Neural Machine Translation using Multihead self-attention for English-Tamil

Abstract

A huge amount of valuable resources is available on the web in English, which are often translated into local languages to facilitate knowledge sharing among local people who are not much familiar with English. However, translating such content manually is very tedious, costly, and time-consuming process. To this end, machine translation is an efficient approach to translate text without any human involvement. Neural machine translation (NMT) is one of the most recent and effective translation technique amongst all existing machine translation systems. In this paper, we apply NMT for English-Tamil language pair. We propose a novel neural machine translation technique using Multihead self-attention and word-embedding along with Byte-Pair-Encoding (BPE) to develop an efficient translation system that overcomes the OOV (Out Of Vocabulary) problem for languages which do not have much translations available online. We use the BLEU score for evaluating the system performance. Experimental results confirm that our proposed translator (9.26 BLEU score) outperforms Google translator (3.75 BLEU score).

1 Introduction

Big countries such as India and China have several languages which change by regions. For instance, India has 23 constitutionally recognized official languages (*e.g.*, Hindi, Tamil, and Panjabi) and several hundreds unofficial local languages. Despite Indian population is approximately 1.3 billion, only approximately 10% of them English speak English. Some studies say that out of these 10% English speakers only 2% can speak, write, and read English well, and rest 8% can merely understand simple English and speak broken English with an amazing variety of accents (sta). Considering a significant amount of valuable resources is available on the web in English and most people in India can not understand it well, it is essential to translate such content in to local languages to facilitate people. Sharing information between people is necessary not only for business purposes but also for sharing their feelings, opinions, and acts. To this end, translation plays an important role in minimizing the communication gap between different people. Considering the vast amount of information, it is not feasible to translate the content manually. Hence, it is essential to translate text from one language (say, English) to another language (say, Tamil) automatically. This process is also known as *machine translation*.

There are many challenges in machine translation for Indian languages. For instance, (i) the size of parallel corpora and (ii) differences amongst languages, mainly the morphological richness and word order differences due to syntactical divergence are two of the major challenges. Indian languages (IL) suffer both of these problems, especially when they are being translated from English. There are only a few parallel corpora for English and Indian languages. Moreover, Indian languages such as Tamil differ from English in word order as well as in morphological complexity. For instance, English has Subject-Verb-Object (SVO) whereas Tamil has Subject-Object-Verb (SOV). Moreover, English is a fusional whereas Tamil is agglutinative languages. While syntactic differences contribute to difficulties of translation models, morphological differences contribute to data sparsity. We attempt to address both issues in this paper.

Though much work is being done on machine translation for foreign and Indian languages but apart from foreign languages most of works on Indian languages are limited to conventional machine translation techniques. We observe that the techniques like word-embedding and Byte-pair-encoding (BPE) are not applied on many Indian

languages which have shown a great improvement in natural language processing. Thus, in this paper, we apply a neural machine translation technique (torch implementation) with word embedding and BPE. Especially, we work on English-Tamil language pair as it is one of the most difficult language pair (ZdenekŽabokrtskỳ, 2012) to translate due to morphologically richness of Tamil language. We obtain the data from EnTamv2.0 and Opus, and evaluate our result using widely used evaluation matric BLEU. Experimental results confirm that we got much better results than conventional machine translation techniques on Tamil language. We believe that our work can also be applied to other Indian language pairs too.

Main contributions of our work are as follows:

- This is the first work to apply BPE with word embedding on Indian language pair (English-Tamil) with Multihead self-attention technique.
- We achieve comparable accuracy with a simpler model in less training time rather then training on deep and complex neural network which requires much time to train.
- We have shown how and why data preprocessing is a crucial step in neural machine translation.
- Our model outperforms Google translator with margin of 5.51 BLEU score.

The rest of the paper is organized as follows. Sections 2 and 3 describe related work and the methodology of our translator, respectively. Evaluation is presented in Section 4. Finally, Section 5 concludes the paper.

2 Literature Survey

Several works have been reported on machine translation (MT) in last a few decades, earliest one in 1950s (Booth, 1955). There are various approaches adopted by researchers such as rule-based MT (Ghosh et al., 2014; Wong et al., 2006), corpus-based MT (Wong et al., 2006), and hybrid-based MT (Salunkhe et al., 2016). Each of these approaches has its own pros and cons. Rule-based machine translation systems traverse the source text to produce an intermediate representation of the text, and depending on the representation this approach is further classified into transfer-based approach (TBA)(Shilon, 2011) and inter-lingua based approach (IBA).[1]

Corpus-based approach uses a large sized parallel corpora in the form of raw data. This raw data contains text with their respective translations. These corpora are used to acquire knowledge for translation. A corpus-based approach divides itself into two sub types: (i) statistical machine translation (SMT) and (ii) example-based machine translation (EBMT) (Somers, 2003). SMT[2] generates its translation on the basis of statistical models. It depends on the combination of language model as well as translation model with a decoding algorithm. EBMT on the other hand uses the existing translation examples for generating a new translation. This is done by finding out the examples matching with the input. Then alignment is performed to find out the parts of translation that can be reused. Hybrid-base machine translation is a combination of transfer approach and any corpus-based approaches in order to overcome their limitations.

Recent research (Khan et al., 2017) suggest that the machine translation performance of Indian language pairs (*e.g.*, Hindi, Bengali, Tamil, Punjabi, Gujarati, and Urdu) is of an average of 10% accuracy. This necessitates the need of building better machine translation systems for Indian languages.

NMT is novel and emerging technique for various languages and shown remarkable results (Hans and Milton, 2016). In this paper phrase-based hierarchical models trained after morphological preprocessing using NMT. Patel *et al.* (Patel et al., 2017) trained their model after suffix separation and compound splitting. Different models were also tried for the same task and achieved a good result on their respective dataset (Pathak and Pakray). We analyze that morphological preprocessing, suffix separation, and compound splitting can be overpass by using Byte-Pair-Encoding and produced similar or better translation without making the model complex.

3 Methodology

In this study, we present a neural machine translation technique using Multihead self-attention and word-embedding along with Byte-Pair-Encoding (BPE) to develop an efficient translation system,

[1] https://en.wikipedia.org/wiki/Interlingual_machine_translation

[2] https://books.google.ch/books?id=4v_Cx1wIMLkC

that overcomes the OOV (Out Of Vocabulary) problem for languages which do not have much translations available online. Thus, first, we provide an overview of neural machine translation, self-attention model, word embedding, and Byte Pair Encoding. Next, we present the framework of our translator.

3.1 Neural Machine Translation Overview

Neural Machine translation is a technique that is based on neural networks and the conditional probability of translated sentence from the source language to target sentences (Revanuru et al., 2017). In the following sub-sections we will provide an overview of sequence to sequence architecture and self-attention model that are used in our proposed translator.

Sequence to Sequence Architecture Sequence to sequence architecture is basically used for response generation whereas in machine translation models it is used to find the relationship between two different language pairs. It consists of two parts, an encoder and a decoder. The encoder takes the input from source and the decoder generates the output based on encoding vector and previously generated words. Assume A be the source sentence and B be a target sentence. The encoder converts the source sentence $a_1, a_2, a_3..., a_n$ into vector of fixed dimensions and the decoder outputs word by word using conditional probability. Here, $A_1, A_2, ..., A_M$ in the equation are the fixed size encoded vectors. Using chain rule, the Eq. 1 is converted to the Eq. 2.

$$P(B/A) = P(B|A_1, A_2, A_3, ..., A_M) \quad (1)$$

$$P(B|A) = P(b_i|b_0, b_1, b_2, ..., b_{i-1}; \\ a_1, a_2, a_3, ..., a_m \quad (2)$$

While decoding, next word is predicted using previously predicted word vectors and source sentence vectors in Eq. 1. Each term in the distribution is represented with a softmax over all the words in the vocabulary.

Attention Model In a basic encoder-decoder architecture, encoder reads the whole sentence, memorizes it and store it in the final activation layer, then the decoder network generates the target translation. This architecture works quite well

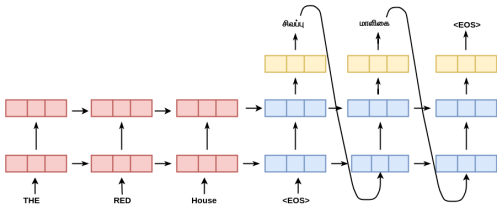

Figure 1: Seq2Seq architecture for English-Tamil

for short sentences, so we might achieve a relatively high BLEU score, but for very long sentences, maybe longer than 30 or 40 words, the performance degrades. To overcome this attention mechanisms were used. The basic idea is: each time the model predicts an output word, it only uses parts of an input where the most relevant information is concentrated instead of an entire sentence. In other words, it only pays attention to some input words. Many types of attention mechanisms are used to improvise the translation accuracy, but the multihead self-attention overcomes the most of the problems.

self-attention In self-attention architecture (reference)at every time step of an RNN, a weighted average of all the previous states will be used as an extra input to the function that computes the next state. With the self-attentive mechanism, the network can decide to attend to a state produced many time steps earlier. This means that the latest state does not need to store all the information. The mechanism also makes it easier for gradient to flow more easily to all previous states, which can help against the vanishing gradient problem.

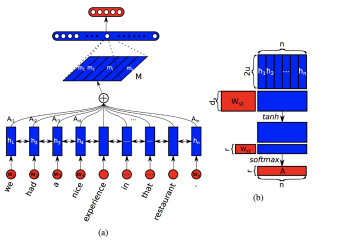

Figure 1: A sample model structure showing the sentence embedding model combined with a fully connected and softmax layer for sentiment analysis (a). The sentence embedding M is computed as multiple weighted sums of hidden states from a bidirectional LSTM ($h_1, ..., h_n$), where the summation weights ($A_{i1}, ..., A_{in}$) are computed in a way illustrated in (b). Blue colored shapes stand for hidden representations, and red colored shapes stand for weights, annotations, or input/output.

Figure 2: Attention model

Word Embedding Word embedding is a way of representing words on a vector space where the words having same meaning have similar vector representations. Each word from vocabulary is represented in hundreds of dimensions. Normally

pre-trained word embeddings are used and with the help of transfer learning words from vocabulary are converted to vector (Cho et al., 2014). In our model, we used FastText word vectors[3] to convert English and Tamil vocabulary into a 300-dimensional vector. Training the model with same layers, optimization method, attention, and regularization we got a BLEU score of Point 6.74.

Byte Pair Encoding BPE (Gage, 1994) is a simple data compression technique. It replaces most frequent pair bytes in a sequence with single unused byte. We use this algorithm for word segmentation. By merging frequent pairs of bytes we merge charters or character sequences (Sennrich et al., 2015). NMT symbols interpretative as sub-words units and networks can translate and make the new word on the basis of sub-words. We learned the independent encodings on our source and target training data with 10,000 and 25,000 words and then apply it on train test and validation data for both source and target. BPE helped in compound splitting and suffix, prefix separation which is used for creating new words of Tamil language. we used BPE along with word embeddings and tried different models.

3.2 Translator

We tried various models to get a better intuition on how parameter tuning along with different techniques affects on Indian language pair. Our first model architecture consists of 2 layer Bi-directional LSTM encoder and 2 layers LSTM decoder of 500 dimensions each with the vocabulary size of 50,004 words for both source and target. First we tried SGD optimization method, Luong attention with a dropout (regularization) of 0.3, and learning rate 1.0. Secondly, we changed the optimization method to Adam and attention to Bahdanau with the learning rate of 0.001. We got our best results with a BPE vocabulary size of 25,000 with 2 Layer Bi-directional encoder-decoder, Adam optimization with a learning rate of 0.001, Bahdanau attention, and word-embedding with the dimension of 500. Finally we used multihead self-attention mechanism with with same Byte-Pair-Encoded vocabulary size, 8 heads,6 bi-LSTM layers 0.1 dropout and other standard parameters. We got the final result after training 100000 steps on GPU().

4 Evaluation

4.1 Evaluation Metric

The BLEU score or bilingual evaluation under study is a method to measure the difference between machine and human translations (Papineni et al., 2002). The approach works by counting and matching n-grams in result translation to n-grams in the reference text, where unigram would be each token and a bigram comparison would be each word pair and so on. The comparison is made regardless of word order. This method is a modification of a simple precision method.

4.2 Dataset

We used the datasets obtained from EnTam V2.0[4] and Opus.[5] The sentences are taken from various domains like news, bible, cinema, movie subtitles and combined to build our final parallel dataset. After preprocessing and splitting it to train, test, and validation, our final dataset contains 1,83,451 training corpus, 1,000 validation and 2,000 test corpus from English to Tamil. The data used is encoded in UTF-8 format.

4.3 Data Pre-processing

Research works (Hans and Milton, 2016; Ramesh and Sankaranarayanan, 2018) suggest that they have used EnTamV2.0 in their experiments. However, we find that in both well-known parallel corpus for English-Tamil datasets (*i.e.*, EnTam V2.0 and Opus) have many repeated sentences, which outcomes the wrong results (may be high or low) after dividing into train, test, and validation sets, as some of the sentences occur both in train and test sets. Thus, it is essential to clean, analyses, and correct before using for experiment. We observed the following four main problems in the online available corpus for English-Tamil dataset.

- Repetition of sentences with same source and same target .
- Sentences with same source and different translation.
- Sentences with different source and same translation.

[3] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[4] http://ufal.mff.cuni.cz/~ramasamy/parallel/html/

[5] http://opus.nlpl.eu/

- Tokenization of Indian languages.

To overcome the first problem we took unique pairs from all sentences and removed repeating ones. We completely removed those sentences which are repeated more than twice because in the second case we cannot identify that which translated sentence is correct for the same source and which source is correct for the same translation in the third case. We observed that there are some sentences which are repeating even more than 10 times in Opus dataset. This confuses the model to learn and identify different new features, overfits the model, and led to the wrong results. This preprocessing is required as it may be possible that train and test contain the same sentences which let to the better prediction for test set but wrong predictions for new sentences.

The second important thing which we observed that there are many tools available for tokenization of English language (*e.g.*, Perl tokenizer) but does not work well for the Tamil language, because there are different morphological symbols which used in word formation of Tamil language which are removed by these tokenization tools in Indian languages (Tamil in our case). Without tokenization model consider *word*, *word,* and *word!* as three different words in the vocabulary of Tamil. We tokenize the Tamil language sentences using our own code before training. This problem can also be overcome by Byte-pair-Encoding.

Finally after working on all these small but effective preprocessing such as removing sentences with the length greater than 50, removing non translated words in target sentences, removing noisy translations and unwanted punctuations, we got our final dataset[6] of 1,86,451 parallel sentences which was cleared from 2,23,685 sentences. It is divided into training (1,83,451 sentences) testing (2,000 sentences) and validation (1,000 sentences) respectively after shuffling.

4.4 Result

We used Google translate API in python to translate the English sentences and compared Google results with our various models. It is also observed that the translations below are handy enough to use in day to day life as well as official works. From test results, we can also deduce that our model overcomes the OOV (Out of Vocabulary) problem in some cases.

[6] https://github.com/himanshudce/MIDAS-NMT-English-Tamil

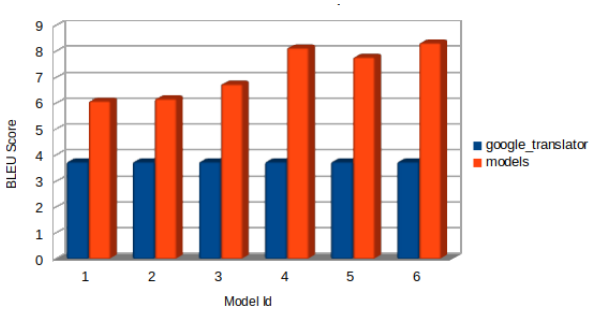

Figure 3: Different model comparison with Google Translator Table1.

Model	BLEU
Google Translator	3.75
Bi-L+S+Lu	6.10
Bi-L+A+B	6.18
Bi-L+A+B+E	6.74
Bi-L(4-Layer)+A+B +BPE(10000)+E	7.78
Bi-L+A+B +BPE(10000)+E	8.14
Bi-L+A+B+BPE(25000)+E	8.33
Bi-L+self-A+BPE(25000)+E	9.26

Table 1: BLEU Score of English-Tamil translated system. Symbols have the following meanings: Bi-L= Bi-LSTM, S= SGD(Wu et al., 2016), L= LSTM, A=Adam(Vaswani et al., 2017), B= Bahdanau (Bahdanau et al., 2014), E=Word Embedding, Lu=Luong(Luong et al., 2015))

4.5 Analysis

We conducted an anonymous survey of ten random sentences from test data and accumulated reviews of Tamil speaking people on that. After comparing accumulated reviews of Google translator and our translator, it was discovered that translations from our MIDAS translator are selected as better translations in 71.66% cases than translations of Google translator. Moreover, two out of ten translations from MIDAS translator are unanimously selected by respondents in compared to only one translation by Google translator.

5 Conclusion & Future Work

In this paper, we applied NMT to one of the most difficult language pairs (English-Tamil). We showed that NMT with pre-trained word embedding and Byte Pair Encoding performs better

than complex translation techniques on Indian languages. Our model outperformed Google translator with a margin of 4.58 BLEU points. Since We achieved fairly good accuracy so our model can be used for creating English-Tamil translation applications that will be useful in domains such as tourism and education. Moreover, We can explore the possibility of using above techniques for various English Indian language translation. In future, we would also like to employ machine translation in detecting offensive languages from code-switched languages too (Mathur et al., 2018).

References

What percentage of people in india speak english? https://tinyurl.com/indianlanguageStats. Accessed: 2018-08-27.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Andrew Donald Booth. 1955. Machine translation of languages, fourteen essays.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Siddhartha Ghosh, Sujata Thamke, et al. 2014. Translation of telugu-marathi and vice-versa using rule based machine translation. *arXiv preprint arXiv:1406.3969*.

Krupakar Hans and RS Milton. 2016. Improving the performance of neural machine translation involving morphologically rich languages. *arXiv preprint arXiv:1612.02482*.

Nadeem Jadoon Khan, Waqas Anwar, and Nadir Durrani. 2017. Machine translation approaches and survey for indian languages. *arXiv preprint arXiv:1701.04290*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the International Workshop on Natural Language Processing for Social Media*, pages 18–26.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Raj Nath Patel, Prakash B Pimpale, et al. 2017. Mtil17: English to indian langauge statistical machine translation. *arXiv preprint arXiv:1708.07950*.

Amarnath Pathak and Partha Pakray. Neural machine translation for indian languages. *Journal of Intelligent Systems*.

Sree Harsha Ramesh and Krishna Prasad Sankaranarayanan. 2018. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–119.

Karthik Revanuru, Kaushik Turlapaty, and Shrisha Rao. 2017. Neural machine translation of indian languages. In *Proceedings of the 10th Annual ACM India Compute Conference on ZZZ*, pages 11–20. ACM.

Pramod Salunkhe, Aniket D Kadam, Shashank Joshi, Shuhas Patil, Devendrasingh Thakore, and Shrikant Jadhav. 2016. Hybrid machine translation for english to marathi: A research evaluation in machine translation:(hybrid translator). In *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*, pages 924–931. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Reshef Shilon. 2011. *Transfer-based Machine Translation between morphologically-rich and resource-poor languages: The case of Hebrew and Arabic*. Ph.D. thesis, Citeseer.

Harold Somers. 2003. An overview of ebmt. In *Recent advances in example-bas ed machine translation*, pages 3–57. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Fai Wong, Mingchui Dong, and Dongcheng Hu. 2006. Machine translation using constraint-based synchronous grammar. *Tsinghua Science and Technology*, 11(3):295–306.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between

human and machine translation. *arXiv preprint arXiv:1609.08144*.

LoganathanRamasamy OndrejBojar ZdenekŽabokrtskỳ. 2012. Morphological processing for english-tamil statistical machine translation. In *24th International Conference on Computational Linguistics*, page 113.