English to Kannada Translation Model (seq2seq)

Shree Akshaya A T
CSE Dept, PES University
Bangalore, India
shreeakshaya2000@gmail.com

Shruthi Shankaran
CSE Dept, PES University
Bangalore, India
shruthishankaran10@gmail.com

H M Thrupthi
CSE Dept, PES University
Bangalore, India
thrupthijyo22@gmail.com

Abstract—**This paper presents an English to Kannada sequence-to-sequence translation model to translate sentences in English to Kannada. For this, we first construct an English to Kannada parallel corpus collected from various sources, pre-process it and feed the data into a sequence to sequence model that consists of an encoder, decoder and attention layer. After training the model, we use it to translate a given English sentence to Kannada. We evaluate the model using the BLEU score. The average BLEU score obtained was 0.76.**

Keywords—sequence-to-sequence, translation, encoder, decoder, attention, natural language processing

I. Introduction

Kannada is one of the Dravidian Languages which is predominantly spoken in Karnataka which is in the southwestern part of India. This language is being spoken by 43 million native speakers. The language has taken the spot of 25th most spoken language all over the world . Most of the languages like Tulu, Kodava are derived from this language.

Today a lot of native speakers prefer to have the information of anything in their native language which makes it more convenient and understandable. Most of the information around us is mostly is English which is not understandable by most of the common people. So in order to make it convenient machine translation models are used.

Various efforts have been made to develop machine translation that can be used practically. The methods used for translation for English to kannada which are on rule-based and corpus-based methods. But there are a lot of advancements in the field of translation like Neural machine translation ,Transformer based NMT(like BERT,ELMO) all these technologies have been advanced for translation.

Neural machine Translation (NMT) is the method that uses neural network models to learn a statistical model for translation. The advantage of this model is to train the given input sentence directly instead using a pipeline version.

In this paper we have used Sequence to sequence (seq2seq) networks along with attention to translate the given source language which is in English to kannada. Sequence to sequence model is a class of Recurrent Neural Network (RNN) in which the model is designed to take sequence of text as input and then output sequence of text. This kind of neural networks accept any size of input/output, need not have fixed size. It also uses the previous data in order to make predictions.

II. RELATED WORK

The paper **Machine Translation for English to Kannada**[1] which was published in 2011 by Mallamma V Reddy, Dr. M. Hanumanthappa deals with the Machine translation process for English to Kannada translation. In the above paper they have explained concepts and algorithms in the implementation of the bilingual translation for Eng to Kan, which translates a given sentence which is in English into Kannada by hybrid methodology(Rule based method + example based methods). The CLIR tool is built by using the ASP.NET and database kannada is also encrypted. The following approach is designed to produce a system by using the 4 patterns as a template: morphological analysis, pattern mapping, looking up in a bilingual dictionary and checking for possible combinations.

In the paper **POS Tagger for Kannada Sentence Translation**[2] which was published in 2012 by Mallamma V Reddy, Dr. M. Hanumanthappa.

The above mentioned paper lightens the processes like preprocessing, translation processes from eng-kan after that obtaining and implementing PSG for Kannada sentences. POS tagging plays a main role in NLP. Paper gives a good POS tagger for Kan language.

Part of Speech tagging involves the process of a parser. further use this parser for tree to tree translation. They have used a Part Of Speech tagger for giving proper tags to each and every word in the training and test sentences. The precision of this POS tagger system is obtained by percentage of words that are correctly translated.

Results: The system has been trained with corpus size of 500, 1000 and 1500 sentences for POS tagging. performance of the system has been evaluated with a set of 500 distinct sentences out of the corpus. By forming the experiment they found that the system performance was good and the accuracy got increased with increase in corpus size.

In the paper **On use of BERT for Neural machine translation**[3] which was published in 2019 by Stephane Clinchant, Kweon Woo Jung, Vassilina Nikoulina. This paper involves translation of English text to German and Russian language.

It explores on the approaches on how the BERT pretrained model can be exploited for a supervised Neural machine translation and to evaluate the robustness of the model by doing out-of -domain test set and noise injected test set. They compare various ways like the Baseline, Embedding, Fine-tuning , freeze models were considered to integrate pretrained BERT models with Neural machine translation. The training of the BERT model is done in 3 different monolingual corpora of different size and domain. So using this pretrained model 2 experiments where consider (1)training using the WMT-14 data (2)reusing the model from the previous and then train with IWSLT dataset.

Results: The model was evaluated based on BLEU score for in-domain test set. In order to evaluate the robustness(in terms of out-domain test set and noise in the input) charf scores in addition to BLEU score. BERT+NMT architecture improves over the baseline model both on in-domain and out-domain test sets.Robustness test shows that that there is no improvement in model when the UNK tokens was included which leads to poor translation quality.

The paper **Incorporating BERT to Neural Machine Translation** [5] which was published in 2020 by Jinhua Zhu, Yingce Xia, Lijun Wu etc. uses BERT-fused model, i.e. it extracts representations for an input sequence using BERT. It fuses with each layer of encoder and decoder through attention techniques. The paper includes supervised, unsupervised and semi-supervised learning methods. Datasets used involve Eng-German, Eng-Spanish, Eng-French and Eng-Chinese translation. It feeds BERT-fused representation into all layers instead of being fed as just input embeddings. Input sequence is transformed into representation that BERT processes (pretraining). Then, by the BERT-encoder-attention module, every layer of encoder of NMT interacts with the BERT-representation and outputs-fused representations using BERT+NMT-encoder.

The decoder also fuses BERT and NMT encoder representations.
Attention mechanism controls each layer's interactions with the representation, and deals with the case when BERT as well as NMT modules do not use the same word-segmentation rules, to result in a different sequence length. In addition to BERT, two extra attention modules are present: the BERT encoder attention and the BERT decoder attention.

Results:
Supervised NMT:- 36.11 BLEU score in German-Eng translation, 43.78 on Eng-French,
Semi-supervised:- 39.10 BLEU score,
Unsupervised Eng-French and unsupervised Eng-Roman translations achieve good results.

III. DATASET

The dataset is an English Kannada parallel corpus that has English sequence followed by its Kannada translation, separated by tab space.The English-kannada corpus has 1000 records.

For preprocessing, for the input sequence in English, we convert it from the unicode format to ASCII. In every sentence, we remove special characters such as '?,!,.' as they do not add any value to the model. Finally, we limit the length of English sentences to 22 words.

We use 2 tokens: SOS and EOS to denote start of sequence and end of sequence. We index the words and store them along with their counts before feeding them to the model.

IV. MODEL

Neural MachineTranslation(NMT) requires few characters for converting a sequence of words from source language to target language.

- Ability to persist sequential data over a period of time.
- Ability to deal with variable length input vectors and output vectors.

ANN doesn't satisfy the above required characters but GRU(Gated Recurrent Unit) capable of learning long term dependencies quickly and remembering information over a long steps. They do this by deciding what to remember and what not to remember.

GRU uses two gates i.e. reset gate and an update reset gate. GRU doesn't have an internal memory. Reset gate makes the decision by combining the new input with the steps that are previously stored in the memory. Update gate makes the decision of how much of the past memory should be kept.

GRU has fewer parameters, computationally more efficient, so using GRU we create a sequence-to-sequence translation model on the data.

Seq2Seq model

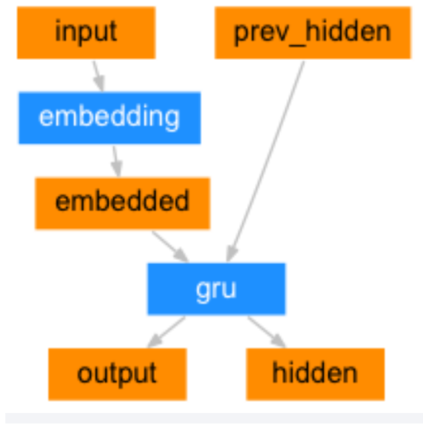
Recurrent Neural Network(RNN) is a network-like structure that functions on a sequence and uses its output as input for coming steps.

A seq2seq network, also called Encoder- Decoder network that model contains two RNNs. Those two RNNs are encoder and decoder. The encoder job is to read an input sequence and output a single vector(i.e. context vector), and the decoder the job is to read that vector to produce an output sentence.

Sentence prediction using a single RNN, where every input corresponds to an output, whereas the seq2seq model frees us from sequence order and length, that is easy to do translation between the languages.

Encoder

The encoder side of our model's network is a RNN that is used to assign value for each and every word of the given input sentence,by doing so all the words are given to the encoder as an input and the output of this will be a vector and hidden state. The hidden state is used for the next input word.

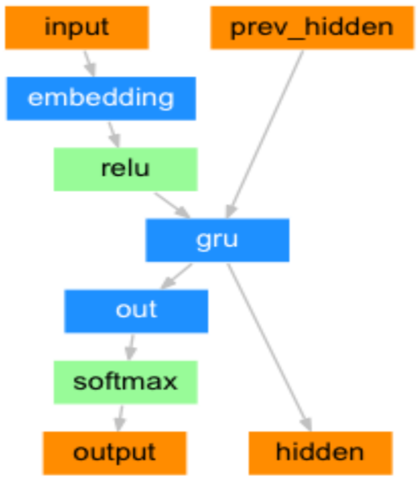


Decoder

The decoder is one more RNN. The input for the decoder is encoder output vectors, and it outputs a sentence of words to create the translation.

Decoder uses only the last output from the encoder. context vector is the last output as it encodes context from the given sentence. This vector can be used as the first hidden state of the decoder side.

On each step of the decoder side, it decodes the hidden state and the given input tokens. The first hidden start will be the context vector and the initial input will always be the token that has a <SOS> (i.e., StartOfSentence) tag.

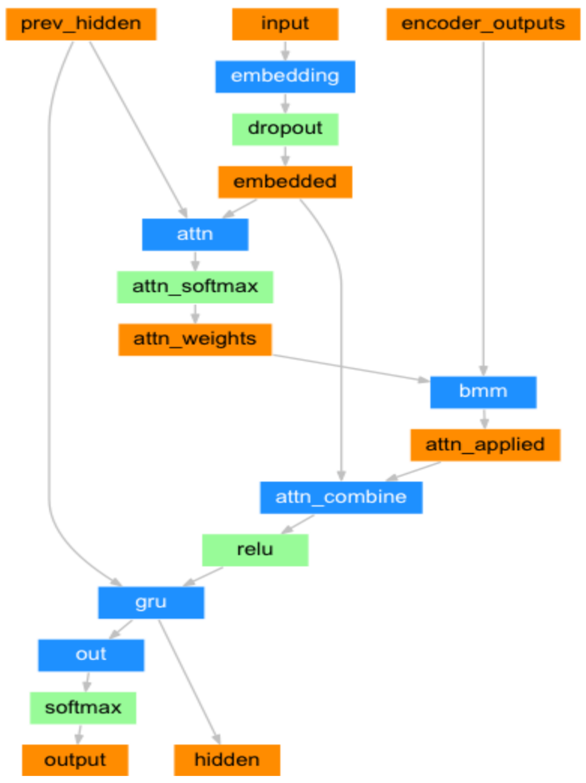


Attention Decoder

The single vector gets all the job of encoding the entire sequence because of passing only the context vector between encoder and decoder.

Attention mechanism makes decoder to concentrate on various part of the encoder's outputs during each step. First we calculate attention weights. Obtained attention weights are multiplied by the encoder output vectors to obtain weighted combination. The results should contain information about that specific part of the input sentence and which enables the decoder to select the right output words.

The work of obtaining the attention weights is performed by one more feed-forward layer, using hidden state and the decoder's input as inputs. Since training corpus has sentences of all sizes, to create and train this attention layer we have to choose a maximum sentence length. Longer sentences will use all the attention weights and shorter sentences will use the first few.

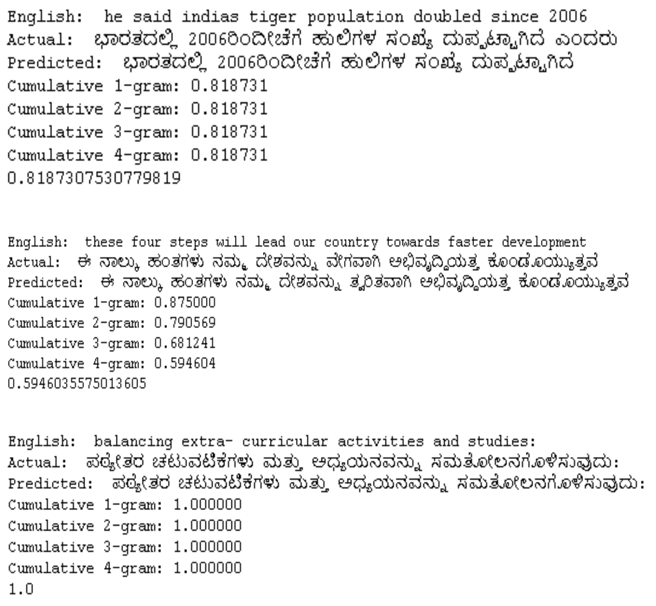


For training the model the data should prepared in which it involves converting the given data that includes the each pair of language sentences to their corresponding input tensor and output tensor(i.e., indexes of the words for both input and output sentences) .On completion of tensors for each sentence <EOS> tag is appended to it.

After the data preparation the model is trained using the tensors which is created. In order to train the given input sentence is given to the encoder and theon decoder side the first input is considered on the input having <SOS> tag and whatever the hidden layer that is generated from the encoder(i.e., the last hidden layer of the encoder), the same is initialized for the decoder's first hidden layer. Encoder keeps track of all outputs and also the latest hidden that is used in the decoder. For training purposes the "Teacher forcing" concept is used in which it uses the real target output as each next input, instead of taking the decoder guess as the next input. This method allows the training to converge at a faster rate and it is used in most of the probabilistic models. For our model we use a teacher forcing ratio of 0.5. The whole training process follows 3 step 1)initializing the encoder, decoder optimizer 2)Creation of training pairs of tensors 3)Calculation of loss on each iteration. Consider 1000 records from the given dataset the training is done for 7000 iterations. On training the 7000 iteration the training loss gradually decreased.

V. EVALUATION

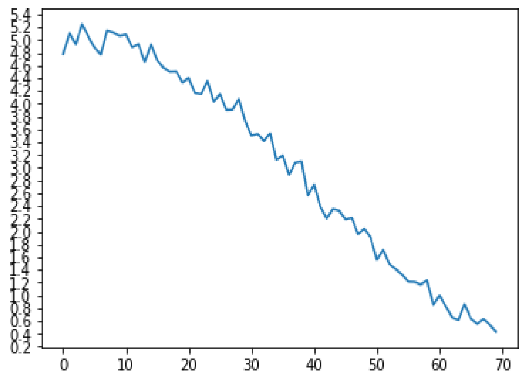
In order to check the quality of translation, BLUE score was used. For evaluating the translation randomly 50 records were generated from the given dataset in which only the input english sentence was considered. The model predicts the translation for the given input sentence and then calculates the N-gram(1,2,3,4) BLUE score and finally cummates the score to give the final BLUE score of the particular translated sentence.



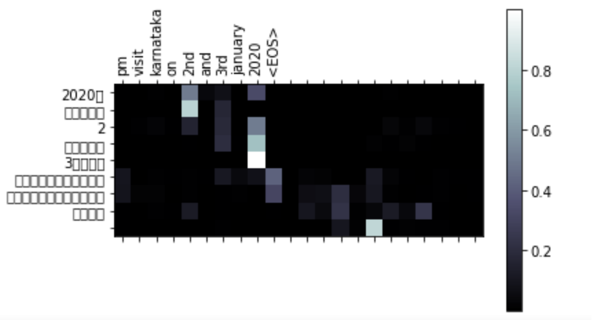
```
English:  he said indias tiger population doubled since 2006
Actual:  ಭಾರತದಲ್ಲಿ 2006ರಿಂದೀಚೆಗೆ ಹುಲಿಗಳ ಸಂಖ್ಯೆ ದುಪ್ಪಟ್ಟಾಗಿದೆ ಎಂದರು
Predicted:  ಭಾರತದಲ್ಲಿ 2006ರಿಂದೀಚೆಗೆ ಹುಲಿಗಳ ಸಂಖ್ಯೆ ದುಪ್ಪಟ್ಟಾಗಿದೆ
Cumulative 1-gram: 0.818731
Cumulative 2-gram: 0.818731
Cumulative 3-gram: 0.818731
Cumulative 4-gram: 0.818731
0.8187307530779819


English:  these four steps will lead our country towards faster development
Actual:  ಈ ನಾಲ್ಕು ಹಂತಗಳು ನಮ್ಮ ದೇಶವನ್ನು ವೇಗವಾಗಿ ಅಭಿವೃದ್ಧಿಯತ್ತ ಕೊಂಡೊಯ್ಯುತ್ತವೆ
Predicted:  ಈ ನಾಲ್ಕು ಹಂತಗಳು ನಮ್ಮ ದೇಶವನ್ನು ತ್ವರಿತವಾಗಿ ಅಭಿವೃದ್ಧಿಯತ್ತ ಕೊಂಡೊಯ್ಯುತ್ತವೆ
Cumulative 1-gram: 0.875000
Cumulative 2-gram: 0.790569
Cumulative 3-gram: 0.681241
Cumulative 4-gram: 0.594604
0.5946035575013605


English:  balancing extra- curricular activities and studies:
Actual:  ಪಠ್ಯೇತರ ಚಟುವಟಿಕೆಗಳು ಮತ್ತು ಅಧ್ಯಯನವನ್ನು ಸಮತೋಲನಗೊಳಿಸುವುದು:
Predicted:  ಪಠ್ಯೇತರ ಚಟುವಟಿಕೆಗಳು ಮತ್ತು ಅಧ್ಯಯನವನ್ನು ಸಮತೋಲನಗೊಳಿಸುವುದು:
Cumulative 1-gram: 1.000000
Cumulative 2-gram: 1.000000
Cumulative 3-gram: 1.000000
Cumulative 4-gram: 1.000000
1.0
```

The average BLEU score obtained by evaluating 50 randomly generated records was 0.76.

VI. DISCUSSION OF RESULTS



The above graph shows the training loss incurred on training for each iteration. The y-axis of the graph indicates the loss , y-axis the number of iterations. From the graph we came to a conclusion that on each iteration the training loss keeps decreasing.



The above graph shows inputs in x axis and outputs in y axis, while the boxes represent the attention layer output which denotes the importance of certain output words for a given input (context).

A good BLEU score was obtained for inputs that are similar to sentences in the dataset. However, building a larger dataset of higher quality and more variety of news would help improve the accuracy of predictions. Also, the training of the model is time-consuming.

VII. CONCLUSION

In this paper we have explained the concepts and methods presented while implementing Neural Machine Translation System for English to Kannada which translates a given input sentence in source language into target language using seq2seq approach. This work can be extended to other domains with the addition of new rules. inclusion of pre-training models like BERT. Comparing the results with the Google Translate API will also be helpful.

REFERENCES

[1] Mallamma.V.Reddy,.Hanumanthappa.M , Kannada and Telugu Native Languages to English Cross Language Information Retrieval" published in (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , Sep-Oct 2011, page-1876-1880. IISN: 0975-9646.

[2] Mallamma.V.Reddy,.Hanumanthappa.M, “POS Tagger for Kannada Sentence Translation ” published in (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 1, May-June 2012, ISSN 2278-6856.

[3] Stephane Clinchant, Kweon Woo Jung ,Vassilina Nikoulina “On the use of BERT for Neural Machine Translation” in Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019),Nov 2019, pages 108–117

[4] Mallamma V.Reddy and M.Hanumanthappa “Indic Language Machine Translation Tool :English to kannada/Telugu” in Proceeding of 100th Science Congress Organized by Indian Science Congress Association (ISCA) - Young Scientists’ Award Programme. 3rd -7th Jan 2013

[5] Incorporating BERT to Neural Machine Translation by Jinhua Zhu1;, Yingce Xia2;, Lijun Wu3, Di He4, Tao Qin2, Wengang Zhou1, Houqiang Li1, Tie-Yan Liu

[6] Improving the Transformer Translation Model with Document-Level Context by Jiacheng Zhangy, Huanbo Luany, Maosong Suny, FeiFei Zhai, Jingfang Xu, Min Zhangx and Yang Liuyz