., 2014) [1] wherein the encoder takes a source-language sentence as input and generates a representation of its meaning. After that, the decoder part takes this representation and generates a target-language sentence that is equivalent in meaning to the source sentence. Another major advancement was from (Bahdanau et al., 2014; Vaswani et al., 2017) [2] where they developed attention mechanisms. Attention mechanisms allow the decoder to focus on different parts of the source sentence when generating the target sentence. This helps to improve the accuracy of translations by allowing the decoder to pay attention to the most relevant information in the source sentence. The development of transformer-based NMT models (Vaswani et al., 2017) [3] had also been a significant breakthrough. Transformer models are based on a self-attention mechanism that allows them to process all parts of the source sentence simultaneously, without the need for recurrent neural networks (RNNs). This makes transformer models more efficient and scalable than RNN-based NMT models. To continue further, (Tang et al., 2018; Firat et al., 2016, Johnson et al, 2017) [4][5][6] developed a multilingual models that can translate between multiple languages. These models are trained on large amounts of data in multiple languages, which allows them to learn the relationships between different languages more effectively. One notable example of a multilingual NMT model is NLLB-200, developed by Meta AI (Costa-jussà et al., 2022) [7]. NLLB-200 is a single AI model that can translate across 200 different languages with state-of-the-art results. NLLB-200 has the ability to translate between a wide ranges of languages, including many low-resource languages. The NLLB model has been shown to achieve state-of-the-art results on a variety of benchmark datasets.

This paper outlines the building a high-performance translation model for the Hindi-Punjabi language pair which poses a significant challenge due to the scarcity of parallel training data. To address this limitation, we employed NLLB-200-3.3B, a state-of-the-art multilingual neural machine translation model designed to excel in low-resource settings. NLLB-200's ability to effectively utilize data from multiple languages, including Hindi and Punjabi, made it an ideal starting point for our translation model. By fine-tuning NLLB-200 on a carefully curated dataset of parallel Hindi-Punjabi sentences, we were able to achieve significant improvements in translation accuracy in compared to training a transformer model from scratch using the available corpus. The resulting translation model demonstrates the potential of NLLB-200 for low-resource machine translation tasks and its ability to bridge the communication gap between speakers of Hindi and Punjabi.

Our team participated in the FIRE-2023 MTIL challenge for both Punjabi to Hindi and Hindi to Punjabi language pairs. We employed NLLB-200, a cutting-edge machine translation model optimized for resource-constrained environments, as the foundation for our submissions. We further enhanced the model's performance by training it on general domain data, governance domain data, and healthcare domain data as described in next section. The effectiveness of our model was evaluated using chrF (character-level F-score) [8], the official metric for the task. Our model achieved the highest chrF scores Punjabi to Hindi in all domains. For Hindi to Punjabi language pairs, except for Governance domain, our model again achieved highest chrF score among all participating teams.

2. Dataset

The dataset for this task consisted of a parallel corpus of Hindi-Punjabi sentence pairs collected from diverse domains, including General, Agriculture, Tourism, Education, Science & Technology, Governance, Health and News articles. Overall 140K parallel sentences were collected and curated for this task. Majority of the corpus, though human translated, but needed some vetting, cleaning and preprocessing before sending for training of the translation model. The dataset was split into training, development, and test sets to facilitate model training and evaluation. In addition, FLORES test set [9] is also used to evaluate the system which contains 1012 sentence pairs.

3. Methodology

In order to build Hindi-Punjabi and Punjabi-Hindi translation models, we fine-tuned the NLLB-200-1.3B pretrained model with our corpus. For evaluating both the models, we used two datasets i.e. our

own (CDACN) test set containing 1000 sentences of mixed of domain and another publicly available FLORES test set. It was necessary to maintain the fairness and avoid biasness.

For our training purpose, we used SOTA Opennmt-Py toolkit which provides different configurations of building NMT models to play with. In this work, we built Transformer model from scratch and Fine-tuned model using the same toolkit. We used following methodology to train our models:

3.1. Data preprocessing

Preprocessing plays a pivotal role in training Neural Machine Translation (NMT) models within the OpenNMT-py toolkit [10], serving as the foundational step in transforming raw textual data into a format suitable for effective model learning and translation. Primarily, the preprocessing workflow encompasses cleaning, tokenization, normalization, subword segmentation, and vocabulary construction. Tokenization, the initial step in preprocessing, involves breaking the text into smaller linguistic units, typically words or subword units, facilitating the model's understanding of the input. Subword tokenization, often implemented using Byte Pair Encoding (BPE) or SentencePiece, is widely preferred for its ability to handle Out of Vocabulary (OOV) words by splitting them into subword units, promoting better generalization and handling of unseen vocabulary during translation. Subword segmentation, using BPE or SentencePiece, further refines the tokenization process by breaking down words into smaller subword units based on their frequency of occurrence within the dataset. Normalization follows tokenization and involves standardizing the text by resolving issues such as punctuation, casing, and other linguistic variations. Vocabulary construction is another pivotal aspect of preprocessing in OpenNMT-py. It involves building a vocabulary set that comprises the most frequent tokens or subword units from the training data. Careful selection of the vocabulary size is crucial as it directly impacts the model's ability to generalize while also influencing the computational requirements. The vocabulary size must strike a balance between coverage of commonly occurring tokens and efficiency in model training. In OpenNMT-py, preprocessing is streamlined using the '*preprocess.py'* script. This script takes the raw text data and performs the necessary preprocessing steps, generating vocabulary files and training and validation datasets in a format compatible with the NMT model.

3.2. Model Training

As stated earlier, we fine-tuned the NLLB-200 pretrained model with our training dataset. We used the NLLB-200-3.3B which is basically a transformer-based encoder-decoder architecture with 3.3B parameters. It is trained on over 2TB of text data in 1220 language pairs, including 202 languages. The model is mainly intended for research in MT, primarily for low-resourced languages. It can perform translation of single sentence between 200 languages. Due to the dependency on this model, we customized the SentencePiece model [11] to work on OpenNMT toolkit and used it as tokenization method. The architecture of training model was also modified accordingly by incorporating 24 layers Transformer Encoder Decoder. The Feed Forward Network (FFN) now had 8192 hidden units and word vector size was doubled to become 1024. Similarly optimizing methods is also modified to use Standard Gradient Descent method.

The Model training begins with feeding the preprocessed parallel training data into the fine-tuning framework. The framework splits the data into batches for efficient training. During the forward pass, the input sentence in the source language is passed through the encoder of the NLLB-200 model. The encoder generates a representation of the input sentence's meaning. The attention mechanism of this architecture allows the decoder to concentrate only on relevant parts of the encoder's representation while generating the output in the target language. The decoder generates a sequence of words in the target language, one word at a time, based on the encoder's representation and the attention mechanism. The predicted output sequence is compared to the actual target sequence to calculate the loss, which represents the model's error. The loss is propagated backward through the model to update the weights of the encoder and decoder. The optimizer adjusts the model's weights to minimize the loss, gradually improving the model's ability to translate sentences accurately.

4. Evaluation

In this section, we have discussed about evaluation of our fine-tuned model using BLEU (Bilingual Evaluation Understudy)[12], chrF (character-level F-score) [8], COMET (Crosslingual Optimized Metric for Evaluation of Translation) [13] and TER (Translation Edit Rate) [14] metrics. BLEU is a popular metric for evaluating machine translation (MT) systems. It is a precision-based metric that calculates the percentage of n-grams (sequences of n words) that are correct in the translated output compared to the reference translation. chrF is a metric for evaluating MT systems that is based on character-level overlap between the translated output and the reference translation. chrF scores range from 0 to 1, with 1 being a perfect score. chrF is less sensitive to word order than BLEU and is more forgiving of errors in morphology and syntax. TER is a metric for evaluating MT systems that is based on the number of edits (insertions, deletions, and substitutions) that need to be made to the translated output to convert it into the reference translation. TER scores range from 0 to 1, with 0 being a perfect score. In table 1, different metric scores on two test sets has been given.

We also present our performance in FIRE 2023 MTIL track which aims to create a strong machine translation system for converting text from one Indian language to another Indian language. There are two main jobs in this track. Task 1 involves making a translation model for general domain, working across 12 different Indian language pairs. Task 2, which is more specific, needs translation models focused on the Governance and Healthcare domain.

Table 1
Evaluation scores on CDACN and FLORES Test set

Language pair	BLEU	chrF	TER
CDACN Test set			
Punjabi-Hindi	50.3	69.8	31.8
Hindi-Punjabi	38.3	62.7	40.1
FLORES Test set			
Punjabi-Hindi	28.2	53.9	59.3
Hindi-Punjabi	21.4	48.1	64.7

Table 2
FIRE 2023 MTIL track Official evaluation scores of two translation tasks on different metrics

Domain	Language Pair	BLEU	chrF	chrF++	TER	COMET
General	Punjabi-Hindi	62.1954	77.4556	76.6006	22.2312	0.8366
General	Hindi-Punjabi	50.9394	69.7897	68.1843	38.0883	0.8454
Governance	Punjabi-Hindi	33.1194	56.1692	54.6360	51.5544	0.8180
Governance	Hindi-Punjabi	56.8942	73.6951	72.8590	25.7565	0.8169
Health	Punjabi-Hindi	37.5176	60.8540	59.5213	42.0599	0.8379
Health	Hindi-Punjabi	65.0554	79.5775	78.8151	20.4537	0.8520

5. Results

Upon analyzing Table 1 and Table 2 of evaluation scores, it can be observed that the model Punjabi to Hindi translation system is performing better than the Hindi to Punjabi system, although the dataset used is same for both the direction. One of the main reason is that Punjabi is a more inflected language than Hindi, which means that there are more cues for the translation systems to use when translating from Punjabi to Hindi.

On our internal CDACN and Flores testsets, the BLEU scores for all translation tasks range from 21.4 to 50.3. This suggests that our both translation systems are able to produce translations that are of reasonable quality. The chrF scores for all translation tasks range from 48.1 to 69.8. This suggests that the translation systems are able to produce translations that are fluent and natural-sounding. The TER scores for all translation tasks range from 31.8 to 64.7. This suggests that the translation systems are able to produce translations that are relatively accurate.

In MTIL challenge chrF is the official metric of evaluation. Here our systems scored the highest of all, reaching the score as high as 79.5775 and lowest being 60.8540 across all domain specific tasks. BLEU and TER scores also corresponds the models' capacity to translate domain-specific language with proficiency showcasing their robustness and adaptability.

6. Conclusion

In this paper, we presented our work of building Hindi-Punjabi bidirectional translation model using fine-tuning methodology. Our system utilized the NLLB-200-3.3B pre-trained model to translate between Hindi and Punjabi across the General, Governance, and Healthcare domains. Our models achieved promising results in the MTIL track challenge in FIRE 2003, highlighting the efficacy of the methodology applied to these machine translation models. These empirical findings also establish a foundation future works of further advancements and exploration in the realm of domain-specific machine translation.

7. Acknowledgements

We are sincerely thankful to the Ministry of Electronics and Information technology (Meity) for funding the NLTM-ILTM. We also express our thanks to Shri Vivek Khaneja, Executive Director, CDAC Noida for his constant support and motivation. Finally, we are thankful to the NPSF-AIRAWAT for providing the GPU compute infrastructure.

8. References

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In NeurIPS. 3104--3112

[2] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to attend and translate." arXiv preprint arXiv:1409.1055 (2014).

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 1706.03762v7 (2017).

[4] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, A. Fan, Multilingual translation with extensible multilingual pretraining and finetuning, https://doi.org/10.48550/arXiv.2008.00401 (2020)

[5] O. Firat, K. Cho, Y. Bengio, Multi-way, multilingual neural machine translation with a shared attention mechanism, Association for Computational Linguistics, 2016, pp. 866–875. doi:10.18653/v1/N16-1101.

[6] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, J. Dean, Google's multilingual neural machine translation system: Enabling zero-shot translation, Transactions of the Association for Computational Linguistics 5 (2017) 339–351. doi:10.1162/tacl_a_00065.).

[7] N. Team, M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, A. Sun, S. Wang, G. Wenzek, A. Youngblood, B. Akula, L. Barrault, G. M. Gonzalez, P. Hansanti, J. Hoffman, S. Jarrett, K. R.Sadagopan, D. Rowe, S. Spruit, C. Tran, P. Andrews, N. F. Ayan, S. Bhosale, S. Edunov, A. Fan, C. Gao, V. Goswami, F.

Guzmán, P. Koehn, A. Mourachko, C. Ropers, S. Saleem, H. Schwenk, J. Wang, No language left behind: Scaling human-centered machine translation, arXiv:2207.04672v3 [(2022).
[8] M. Popović, chrf: character n-gram f-score for automatic mt evaluation, Association for Computational Linguistics, 2015, pp. 392–395. doi:10.18653/v1/W15-3049.
[9] F. Guzmán, P.J Chen, M. Ott, J. Pino, G. Lample, P. Koehn, V. Chaudhary, and M.A.Ranzato.,The FLORES Evaluation Datasets for Low-Resource Machine Translation: Nepali–English and Sinhala–English, 2019 In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), p. 6098–6111, Hong Kong, China. Association for Computational Linguistics.
[10] https://github.com/OpenNMT/OpenNMT-py
[11] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing", 2018, Proc. EMNLP, pp. 66-71.
[12] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu, Association for Computational Linguistics, 2001, p. 311. doi:10.3115/1073083.1073135.
[13] R. Rei, C. Stewart, A. C. Farinha, A. Lavie, Comet: A neural framework for mt evaluation, Association for Computational Linguistics, 2020, pp. 2685–2702. doi:10.18653/v1/2020.emnlp-main.213.
[14] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. "A Study of Translation Edit Rate with Targeted Human Annotation". In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pages 223–231.
[15] B. Zoph, D. Yuret, J. May, K. Knight, Transfer learning for low-resource neural machine translation, Association for Computational Linguistics, 2016, pp. 1568–1575. doi:10.18653/v1/D16-1163.
[16] S. Gangopadhyay, G. Epili, P. Majumder, B. Gain, R. Appicharla, A. Ekbal, D. Sharma, Overview of MTIL Track at FIRE 2023: Machine Translation for Indian Languages, in Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2023, 2023.
[17] S. Gangopadhyay, G. Epili, P. Majumder, B. Gain, R. Appicharla, A. Ekbal, D. Sharma, Overview of MTIL Track at FIRE 2023: Machine Translation for Indian Languages. In *Working Notes of FIRE'23, 2023*.