NMT based Tamil Translation

Aditya Kumar Pathak [1], Himanshu Choudhary [2], Rajiv Ratn Shah [3]

[1] IIIT Bhubaneswar, India
[2] DTU Delhi , India
[3] IIIT Delhi, India
adityapathak.cse@gmail.com,
himanshuchoudhary_bt2k16@dtu.ac.in,
rajivratn@iiitd.ac.in

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 and Hindi-Tamil language pair. We propose a novel neural machine translation technique using pre-trained subword embeddings (BPEmb) 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. We got the 26.97% Precision,37.98% Recall for English to Tamil and 25.18% Precision and 27.24% Recall for Tamil to Hindi respectively.

Keywords: Neural Networks · Machine Translation · Attention mechanism · Word Embeddings · Byte-Pair-Encodding · BPEmb

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 [1]. 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. Since both the languages (Tamil and Hindi) are known for its morphologically richness, that makes the task much more difficult to work.

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 along with 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 pre-trained subword embeddings [3]. Especially, we work on English-Tamil and Tamil-Hindi language pair as these are the most difficult language pair [2] to translate due to morphologically richness of Tamil and Hindi language. We obtain the data from VPT-IL and evaluate using precision and recall we believe that we got better results than conventional machine translation techniques on Hindi and Tamil language. 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 along with pre trained word embedding on Indian language pair English-Tamil and Hindi-Tamil with NMT 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.

2 Methodology

In this study, we present a neural machine translation technique using Byte-Pair-Encoding along with pre trained word embedding to develop an efficient translation system, called MIDAS translator 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, attention model, word embedding, and Byte Pair Encoding. Next, we present the framework of our MIDAS translator.

2.1 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 [4]. In the following sub-sections we will provide an overview of sequence to sequence architecture and attention model that are used in our proposed MIDAS 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.

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

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

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.

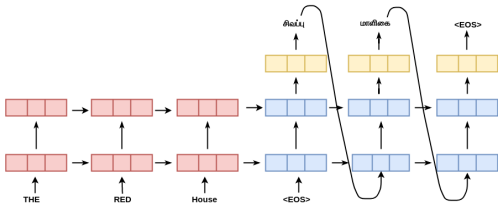

Fig. 1. Seq2Seq architecture for English-Tamil

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 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. Using attention[1] mechanism with a basic encoder-decoder architecture is a solution for that. It translates similar to humans by looking at part of the sentence at a time. The mechanism decides how much attention should be paid to a particular word while translating the sentence. The mechanism is shown in Fig. 2. The Encoder generates the attention vectors $h_1, h_2, h_3.....h_t$ from the inputs $A_1, A_2, A_3 A_t$. Then, context vector C_i is calculated using concatenation of these vector for each output time step. Then Using the context vector C_i hidden state S_i and previously predicted words, decoder generates the softmax output B_i.

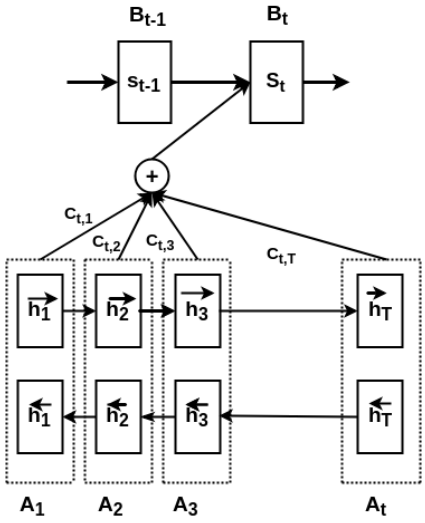

Fig. 2. Attention model

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 [5]. In our model, we also tried

[1] https://hackernoon.com/attention-mechanism

FastText word vectors[2] to convert English, Tamil and Hindi vocabulary into a 300-dimensional vector.

Byte Pair Encoding BPE [6] 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 [7]. NMT symbols interpretative as subwords units and networks can translate and make the new word on the basis of sub-words.

BPEmb Subword embeddings is collection of both Byte-Pair-encodding and Word embeddings.In our model we used pre-trained subword embeddings which was trained on Wikipedia. Subwords allow guessing the meaning of unknown / out-of-vocabulary words and Byte-Pair Encoding gives a subword segmentation that is often good enough, without requiring tokenization or morphological analysis. We used BPEmb with 2000 merge operations and 50 dimension vectors 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 English Tamil and Hindi language.

2.2 MIDAS 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 50 dimensions 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 2000 with 2 Layer Bi-directional encoder-decoder, Adam optimization with a learning rate of 0.001, Bahdanau attention, and BPEmb with the dimension of 50. We used GPU (Nvidia GeForce GTX 1080) for the training of different models which increase the computation speed.

3 Evaluation

3.1 Evaluation Metric

The BLEU score or bilingual evaluation under study is a method to measure the difference between machine and human translations 8. 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.

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

3.2 Precision and Recall

In pattern recognition, information retrieval and binary classification, precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance.

3.3 Dataset

We used the datasets obtained from VPT-IL. After preprocessing and splitting it to train and validation, our final dataset contains 1223 English-Tamil training corpus and 200 validation corpus. For Hindi to tamil we used 1772 training corpus and 200 validation corpus as provided by the VPT-IL organization. The data used is encoded in UTF-8 format.

3.4 Results

We got the following results after training on the 1223 English-Tamil training corpus with 200 validation corpus which were splitted from 1443 corpus and 1772 Hindi to tamil training corpus and 200 validation corpus which were splitted from 1992 corpus as provided by the VPT-IL Organization.

Model	% Precision	% Recall
English-Tamil	26.97	37.98
Tamil-Hindi	25.18	27.24

Table 1. Test Score for Our Model

4 Related Work

The similar work is given in the paper 9. In this paper the Phrase-based and Hierarchical models trained after morphological preprocessing using neural machine translation. In paper 10 model is trained after suffix separation and Compound Splitting. Different models were also tried for the same task and achieved a good result on their respective dataset as given in 11. We analyze that morphological preprocessing, suffix separation and compound splitting can be overpass by using Byte-Pair-Encoding and produce similar or even better translation without making the model complex.

5 Conclusion & Future Work

In this paper, we applied NMT to two most difficult language pairs English-Tamil and Tamil-Hindi. Since amount of data highly affects the neural network models, hence we got the comparable results. We believe that our model will perform much better on larger datasets and pre-trained subword embeddings will performs better than complex translation techniques on Indian languages. 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 12.

References

1. What percentage of people in India speak English ? https://tinyurl.com/indianlanguageStats
2. Zdenek abokrtsk, LoganathanRamasamy OndrejBojar. Morphological processing for English-Tamil statistical machine translation.(2012): 113.
3. Benjamin Heinzerling, and Michael Strube. Toward the next generation of recommender systems: BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. Proceedings of the Eleventh International Conference on Language Resources and Evaluation,2018.
4. Revanuru, Karthik and Turlapaty, Kaushik and Rao, Shrishas: Neural Machine Translation of Indian Languages.Proceedings of the 10th Annual ACM India Compute Conference on ZZZ, 2017, pp.11-20.
5. Cho, Kyunghyun and Van Merriënboer, Bart and Bahdanau, Dzmitry and Bengio, Yoshua, 2014, December. On the properties of neural machine translation: Encoder-decoder approaches: arXiv preprint arXiv:1409.1259 technology.
6. Gage, Philip., 1994. A new algorithm for data compression. The C Users Journal, p.23-38.
7. Sennrich, Rico and Haddow, Barry and Birch, Alexandra.,Neural machine translation of rare words with subword units:, arXiv preprint arXiv:1508.07909,2015.
8. Papineni, Kishore and Roukos, Salim and Ward, Todd and Zhu, Wei-Jing,2002. BLEU: a method for automatic evaluation of machine translation:, Proceedings of the 40th annual meeting on association for computational linguistics,pp.311-318.
9. Hans, Krupakar and Milton, RS, 2016, Neural Machine Translation for Low Resource Languages using Bilingual Lexicon Induced from Comparable Corpora. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop (pp. 112-1197).
10. Patel, Raj Nath and Pimpale, Prakash B and others, 2017, . MTIL17: English to Indian Langauge Statistical Machine Translation. arXiv preprint arXiv:1708.07950.
11. Pathak, Amarnath and Pakray, Partha., 2017.: Neural Machine Translation for Indian Languages. Journal of Intelligent Systems.
12. Mathur, Puneet and Shah, Rajiv and Sawhney, Ramit and Mahata, Debanjan., 2018. Detecting Offensive Tweets in Hindi-English Code-Switched Language. Proceedings of the International Workshop on Natural Language Processing for Social Media, pp.18-26.