The Effect of Similarities Between Indian Languages in Zero-Shot Translations

Aman Jaiman
University of Maryland
amanjaiman@outlook.com

Abstract

Zero-shot translation methods allow neural machine translation to succeed even when there is a lack of parallel data for a particular language pair. We build upon prior work to perform low-resource zero-shot translation across seven Indian languages spread across two language families. We analyze their translation quality, then turn to analyzing their performance from the lens of language similarity. Specifically, we examine the correlation between various similarity metrics and the translation quality in the zero-shot setting. We find that for Indo-Aryan languages, language similarity can be a good predictor for the quality of the translation. Out-of-family translations, such as from Telugu to an Indo-Aryan language, can be correlated with language similarity as well. Geographical proximity and Levenshtein distance seem to play the biggest role in translation quality in the zero-shot setting.

1 Introduction

Low-resource languages struggle to achieve the high performance promised by neural machine translation due to the limited amounts of data that can be used for training. This calls for new avenues to enable machine translation for languages where there is no data. Low-resource machine translation is especially important for Indian languages. According to Banerjee et al. (2005), there are 325 languages that are used within communities in India, and 31 of them are each spoken by over one million people in India. Yet, only four of those languages are classified as level 3 or above by Joshi et al. (2020) while all other Indian languages are low-resource.

There is especially a lack of parallel data between Indian languages; most of the parallel data for Indian languages is paired with English. While work in translating Indian languages to and from English is important, the people of India also require translations for Indian languages they do not speak. The 2011 Census of India reported that nearly 500 million Indians were domestic migrants, showcasing the importance of translation services just within India.

One approach for translating between languages where there is no parallel data is to use zero-shot translation. As proposed in Johnson et al. (2017), in the zero-shot setting we have two languages, X and Y, which are paired with a third language Z. The model is trained on sentence pairs from X-Z and Y-Z in both directions. At test time, the model should be able to translate X-Y in both directions without ever seeing the data during train time.

The work in this paper is built upon the work done by Huidrom and Lepage (2020). In their work, they perform single model, multilingual model, and zero-shot model experiments using four Indian languages languages from the PMIndia dataset (Haddow and Kirefu, 2020): Assamese, Bengali, Hindi, and Manipuri. The data is low-resource, with training being done with just 5,000 sentences per language pair. Their zero-shot work focuses only on translating from Manipuri to X (one of the other three languages), as Manipuri is the lowest resource language they look at.

We start by replicating a portion of Huidrom and Lepage (2020) to make sure our experimental setup is the same. We then perform an extension study motivated by the points they make in their analysis of zero-shot translation in Indian languages. In their work, they find the most success doing zero-shot translations from Manipuri to Bengali, which they attribute to the fact that Bengali has lexical influence over Manipuri. This poses an interesting question of how much zero-shot translation performance depends on language similarities.

Our work is specifically interested in answering the following question: How do different similarity measures between two Indian languages impact the performance of zero-shot translation between those languages? In particular, we perform experiments for zero-shot translations across seven different languages - resulting in 21 language pairs - and analyze their results along with metrics of similarities. All experiments are done in the same low-resource setting as Huidrom and Lepage (2020).

The question posed is important because it allows us generalize their zero-shot findings for other common-but-low-resource Indian languages. This work also aims to put languages other than Hindi in the spotlight in hopes of garnering attention in future work towards increasing the resource level for those other languages. In addition, our work could enable future research in improving machine translation systems for similar languages where there is not much parallel data. Lastly, research on other languages that are similar, such as Spanish and Portuguese, may benefit from these findings.

2 Indian Language Similarities

Noting that similarities in languages may play a role in translation quality, we turn to Kumar et al. (2021) to examine similarities in Indian languages. This work uses nine languages (the seven we use here along with Bengali and Marathi) and performs various string matching algorithms between language pairs to generate similarity scores for every pair. The algorithms used are token overlap, Levenshtein distance (Levenshtein, 1966), longest common subsequence (Larsen, 1998), and shingle (q-gram) similarity (Kondrak, 2005).

Using the results from Kumar et al. (2021) and higher-level similarity metrics, such as language-family and geographical proximity, we compare the similarity scores to our model's performance to reveal insights about similarities and zero-shot translations.

3 Experimental Setup

3.1 Data

We use the PMIndia dataset (Haddow and Kirefu, 2020) for all of our experiments. The dataset consists of monolingual data for 13 Indian languages and parallel data for the same 13 Indian languages paired with English. All of the data comes from official documents from the Prime Minister's Office of the Government of India.

We use parallel data for seven of the Indian languages provided by the dataset. The languages are chosen by language family in order to perform experiments within and across different language families. We choose four Indo-Aryan languages: Gujarati (gu), Hindi (hi), Punjabi (pa), and Urdu (ur), and three Dravidian languages: Malayalam (ml), Tamil (ta), and Telugu (te). The reason for these specific languages was the authors' familiarity with the languages to assist with certain evaluations. Additionally, all the languages chosen were analyzed by Haddow and Kirefu (2020). Table 1 shows the number of unique sentence pairs for each of the language pairs we use. All language pairs have less than 60,000 sentence pairs, as we are in the low-resource setting.

Language Pair	Sentences Pairs
gu-en	50,380
hi-en	56,831
pa-en	34,699
ur-en	11,167
ml-en	33,669
ta-en	39,526
te-en	40,283

Table 1: Unique sentences pairs for each of the Indian Language-English parallel data we use. The first four are the Indo-Aryan languages, and the last three are Dravidian languages.

We use the same dataset split as Huidrom and Lepage (2020). For each model, we have 4 language pairs using English (en) and the two Indian languages (in1, in2): en-in1, in1-en, en-in2, and in2-en. For each language pair, we randomly select 7,000 sentences, which are then broken up into 5,000 sentences for training and 1,000 sentences each for validation and testing. An important thing to note here is that pivoting using English is not done because there are a limited number of shared sentences across the language pairs. We add an artificial token to the beginning of each source sentence to indicate the source and target language. For example, if we are translating from English to Hindi, we add `<en2hi>` to the beginning of the sentence.

3.2 Tools

We use the OpenNMT-py toolkit (Klein et al., 2017) for all of the experiments in order to replicate the pre-processing of the data and the model structure that was used by Huidrom and Lepage (2020). We write YAML scripts with the data location and training parameters, which OpenNMT-py uses to build the vocabulary and train the model. We also use `subword-nmt` for data pre-processing and `sacreBLEU` (Post, 2018) for evaluation using

Model Parameters	
RNN Size	500
Encoder Type	BRNN
Decoder Type	RNN
Layers	2
Attention	Global
Dropout	0.3
Optimization	
Batch Type	Sentence
Batch Size	64
Optimizer	Adam
Learning Rate	0.001

Table 2: Main model parameters for the default model provided by OpenNMT-py. We use this model as it is what is used by Huidrom and Lepage (2020)

BLEU (Papineni et al., 2002).

We ran all of the models on Google Colaboratory Pro (Colab) utilizing their GPU resources. The type of GPU varied between a Tesla T4 or a Tesla P100-PCIE-16GB. This was based on what was provided by Colab. Using either GPU gave us similar runtimes for training, with the average train time being just under 25 minutes.

Our yaml templates can be found on our Github[1], while the data generation, training, and testing code can be found on our Google Colab[2].

3.3 Model

We use the default model that is provided by OpenNMT-py. The model used is a 2-layer seq2seq architecture that uses global attention. The encoder is a bidirectional LSTM and the decoder is an LSTM. Both layers have 500 hidden units. The full hyperparameters are presented in Table 2. The default model starts learning rate decay at 50,000 training steps, but the model is only run for 10,000 steps. Therefore, there is no learning rate decay during our training.

3.4 Training

For each language pair, we first randomly generate the necessary training, validation, and testing datasets. Next, the training data is concatenated and we apply Byte-Pair Encoding (BPE) to all the data with 10,000 merge operations. The necessary training token is then pre-appended to all of the source sentences based on the source and target language. We generate a shared vocabulary for our languages using the OpenNMT-py scripts. Lastly, We train the model for 10,000 steps, running validation every 1,000 training steps.

[1]https://github.com/amanjaiman/ILTUZ

[2]https://colab.research.google.com/drive/13Htu7-8Bz0hLnkiJ3givQfTJCgpi5RCk?usp=sharing

Initial Setup	X -> En, En -> X, Y -> En, En -> Y
Better translation from X to Y	X -> En, En -> Y, Y -> En -> En -> X
Better translation from Y to X	Y -> En -> En -> X, X -> En, En -> Y

Table 3: Initial training data setup was outperformed heavily by the adjustments made. The language that appeared first had better translations out of it during test time. Here, En represents English while X and Y represent two Indian Languages

During our testing, we found that the order of the languages in the training data can make a difference in the output of the model. Initially, the training data was concatenated such that one Indian language paired with English appeared before the other. When this was changed to incorporate a seemingly transitive sequence (X -> En, En -> Y), the model performance increased. Moreover, changing the order of the languages made performance increase: translating from the language that appears first was better than translating from the second language. Examples of our training data setup are presented in Table 3.

We end up with 28 models, one for each language pair. The models are evaluated on the respective test sets using BLEU.

4 Results

We do two initial experiments to ensure our training setup is the same as that presented in Huidrom and Lepage (2020). The first is replicating their multilingual model by training one model on eight language pairs using the four languages they use paired with English. The second is recreating the zero-shot experiment from Manipuri to Hindi, to verify our results. Both experiments produce comparable results to the original findings, so we move forward with our experiments.

The BLEU scores for each zero-shot translation is shown in Table 4. We notice some general trends in the results. There are some languages that do better as source languages and some that do better as target languages. For example, Telugu does well translating to the Indo-Aryan languages, but the

BLEU scores are lower when the Indo-Aryan languages translate to Telugu. By contrast, translating to Urdu yields better BLEU scores than translating from Urdu.

	Gu	**Hi**	**Pa**	**Ur**	**Ml**	**Ta**	**Te**
Gu	x	2.0	1.0	1.2	0.3	0.2	0.7
Hi	2.2	x	1.0	0.8	0.3	0.2	0.6
Pa	1.0	0.5	x	0.7	0.2	1.0	0.1
Ur	0.9	0.7	0.8	x	0.2	0.8	0.3
Ml	0.8	1.1	1.2	1.2	x	0.2	0.1
Ta	0.8	0.9	2.0	1.3	0.1	x	0.1
Te	1.1	1.3	0.6	1.4	0.2	0.1	x

Table 4: BLEU Scores for all models. Source language is on the left, target language is on top. We have split the table into four quadrants to clearly show the inter-family and outer-family translations.

Another thing we note is that the models produce extremely low BLEU scores overall. While the actual 1-gram scores were higher, between 15-18 BLEU, the models were not able to generate proper sentence structure or lost some meaning during the zero-shot translations. As shown in Table 5, the zero-shot translation is able to produce the correct script and keeps a lot of the same characters, but the sentence produced has lost its meaning. While low, the scores are still on par with the results that Huidrom and Lepage (2020) got for their zero-shot translations with similar amounts of data. Increasing the data could increase performance, and it could be done just for the target language, as shown by Huidrom and Lepage (2020).

Translating into the Dravidian languages performed worse compared to translating into the Indo-Aryan languages, although translating from Dravidian to Indo-Aryan languages produced convincing results. These results are similar to those shown in Dewangan et al. (2021), where Dravidian languages also do not perform as well. Our experiments use much less data (46,000 compared to 5,000 training sentences) and are in a zero-shot setting, so we expect our BLEU scores to be lower while following the same trend. Translating within the Dravidian languages was approaching 0, and for this reason, we leave out Dravidian -> Dravidian translations in the similarity analysis.

5 Similarity Analysis

First, we look at translation quality between the two language families. For translating into Gujarati, an Indo-Aryan language, we see that the BLEU score is higher on average when the source language is also from the same family. This result is not shared across the other Indo-Aryan languages, which goes against the findings of Dewangan et al. (2021) where BLEU scores are higher when translating within the same language family. We hypothesize that this may be due to it being in the zero-shot setting or not having enough data in the experiments, but future works needs to be done to determine the reason. Indo-Aryan languages as source languages do produce higher BLEU scores when translating to other Indo-Aryan languages as compared to Dravidian languages.

Another aspect of similarity we look at is geographical proximity, which we define by how far apart the centers of the regions primarily speaking the languages are. We find the best performance when translating between Hindi and Gujarati, which are two languages that are found extremely close to each other within India. Translating into Punjabi, another North Indian language, is also better when the source language is Gujarati and Hindi. When translating from Dravidian to Indo-Aryan languages, Telugu performs the best for three out of the four target languages. This may be attributable to the geographic proximity of Telugu, which is spoken in South Central India, to the Indo-Aryan languages, which are found in North and Central India. Malayalam and Tamil, by contrast, are found at the Southern tip of India, and do not perform as well as Telugu.

Next, we explore the relationship between the various metrics explored by Kumar et al. (2021) and how they relate to translation quality. Their full results can be found in Figure 1 in Appendix A. We use the similarity scores reported for the metrics along with the BLEU score from our work to calculate the Pearson correlation coefficient ρ. For each source language, we pair its target translation BLEU score with the similarity between that target language. The correlation scores are reported in Table 6. We can see that for the Indo-Aryan languages, there is positive correlation between all of the similarity metrics and translation quality (i.e. the higher the similarity the better the output translation). For translating from Gujarati and Hindi particularly, ρ is fairly high. We also note that using Levenshtein distance as the similarity metric generally correlates better to higher BLEU scores, with longest common subsequence and shingle q-

	Hindi Text	Meaning
Translated from Gu to Hi	साथ काम करने और इसके साथ हैं।	Work together and be with it.
Reference Text	भारत में करोड़ों नए आवासों की जरूरत है।	India needs millions of new houses.

Table 5: Example zero-shot translation from Gujarati to Hindi. While the script is correctly translated, the meaning of the sentence is lost.

gram following closely after. Using token overlap seems to be the worst predictor of BLEU scores among the similarity metrics.

Two languages, Malayalam and Tamil, do not show positive correlation between BLEU and similarity scores. We actually see these two languages translating fairly well into the Indo-Aryan languages, which isn't expected by the similarity metrics or geographical proximity. Meanwhile, Telugu does show positive correlation. These results could indicate that because Telugu is more similar to the Indo-Aryan languages, the similarity score can actually be used as a predictor of BLEU score. When the similarity score is too low, like for Malayalam and Tamil, it may be harder to use it as a predictor.

Source	**TO**	**LD**	**LCS**	**QGram**
Gu	0.835	0.968	0.95	0.949
Hi	0.292	0.557	0.57	0.564
Pa	0.065	0.172	0.096	0.144
Ur	0.25	0.443	0.425	0.455
Ml	-0.608	-0.79	-0.719	-0.787
Ta	-0.051	-0.636	-0.523	-0.4
Te	0.555	0.499	0.287	0.401

Table 6: Pearson correlation coefficient between BLEU for source languages and token overlap (TO), Levenshtein distance (LD), longest common subsequence (LCS), and shingle q-gram (QGram).

6 Future Work

Our work opens the door for research in many areas related to this paper. To continue building upon the work done by (Huidrom and Lepage, 2020) we would want to explore the second part of their zero-shot study, where the amount of training data for the target language is increased in stages. In their work this shows massive improvement for the translation quality, improving their Manipuri to Hindi BLEU score by over 7 times the original. This increase in data could also be compared to directly increasing the training data from the start. We use 5,000 training sentences to stay consistent with Huidrom and Lepage (2020), but increasing the data in one of these ways may make it easier to analyze how similarities play a role in the zero-shot translations. Splitting up the low-resource setting from the similarity analysis may prove fruitful to gaining better insights in both respectively. Additionally, in this work we do not perform pivoting because there is a lack of shared sentences between the language pairs. If we increase the data size, pivoting may be possible and could lead to better translation results.

Indian languages tend to have a lot of regional dialects. As a result, nearby neighborhoods may struggle to properly understand each other's languages. Looking into the specific dialects of the data could lead to insights about potential issues for translating. As an example, Hindi is spoken in Rajasthan, but in a different style than in Delhi. Rajasthan is closer to Gujarat, where Gujarati is spoken, so translating to Gujarati from a Rajasthani source could lead to better results than a Delhi source. Similarity scores could be reproduced based on regional dialects to allow for a greater scope of analysis for Indian languages.

This work could also be applied to other languages. This would give us more data on the translation quality between language families and different regions of the world.

7 Conclusion

This work expands on the results from (Huidrom and Lepage, 2020) by producing zero-shot results for seven other Indian languages in the same low-resource data and training regime. Our results are similar to the zero-shot results shown in that paper. We also explore language similarities in Indian languages and see if similar languages produce higher quality translations. We find promising results for Indo-Aryan languages, and in particular see that language similarity is a good predictor for translation quality when translating from Gujarati. Similar conclusions can be drawn from translating from Telugu, showing especially the importance of geographical proximity when translating outside of language families. More work needs to be done for Dravidian languages, however, as translation results were low when Malayalam and Tamil were the target languages. With the current disconnect in machine translation research and domestic Indian communities, we hope this work can spur further research in zero-shot translations for Indian languages.

References

P. Banerjee, S.B.R. Chaudhury, S.K. Das, and B. Adhikari. 2005. *Internal Displacement in South Asia: The Relevance of the UN's Guiding Principles*. SAGE Publications.

Shubham Dewangan, Shreya Alva, Nitish Joshi, and Pushpak Bhattacharyya. 2021. Experience of neural machine translation between indian languages. *Machine Translation*, 35:71–99.

Barry Haddow and Faheem Kirefu. 2020. Pmindia - a collection of parallel corpora of languages of india. *ArXiv*, abs/2001.09907.

Rudali Huidrom and Y. Lepage. 2020. Zero-shot translation among indian languages. In *LORESMT*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Z. Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Grzegorz Kondrak. 2005. N-gram similarity and distance. SPIRE'05, page 115–126, Berlin, Heidelberg. Springer-Verlag.

Sourav Kumar, Salil Aggarwal, Dipti Misra Sharma, and R. Mamidi. 2021. How do different factors impact the inter-language similarity? a case study on indian languages. In *ACL*.

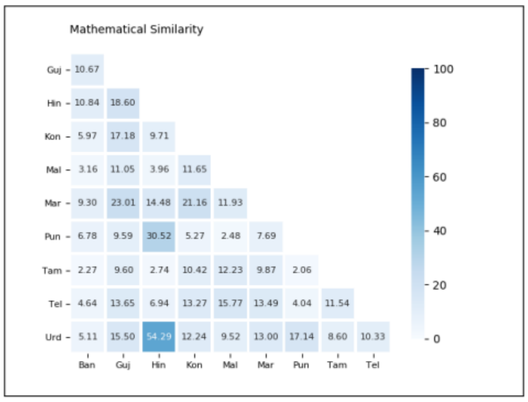
Kim Skak Larsen. 1998. Length of maximal common subsequences. *DAIMI Report Series*, 21.

V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

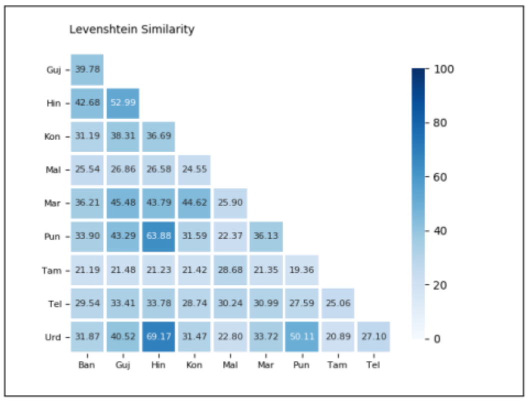
Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

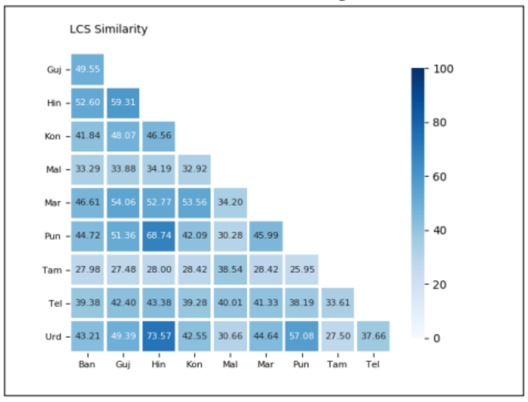
A Similarity Scores



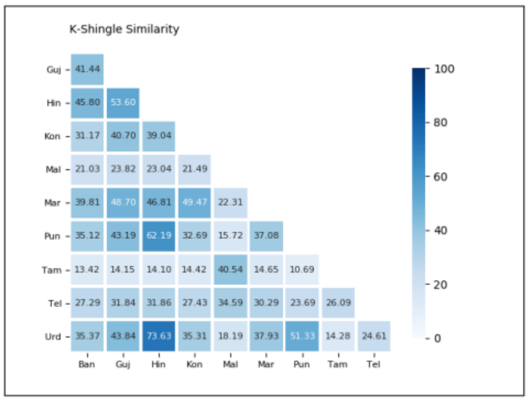
(a) Token Overlap



(b) Levenshtein



(c) Longest Common Subsequence



(d) Shingle Q-gram

Figure 1: Similarity matrix for the different algorithms.
From Kumar et al. (2021)