Neural Machine Translation : Malayalam to Telugu

Pranav Deep
M.Tech - AIML
IIIT Sri City
Andhra Pradesh, India
Email: pranavdeep.i@iiits.in

Dr Himangsu Sarma
Asst. Professor (CSE)
IIIT Sri City
Andhra Pradesh, India
Email: himangshu.sarma@iiits.in

***Abstract*—Machine Translation is the task of converting text of one language to text of target language. Text can be in the form of words or sequences. While word based translation is relatively a simpler task, sequence to sequence translations are not easy for computers to understand. The use of Neural Networks to solve Machine Translation tasks has seen a significant boost in the performance of sequence to sequence models. This article presents a state of the art approach called "Transformers" model to solve this task.**

1. Introduction

The need for Machine Translation has existed since the inception of modern computer technology. However, only rule based approaches were used to perform translation in 19Th century. Classical rule based models were replaced by statistical models in 20Th. However, both these methods did not achieve substantial results in machine translation. The rise of neural networks outperformed both statistical and rule based methods. However, Neural network models are data hungry. These models need aligned parallel corpus that's difficult to procure.

Koehn (2005) [1] used European parliament proceedings to create evolving parallel corpora. This has later been incorporated in a variety of Machine Translation approaches. Shashank(2020) [2] provided a similar approach by using Prime Minister Narendra Modi's various speeches at his public rallies, Mann ki Baat etc. Shashank(2020)[2] provided a multilingual corpora of various Indic languages.

1.1. Motivation

Peter Norvig's [3] famous quote "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 the challenge of automatic machine translation difficult, perhaps one of the most difficult in artificial intelligence" emphasizes both the need and the intricacies of Machine translation.

To add, most of the Indic languages are categorized as "low-resourced languages" considering the resources for parallel corpora. This article attempts to work on 2 such low resourced languages, by using semi-processed Malayalam corpus obtained from PM's Mann ki Baat.

2. State of the art/Background

Shashank(2020) [2] presented sentence aligned parallel corpora across 10 Indian Languages - Hindi, Telugu, Tamil, Malayalam, Gujarati, Urdu, Bengali, Oriya, Marathi, Punjabi, and English - many of which are categorized as low resource. Along with a test corpus, they reported the methods of constructing such corpora using tools enabled by recent advances in machine translation and cross-lingual retrieval using deep neural network based methods.

In [4], Anoop presented the IIT Bombay English-Hindi Parallel Corpus. The corpus is a compilation of parallel corpora previously available in the public domain as well as new parallel corpora we collected. The corpus contains 1.49 million parallel segments, of which 694k segments were not previously available in the public domain. The corpus was pre-processed for machine translation, and they reported baseline phrase-based SMT and NMT translation results on their corpus. They achieved a BLEU score of 12.23 for NMT (English - Hindi) and 12.83 for Hindi to English.

In [5],Karthik explored Neural Machine Translation that has led to remarkable improvements compared to rule-based and statistical machine translation techniques, by overcoming many of the weaknesses in the conventional techniques. They applied NMT techniques to create a system with multiple models and then applied it for six Indian language pairs. They demonstrated that they outperformed Google Translate on their test data set by a margin of 17 BLEU points on Urdu-Hindi, 29 BLEU points on Punjabi-Hindi, and 30 BLEU points on Gujarati-Hindi translations.

In [6], Miguel investigated disabling label smoothing for the target-to-source model and sampling from a restricted search space issues of sampling based methods by Generalizing Back-Translation in Neural Machine Translation. Back translation is data augmentation by translating target monolingual data. They reformulated back translation in the scope of cross-entropy optimization of a neural machine translation model. They achieved BLEU score of 40 in one of their controlled scenario tests.

The very famous concept "Attention is all you need" presented by Ashish [7] turned out to be a game changer in NLP and NLU. The dominant sequence transduction models are based on complex RNN's or CNN's in an encoder-decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. They proposed a simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Their model achieved 28.4 BLEU on an English to German translation task beating the then state of the art by 2 BLEU. Their model achieved better performance with lesser training time. It has found to achieve very good generalization as well.

In 2019, Devlin proposed a new model : BERT Bidirectional Encoder Representations from Transformers [8] - . BERT was designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task specific architecture modifications. BERT obtained spectacular results on eleven natural language processing tasks, by getting 7.7% improvement on GLUE score to 80.5. MultiNLI accuracy to 86.7% (4.6% absolute improvement), It achieved significant improvements in several other NLU/NLP tasks.

Jinhua modified the BERT [9] model by using BERT as contextual embedding which fared better than using it for fine-tuning. The proposed new algorithm BERT-fused algorithm first uses BERT to extract representations for an input sequence and then the representations are fused with each layer of the encoder and decoder of the NMT model through attention mechanisms. They performed their experiments on English→German translation dataset and achieved Leveraging output of BERT as embedding BLEU score of 29.67. BERT-fused model outperformed transformers [7] by 3 BLEU score on an average(Testd on various translation datasets).

3. Proposed System

The proposed system incorporates use of a parallel corpus of Malayam-Telugu prepared by Sashank(2020)[2]. The implementation incorporates : Tokenization, Word Embeddings, Manual data pre-processing, Neural Network Model.

3.1. Tokenization

Tokenization splits the sentence into tokens(meaningful word). This is done with the help inLTK library [10]. They created subword vocabulary for each one of the languages by training a Sentence piece tokenization model on Wikipedia articles data set, using unigram segmentation algorithm (Kudo and Richardson, 2018 [11]). An important property of Sentence piece tokenization, necessary to obtain a valid subword-based language model, is its reversibility. Both Malayalam and Telugu sentences are tokenized using the above mentioned tokenization method.

3.2. Word Embedding

3.2.1. iNLTK Sentence Embedding. They trained FastText word embeddings for each language using IndicCorp data set, and evaluated the quality on: (a) word similarity, (b) word analogy, (c) text classification, (d) bilingual lexicon induction tasks. They compare those embeddings with two pre-trained embeddings released by the FastText project trained on Wikipedia (FT-W) and Wiki+CommonCrawl (FT-WC) respectively. They achieved 92.83% accuracy on this data set for Malayalam, 99.17% accuracy for Telugu and an average of 97% accuracy for all Indic languages [12].

iNLTK provides a simple API to generate these embeddings for a given corpus.

3.2.2. BERT Embedding. In [13], Matthew introduced a new type of deep contextualized word representation that models both complex characteristics of word use (e.g., syntax and semantics), and how these uses vary across linguistic contexts (i.e., to model polysemy). These word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. Their promising results show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP problems, including question answering, textual entailment and sentiment analysis.

ELMo: Embeddings from Language Models. ELMo word representations are functions of the entire input sentence. They are computed on top of two-layer biLMs with character convolutions as a linear function of the internal network states. This setup allowed a semi-supervised learning, where the biLM is pretrained at a large scale and easily incorporated into a wide range of existing neural NLP architectures.

Using these word embeddings, several simlilarity taks can be performed. Eg : BLEU score, Cosine Similarity, BERT similarity etc. To perform the above tasks, BERT embeddings are proposed.

3.3. Neural Network Model

Ashish(2017) presented a neural network architecture with encoder maps an input sequence of symbol to a sequence of continuous representations. Given a sequence of continuous representations, the decoder then generates an output sequence of symbols one element at a time. At each step the model is auto-regressive, consuming the previously generated symbols as additional input when generating the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder[7].

3.3.1. Encoder. The encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position wise fully connected feed-forward network. Then, a residual connection around each of the two sub-layers, followed by layer normalization is incorporated [7].

3.3.2. Decoder. The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack [7].

3.3.3. Attention. An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key [7].

4. Implementation

Major steps of implementation are : (a) Data set Preparation (b) Model building (c) Training & Testing

4.1. Data Preparation

The proposed transformer model needs a sentence aligned parallel corpora. The data was procured from IIIT - Hyderabad Indic Languages [2].

4.1.1. Sentence Alignment. Empirical Analysis : Procured datasets are then tested for sentence alignment manually for a 25 samples alternatively in every 100 samples. The corpus was found to have a good sentence alignment.

4.1.2. Statistics of the dataset. Total corpus size = 10,480
Train samples = 6707
Validation samples = 1677
Test samples = 2096

4.1.3. Data Pre-processing. Tokenization mentioned in Section 3.1 was performed on the entire corpus. After tokenization, lot of symbols like #, -, !, @ etc. were added to the corpus. There was a lot of noise in the data that went unnoticed in the empirical analysis. Some were manually removed, some where removed with a simple python script.

4.2. Model

Framework Used : ONMT Model based on PyTorch,
Model Name(Both encoder-decoder type) : Transformers,
Encoder-Decoder layers(No.) : 6,
Heads : 8,
Word to vec size : 512,

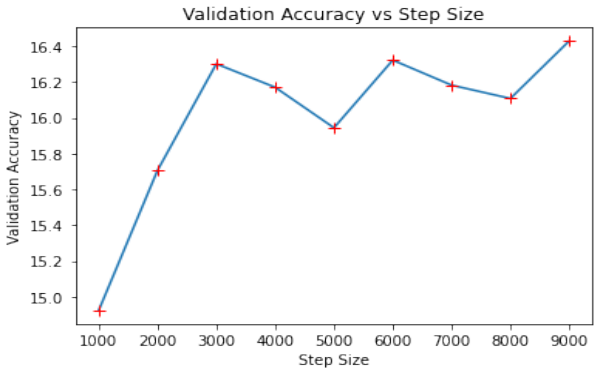

Figure 1. Accuracy vs Step size

rnn size : 512,
No. of epochs : 9000,
Dropout : 0.1,
Attention dropout : 0.1,
Optimizer : Adam's,
Learning Rate : 1.000005,
Batch Size : 2048,
Batch Type : tokens
Normalization type : tokens

5. Results

No. of epochs : 3000, Validation Accuracy : 16.3% , Validation Perplexity : 8088.5
No. of epochs : 9000, Validation Accuracy : 16.4% , Validation Perplexity : 8273.3
The Validation Accuracy versus Validation Perplexity plot can be seen in Fig. 2
The Validation Accuracy versus Step size plot can be seen in Fig. 1
The Validation Perplexity versus Step size plot can be seen in Fig. 3

5.1. Similarity Check

The following similarities were tested for predicted Telugu corpus and original telugu corpus. BERT embeddings were used for both the corpora.
BERT Score (Average) : 0.128
BLEU Score (Average) : 0.0203

6. Conclusion & Future Work

Very little accuracy has been achieved on the given data set. Model training took about 4 hours of time. However, not much performance improvement were obtained after 3000 epochs. Data had a lot of redundant elements which the model was not able to learn. That's because of inadequate pre-processing. Data can be further cleaned up.

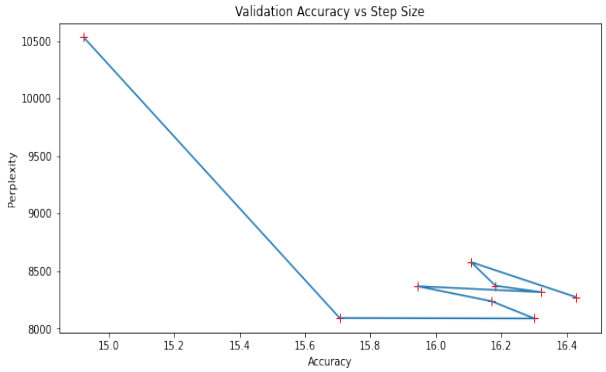

Figure 2. Accuracy vs Perplexity

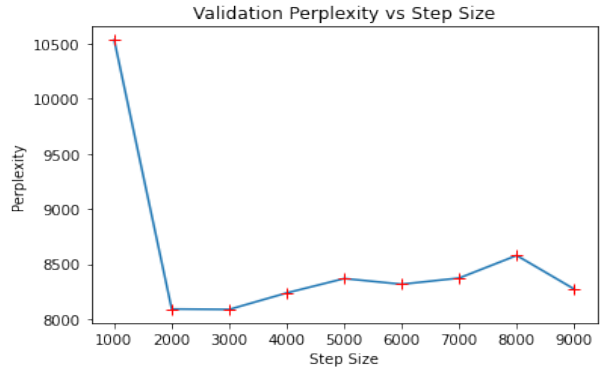

Figure 3. Perplexity vs Step size

6.1. Scope for Improvement

Lot more data can be scraped from PM's Mann ki Baat corpus directly. Data should be checked for appropriate sentence alignment. Corupus size should be increased to at least 100k manually translated sentences. Hyper-tuning the model parameters. BERT models can be used and various word embeddings techniques elaborated in section 3 can be applied to test. Testing time can be brought down by using Bi-directional approaches like BERT or Bi-LSTM etc.

References

[1] G. Cooper, H. Park, Z. Nasr, L. Thong, and R. Johnson, "Using virtual reality in the classroom: preservice teachers' perceptions of its use as a teaching and learning tool," *Educational Media International*, vol. 56, no. 1, pp. 1–13, 2019.

[2] S. Siripragada, J. Philip, V. P. Namboodiri, and C. V. Jawahar, "A multilingual parallel corpora collection effort for indian languages," *CoRR*, vol. abs/2007.07691, 2020. [Online]. Available: https://arxiv.org/abs/2007.07691

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[4] A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, "The iit bombay english-hindi parallel corpus," *ArXiv*, vol. abs/1710.02855, 2018.

[5] K. Revanuru, K. Turlapaty, and S. Rao, "Neural machine translation of indian languages," in *Proceedings of the 10th Annual ACM India Compute Conference*, ser. Compute '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 11–20. [Online]. Available: https://doi.org/10.1145/3140107.3140111

[6] M. Graça, Y. Kim, J. Schamper, S. Khadivi, and H. Ney, "Generalizing back-translation in neural machine translation," 2019.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[9] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T. Liu, "Incorporating bert into neural machine translation," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=Hyl7ygStwB

[10] G. Arora, "inltk: Natural language toolkit for indic languages," *CoRR*, vol. abs/2009.12534, 2020. [Online]. Available: https://arxiv.org/abs/2009.12534

[11] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *CoRR*, vol. abs/1808.06226, 2018. [Online]. Available: http://arxiv.org/abs/1808.06226

[12] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. Khapra, and P. Kumar, "Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages," 01 2020, pp. 4948–4961.

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365