Analysing different techniques of Sanskrit-Hindi Machine Translation

Keerti Harpavat **Priyanshu Singh** **Rythum Singla** **Rounak Parihar**
Indian Institute of Technology Guwahati
Group No: 2
{keert170101031, singh170101049, rythu170101058, rouna170101056}@iitg.ac.in

Abstract

Sanskrit is one of the oldest language in India and considered the language of the gods, rightly so due to its vast presence in ancient Indian religious texts. However, its popularity in modern India is diminishing. There are very few resources for translating to and from Sanskrit, which is a big reason for the general indifference towards Sanskrit in modern times. Machine Translation(MT), in last few years, has become extra-ordinarily better and is conventionally done using supervised learning techniques. There has been recent research in the unsupervised MT domain which should potentially work well for Sanskrit due to the lack of parallel corpus related to Sanskrit. To bridge the gap between Sanskrit and its modern successor Hindi, we propose building a machine translation system using conventional supervised learning methods viz. Statistical and Neural with the help of Sanskrit-English parallel corpus and comparing it with the recent unsupervised machine translation. We also construct a Sanskrit-English parallel corpus of 34k lines and a monolingual Sanskrit corpus of ∼4.3 million lines. Our code and data resources are available on Github.

1 Introduction

As communication and travel advance, geography is becoming less and less of a barrier for doing interaction. Naturally, for the ease of understanding between different languages, translation between languages is essential. Machine translation is the task of automatically converting source text in one language to text in another language. Given a sequence of text in a source language, there is no one single best translation of that text to another language. This is because of the natural ambiguity and flexibility of human language. This makes it a difficult task to automate. There are many different approaches in practice, to automate this task.

Classical machine translation methods often involve rules for converting text in the source language to the target language. The rules are often developed by linguists and may operate at the lexical, syntactic, or semantic level. There are many issues with this approach, the most prominent being the necessity of expert linguists to formulate such rules. The sheer large number of rules and exceptions required to achieve acceptable performance is also a major bottleneck.

Statistical machine translation (SMT) is a machine translation paradigm where translations are generated on the basis of statistical models, by learning parameters from the analysis of bilingual text corpora, and matching words from source to target language. It performs decently with moderate amount of parallel data.

Neural Machine translation models require large amount(of the order of millions) of parallel corpus to achieve decent results, and generally out-perform statistical MT models when the data is more, but it is computationally more expensive.

Each of the approaches mentioned above have their own advantages and disadvantages, and the best one to use depends on the source and target languages, the availability of data for these languages, and the computational resources.

Machine translation systems now achieve near human-level performance on some language pairs, yet their efficiency strongly relies on the availability of large amounts of parallel sentences, which hinders their applicability to

the majority of language pairs. Such a case of scarce resources exists for one of the oldest language in India, Sanskrit and the most spoken language in modern India, Hindi. Due to this drawback, there have been very few machine translation works for Sanskrit-Hindi. In this project, we follow different approaches for Sanskrit-Hindi machine translation and analyze the efficiency of these approaches.

2 Review of existing work

Statistical models One of the works related to Sanskrit-Hindi MT has been done by (Pandey et al., 2018) in which they train a Moses based statistical MT model on 43k parallel sentences, some of which were taken from Dept. of Public Relations, Madhya Pradesh website which unfortunately for us does not exist anymore and the rest they manually created. They also used 56k monolingual sentences in Hindi(target) language. The data they have considered is specific to only the news and literature domain.

Moses(Koehn et al., 2007) is an established baseline for testing new paraphrasing corpora and models. It is designed to translate from a single source language to a single target language.

Neural models Neural machine translation is an approach to machine translation that uses an artificial neural network like Recurrent Neural Networks(RNN), Long Short Term Memory(LSTM), to predict the likelihood of a sequence of words, typically modelling entire sentences in a single integrated model. Seq2Seq(Sutskever et al., 2014) is one of the most popular neural models for machine translation.

Unsupervised MT This is a relatively new technique in the domain of machine translation, brought forward by (Conneau et al., 2017)(Lample et al., 2018a)(Lample et al., 2018b). In these works, they have shown that it is possible to build a parallel corpus using only monolingual data of the two languages using language modelling and iterative back-translation. We use their open source code as reference for implementing our Sanskrit-Hindi MT system.

3 Dataset

3.1 Unsupervised MT

3.1.1 Training Data(Monolingual)

For the unsupervised approach, we required monolingual corpus for both Sanskrit and Hindi. Since there is no standard dataset for Sanskrit, we created a large dataset ourselves, by scraping the Sanskrit Wikisource, which includes most of the ancient religious texts, like Mahabharata, Ramayana, Puranas, Upanishads, etc. Our dataset contains 4.3 million sentences, which amounts to 32.5 million words. Our dataset is publicly available here. We took a part of the Hindi dataset as provided by (Kunchukuttan et al., 2018). For the ancient Sanskrit texts, we do not have their Hindi translations, so we chose this dataset. However, this comes with its own issues, since the distribution of the two datasets are quite different, so the final BLEU score achieved after translation is below expectations.

3.1.2 Validation & Test Data(Parallel)

It is not easy to find a decent amount of parallel data, even for testing. We scraped Ramcharitmanas Sanskrit and Hindi versions(6000 sentences), and split it equally in validation and test sets. We continue our quest of finding more and more parallel data, but this can be considered a bare minimum to test the performance of our model. This dataset is publicly available at our project's Github repo. All the data provided is cleaned by removing blank lines, unknown characters and digits.

3.2 Supervised MT

3.2.1 Sanskrit-English parallel data

For implementing the supervised machine translation methods, we require parallel data. Since, Sanskrit-Hindi parallel data is very hard to find, we used an intermediate language English. We will first translate from Sanskrit to English and then use Google Translate API to translate English sentences to Hindi. We have created 34k lines of parallel data by scraping ISTA and Gita-Supersite for Ramayan, Bible, Manu Smriti, Bhagvadgita and Rig Veda. We have taken 32k lines for training and the rest of the data has been equally divided for validation and testing. All of our data is publicly available at our project's Github repo.

4 Method & Implementation

4.1 Unsupervised MT

We followed the model proposed by (Lample et al., 2018b), with necessary changes to incorporate our own languages and datasets.

- We used the popular tool for Indian languages - indic-nlp library for tokenization of both Sanskrit and Hindi data.
- As a statistical model for machine translation, we used Moses(Koehn et al., 2007).
- For cross-lingual word embeddings, we used the pre-trained word embeddings provided by (Kumar et al., 2020)

Using the cross-lingual embeddings provided by (Kumar et al., 2020), we build an initial phrase table. We then generate smoothed n-gram language models for both Sanskrit and Hindi, using KenLM(Heafield, 2011). Using this language model and the phrase table obtained earlier, we obtain a rough translation of the source(Sanskrit) monolingual corpus into target(Hindi) language. This way we generate a roughly parallel data. Now, we train Moses statistical machine translation system in a supervised manner on this parallel data to learn a mapping from target to source language. Using this mapping, we generate source sentences from target sentences, and compare between the original source sentences and generated source sentences. The difference between these two constitutes our loss function. We can iterate over this process as many times as needed.

4.2 Supervised MT

4.2.1 Statistical MT

For doing Statistical Machine Translation, we implement Moses. We follow their official documentation for this task. The first step is the tokenization and cleaning of data. We limit the sentence length to 80 words, as very long sentences affect training Moses. This filters out around 300 sentences from our corpus. The Truecaser module then changes the case of all the sentences and then the parsed input is passed to the Language Model. Next step is training, in which we run GIZA(Gao and Vogel, 2008), which analyzes the parallel sentences and iteratively aligns words across the languages, forming a phrase table. This requires some tuning, as the weights are not yet optimized. This completes the training process, and then we test our model by converting Sanskrit sentences to English, followed by using Google Translate API to translate these English sentences to Hindi. Fig.1 provides the complete flow of the procedure followed in Moses.

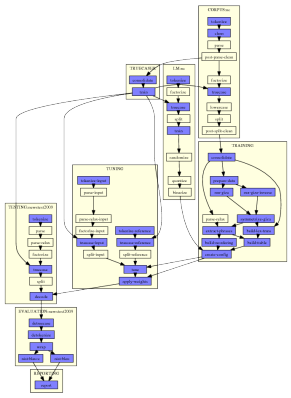

Figure 1: Statistical MT using Moses

4.2.2 Neural MT

For Neural Machine Translation, we implement the popular Neural model Seq2Seq(Sutskever et al., 2014) with a small modification of using attention network in the decoder. We normalize the text and filter long sentences. We then make list of tokens from sentences for both the languages. Seq2Seq is an Encoder-decoder network consisting of 2 RNNs - *encoder* and *decoder*. The encoder reads an input sequence and outputs a single vector, and the decoder reads that vector to produce an output sequence. We are using attention decoder because if only the context vector is passed between the encoder and decoder, that single vector carries the burden of encoding the entire sentence. Attention allows the decoder network to 'focus' on a different part of the encoder's outputs for every step of the decoder's own outputs. Fig.2 describes the work-flow of Seq2Seq model.

Challenges There are various challenges that we faced while doing Sanskrit-Hindi Machine Translation. Some of which are :-

- There is an inherent imbalance in the number of resources for Sanskrit and

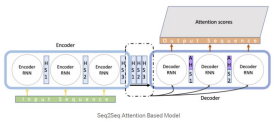

Figure 2: Seq2Seq model for Neural MT

Method	BLEU Score(%)
Unsupervised MT	0.05
Supervised Statistical MT	28.47
Supervised Neural MT	31.34

Table 1: BLEU Scores for different methods

Hindi, which is evident from the minimal number of pre-trained word embeddings - 40000 for Sanskrit as compared to 1.2 million for Hindi. Training cross-lingual embeddings required for Unsupervised MT is a very computationally heavy task, for which we didn't have sufficient GPU resources. So we had to rely on the pre-trained embeddings provided by (Kumar et al., 2020). Our data and the data used for this training are quite different from each other, which is one of the reasons for the poor performance of our Unsupervised MT model.

- Getting Sanskrit-Hindi parallel data for supervised learning was difficult, so we used an intermediate language English. Getting even Sanskrit-English parallel data was not easy, after days of searching, we were only able to get around 34k lines. For translating English to Hindi, we use Google Translate API, but it also has a limit on the number of requests it services. Moreover, cascading two machine translation systems one after the other further degrades the performance.

- We contacted personnel from JNU[1], Gitapress(Gorakhpur)[2], ISTA[3] to help us by providing the Hindi versions of ancient texts, but unfortunately, we did not receive any reply.

5 Results

We evaluate the performance of our model on two fronts - analytically based on BLEU score, and human evaluation. We get the following results based on BLEU scores:

[1]Rajneesh Pandey, Girish Nath Jha, authors (Pandey et al., 2018)

[2]Famous publishers of ancient sacred texts

[3]Sacred text archive containing Sanskrit and English translations of many religious texts

Analysis & Inference

5.1 Unsupervised MT

For Unsupervised MT system, we get very poor BLEU score of 0.05%. This denotes that the model is learning almost nothing. On human evaluation, we see that most of the words in the output are predicted as UNKNOWN. There are various reasons which might have led to such poor performance - the scarcity of pre-trained embeddings available, the misalignment between our training data, and the data which the embeddings were trained on.

5.2 Supervised Statistical MT

Here we obtain a BLEU score of 28.47%. This denotes reasonable performance, although it is lower as compared to (Pandey et al., 2018), where they got 57 BLEU score. Their model was trained on 43k parallel sentences, compared to our 32k. On human evaluation, we see that it matches some keywords in many sentences, although forming a complete sentence is rare. This model, however, shows the promise that given more data, it can achieve better results, as the model is learning many word alignments.

5.3 Supervised Neural MT

This gives a BLEU score of 31.34%. But on human evaluation, we see that it is starting the output sentence well, but after 3-4 words, it starts repeating the words. So even though the BLEU score suggests it is doing better than the statistical model, in reality, it is not learning to form a complete sentence. Before starting this model, we didn't expect it to perform very well, as it is an experimentally established fact that neural models require quite large amount of data, much more than the 32k parallel sentences we trained it on. This lack of data might have led our model to under-fit, so it only predicts a few words correctly, and predicts the same words again and again.

Here are some results which we obtained using these different methods. Here GT denotes Ground Truth Hindi translation, SMT denotes Statistical MT, NMT denotes Neural MT. (Not showing the output of Unsupervised MT as it contains mostly UNKNOWN token.)

- इन्दूं सोमासो अक्षरन|
  <Input Sanskrit Sentence>
  इन्द्रजाल को बहना। <GT>
  ताकतवर बल के साथ इंद्र ने सोमासो प्रवाहित किया है <SMT>
  इंद्रा और उसके साथ और और और और <NMT>
- प्रथमाद् ईश्वरः स्वीयानि सर्व्वकर्म्माणि जानाति ।
  <Input Sanskrit Sentence>
  भगवान के लिए जाना जाता है दुनिया की शुरुआत से अपने सभी काम कर रहे हैं। <GT>
  तुम उनके कामों से परमेश्वर को जानते हो। <SMT>
  और उसने कहा, भगवान, और कहा, भगवान, उसे, और, और, और, और <NMT>
- पर ते पूर्वाणि करणानि विप्राविद्वां आह विदुषे करांसि|
  <Input Sanskrit Sentence>
  बुद्धिमान व्यक्ति ओ ऋषि और संत शासक, वह आदमी जो सभी प्राचीन कारनामों को जानता है। <GT>
  मैं तेरा कारनामे घोषित करूंगा <SMT>
  वह और और और और के मार्च को घोषित करेगा <NMT>

6 Conclusion & Future Work

In this project, we tried to perform the relatively untouched area of machine translation - Sanskrit-Hindi translation. We followed different approaches and analyzed the pros and cons of these approaches in context of our data and resource availability. We had a lot of expectation from the unsupervised MT model owing to the amount of data we collected for it, but from the results, we can conclude that for this system to work, we need to ourselves train the cross-lingual embeddings, so that similar data distribution is maintained. For the supervised statistical MT model, again the amount of parallel data is a bottleneck, which, if overcome, has the potential to significantly improve the performance. For the neural model, we need even more amount of data, but once we have a large amount of data, we believe it should be able to out-perform statistical MT model.

Overall we can say that with the constraints on data and resources, statistical MT performs better than the other two methods. But there is a lot of scope of improvement for each of these methods. One of our major contributions is the unification of different data resources, along with scraping many of them ourselves, and organizing them at a single place. In addition, we created a reasonably large amount of monolingual Sanskrit dataset. These advances can help future work in this area of research. We aim to further improve the availability of data, and try other approaches like Zero-Shot translation, which has shown some promise for other language pairs. Owing to the availability of computational resources, we also aim to train our own cross-lingual embeddings, which can help improve the unsupervised MT model, which seems the most prominent way forward if additional parallel data is not created. The current methods could also be tried with various settings, tuning them to obtain better results.

References

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software engineering, testing, and quality assurance for natural language processing*, pages 49–57.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, page 177–180, USA. Association for Computational Linguistics.

Saurav Kumar, Saunack Kumar, Diptesh Kanojia, and Pushpak Bhattacharyya. 2020. "a passage to India": Pre-trained word embeddings for Indian languages. In *Proceedings of the 1st Joint*

Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), pages 352–357, Marseille, France. European Language Resources association.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The iit bombay english-hindi parallel corpus.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rajneesh Pandey, Atul Kr. Ojha, and Girish Nath Jha. 2018. Demo of sanskrit-hindi smt system.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.