International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Kannada to English Machine Translation Using Deep Neural Network

Sangamesh Ramesh Yankanchi[1], Abhishek Rathod[2]

[1,2]School of Computer Science Engineering (SOCSE) RV University, RV Vidyanikethan Post 8th Mile, Mysuru Road, Bengaluru – 560059

Abstract

This paper proposes an NMT model which takes Kannada text as input and translates to the corresponding English using a sequence-to-sequence architecture with LSTM layers. The mechanism of encoder-decoder facilitates capturing and holding onto sentence context and, therefore, improves the accuracy of translation. Accuracy of 90.32% was obtained by our approach. In the future work, Gated Recurrent Units and other transformer architectures may be extensively investigated for further improvement in Kannada and other low-resource language translation.

Keywords: Neural Machine Translation, Kannada, Sequence-to-Sequence, LSTM, Low-resource Languages, Deep Learning

1. INTRODUCTION

This paper discusses motivation, challenges, and impor- tance of developing an effective Kannada-to-English machine translation model. In a globalizing world, machine translation (MT) plays a leading role as it bridges language barriers, enforces cross-cultural understanding, and allows for informa- tion to flow across languages [1]. Despite huge development in the translation of languages like English, Spanish, and Chinese, many languages, such as Kannada, remain under- served simply because of their resource and data scarcity [8]. Translation of Kannada would be challenging mainly because of its peculiar syntax, rich morphology, and vocabulary, as more than 40 million people communicate in the language in the Indian state of Karnataka.

Translation of Kannada into English plays an important role in increasing the educational and professional resources available for Kannada speakers, access to information world- wide, and preservation of culture [13]. However, the linguistic difference and unavailability of quality bilingual data make translation into English more complex. This work conducts research in designing a robust translation model based on Seq2Seq architecture [2] along with the mechanisms of atten- tion [4], trying to overcome challenges that arise and related future research. The results carry the potential to promote advancements in low-resource language machine translations as well as enhance digital inclusivity for Kannada-speaking communities.

2. RELATED WORK

This section reviews existing research, methods, and tech- nological improvements pertinent to Kannada-English MT. MT has traversed a long journey from its initial forms of rule-based and statistical approaches relying on predefined linguistic rules or large dictionaries, but it failed

miserably in capturing complex language structures that needed manual intervention. Then came the IBM Model series, which was a breakthrough in the probabilistic methods by leveraging statistical patterns in bilingual corpora. However, these initial approaches were limited, especially when it came to languages like Kannada, which lack parallel data and possess a syntactic structure much different compared to major languages.

Neural networks and enhanced architectures evoked a rapid increase in accuracy [3]. After that, the limelight brought Seq2Seq models with encoder-decoder based structures into view, since they effectively captured sequential dependencies [6]. The introduction of attention mechanisms, especially in Transformer models [5], allowed the model to dynamically focus on the relevant words and, therefore, carry out more coherent translations while handling complex sentence struc- tures [18]. Low-resource language translations like Kannada still prove challenging because of minimal datasets and the complexity of Kannada's morphology [7]. Some recent strate- gies–transfer learning, for example, or unsupervised learn- ing–promise low-resource MT but are rarely applied to Kan- nada [15]. This contribution addresses these gaps using a Seq2Seq model adapted with special attention mechanisms customized to the complexities of Kannada language in order to contribute toward low-resource MT.

3. METHODOLOGY

It would call for several stages of development: data prepro- cessing, architectural design of the model, training, and finally testing and evaluating the resulting machine translation model for Kannada-to-English.

A. Data Collection and Preprocessing

Since Kannada is a low-resource language, gathering a sufficiently large and clean dataset is challenging. For the task in hand, we have used a parallel Kannada-English dataset consisting of sentence pairs which helps the model learn contextual as well as syntactic relationships between the two languages. Preprocessing is a must to clean, normalize and tokenize the data [9]. This includes:

Text Normalization: Kannada and English text should be lowercased, punctuation removed, and special characters handled.

Tokenization: Breaking a sentence into words or subword units since tokenization is intrinsic in handling meaning in a very morphological language such as Kannada.

Padding and Sequence Length Control: Since each and every one of the sentences had different lengths we padded each sequence to a fixed length MAX_LENGTH to the max- imum length of words in our data, which happened to be at 15 words long sentences.

Dictionaries: Created a word-to-index and an index-to- word lookup dictionary both for Kannada as well as English translation during training time.

B. Model Architecture

The encoder-decoder model, with attention below explains the various sub-components within that architecture.

Encoder (Encoder RNN): The encoder is specifically designed to read in the input Kannada sentence and encode it into a series of hidden states [16]. Each word within the sentence is mapped into an embedding vector that the GRU processes, capturing sequential dependencies.

Attention Mechanism (AttnDecoderRNN): Syntax is very different between Kannada and English; hence, an attention mechanism is required. The reason it is introduced is that at each decoding time step, it will allow the decoder to attend to certain encoder outputs by dynamically shifting to relevant

parts of the input sentence to attain better quality translation.

Decoder with Attention (DecoderRNN): It feeds one word at a time into the decoder. The decoder reads the hidden states of the encoder and its attention scores to generate the probability distribution over the vocabulary in the target language English. This mechanism is very important in order to handle syntactic and contextual shifts from Kannada to English.

Input sentence	ಅದು ನನಗೆ ಇಷ್ಟ
Output sentence	I like that
Input sentence	ನನಗೆ ಸಹಾಯ ಮಾಡಿ
Output sentence	help me
Input sentence	ಅವಳು ಯಾರು?
Output sentence	who's is she?

Fig. 1. Kannada to English Sentence Translation Examples

C. Training Process

Hyperparameters: Training involved a hidden layer size of 100 units and dropout set at 0.1 for regularization, learning rate to 0.01 [17]. Parameters for the training were 'teacher forcing ratio', indicating points at which to feed it the true target word and to use predictions from the model instead.

Optimization and Loss Calculation: SGD optimizer up- dated the weights of the model. Cross-entropy loss calculated how accurate the predictions are. Periodic checks on loss would be done to check upon the learning progress for 9500 iterations.

Teacher Forcing: Teacher Forcing is applied at a rate of 50% while training. Feed this model with the original output word from the previous step, not with its prediction. This method can result in fast learning in the early periods by inducing randomness to the predictions to make it more robust.

D. Evaluation and Testing

We use a function to check the quality of translation by testing the model on new Kannada sentences, which is achieved by passing the Kannada sentence through the trained encoder-decoder model and translates it into the corresponding English sentence.

Metrics: To assess the quality of translation, we employed BLEU score [14] and accuracy of translation regarding fluency and adequacy of produced sentences in English.

E. Deployment

The final trained model weights and lookup dictionaries for Kannada-to-English translation were saved and serialized for deployment. Models may be loaded, and real-world Kannada sentences can be evaluated for translations without any further retraining. It is built to be open for updating or retraining with new data if further Kannada-English datasets are available. This methodology captures Kannada-to-English syntactic and semantic mapping effectively making use of the Seq2Seq ar- chitecture as well as the attention mechanisms for dealing with low-resource languages. In fact, the quality of the translations has improved enough; hence it adapts to the nuances of the Kannada language very well.

4. DESIGN AND IMPLEMENTATION

1. Data Preprocessing

Data Collection: **Dataset Sourcing:** Gather parallel corpus of Kannada-English sentence pairs [19]. This is split into folders named "Kannada" and "English." Paths to these text files are defined for preprocessing.

Data Collection Challenges: As it is a low-resource lan- guage, Kannada would not have many

examples in the dataset. For better performance of the model, data collection needs to be done from different sources such as public datasets available for different languages and language translation repositories.

Creating a Dataframe: Create DataFrame. The function, that we will name 'createdf', reads Kannada-English sentence pairs into a 'pandas' DataFrame where each row represents a Kannada sentence along with its translation into English. This is an easy way to create a usable system for data handling and preparation for the model.

Preprocessing Steps: All sentences are tokenized to have a uniform format. This also includes the removal of special characters and management of text normalization, which is quite significant for Kannada as its script representation is pretty divergent.

Data Splitting and Encoding: **Training and Validation Splits:** Using the 'train_test_split', the dataset splits between the training and validation set to avoid data leakage and ensure that the model generalizes.

Token Encoding: For large vocabulary sizes in natural language tasks, Kannada and English words are encoded using a vocabulary dictionary or byte-pair encoding (BPE). This encoding maps each token to a unique integer that makes possible the training of neural networks.

Text Transformation: Padding and Sequences: Sentences are transformed into sequences of a certain fixed length, and padded if necessary, to ensure uniform input for the model.

Tensor Transformation: This is where the encoded se- quences are passed in the tensor format to the neural networks via PyTorch or TensorFlow.

2. Training Model

This section states the architecture of the neural network, which takes the optimization parameters along with the pre- processed Kannada-English dataset to train the model.

Seq2Seq with LSTM Architecture: **Encoder-Decoder Structure:** A Seq2Seq model using LSTM units is adopted as follows:

Encoder: It accepts the input Kannada sequences and encodes them as context vectors from which semantic meaning is derived.

Decoder: It decodes the acquired context vectors to the English translation sequences.

Attention Mechanism: To enhance the accuracy of trans- lation, attention is introduced that enables the model to pay attention to the relevant parts of the input Kannada at each decoding step.

Regularization: Dropout layers will be added in preventing overfitting, especially if you have a smaller set.

Transformer Architecture (Optional): **Multi-Head At- tention and Self-Attention Layers:** Apply transformer layers, but especially multi-head attention will make the model learn context relations between words in both languages, Kannada, and English.

Positional Encoding: Transformers do not include built-in sequence order handling and can be used by feeding them positional encoding.

Optimization and Loss Function: **Loss Function:** In text translation, the most normally used type of loss function is cross-entropy to maximize the probability for correct word prediction at every step in time.

Optimizer: Stable training can be ensured using Adam or SGD with a learning rate scheduler. The fine-tuning of hy- perparameters such as learning rate, batch size, and sequence length enhances the model.

3. Translation Evaluation

Now, the model is evaluated against new Kannada sentences to see how well they are translated. It decodes the prediction in words and returns the English sentences.

Evaluation Metrics: BLEU Score: The translation of the model is compared against the original reference English sentences with a BLEU score. The BLEU score serves as the measure of similarity. This would imply quality control in translation through a review process by humans that remove and fill in the flaws of the computer metrics, more so concerning the linguistic matters regarding Kannada language.

4. MODULAR DESIGN APPROACH

Every part in this pipeline is designed such that changes in various parts may be made easily by modifying parts.

Data Processing Module: It takes care of all the preprocessing, tokenization, padding, and conversion of the text to tensor. This module makes it highly easy to change the mode of dealing with input data in a model without affecting any architecture.

Model Module: It contains definitions of Seq2Seq model, LSTM or Transformer. Due to the smooth adjustment made in architectures and hyper parameter tuning the model module here is relatively flexible.

Evaluation Module: The package includes evaluation metrics and translation testing functions. With these in separate packages, it can easily integrate newer evaluation functions if they should prove necessary to use.

5. IMPROVEMENTS TO INITIAL DESIGN

Other modifications that improve the accuracy and effi- ciency of these models include:

Data augmentation: Backtranslation is used to further add variety to the data.

Learning rate schedulers: there are schedulers used for boosting the stability of training to complex models.

Fine-Tuning the Attention Layers of the Transformer: This fine-tuning actually enhances the performance of dealing with complex sentence structures in the transformer.

6. EXPERIMENTAL RESULTS

This section reports the results obtained for the Seq2Seq models using LSTM and Transformer architectures. For Kannada-English machine translation, BLEU scores are re- ported along with the improvements on the synthetic augmen- tation data used.

Summary of Model Performance:

Seq2Seq LSTM: Kannada-English average BLEU score
= 90.32%

Transformer: Reaches better generalization than before achieves much higher BLEU scores.

It gives an outline of the possibility of using deep learning techniques, specifically Seq2Seq with LSTM and Transformer- based architectures in the translation of Kannada-to-English. Data augmentation methods proved to be very important in im- proving performance using this model. The modular pipeline and flexibility towards any kind of future improvements, such as adapting new architectures or more sources of data, can easily be scaled up to other low-resource language pairs [10].

7. CONCLUSION AND FUTURE WORK

This paper explores the design and evaluation of an attention-based Seq2Seq architecture Kannada-to-English ma- chine translation model. Due to the complexity involved with Kannada's morphology and

syntactical differences with that of English, the challenge is quite high when attempting machine translation in the absence of abundant digital resources. Our model used attention mechanisms towards achieving better contextual word focus with higher translation accuracy com- pared with the challenges of word order and coherence. This way, it bridged language gaps, expanded access to English- based digital content, and preserved cultures for Kannada- speaking communities.

Further improvement in this model can be achieved by pre- trained multilingual models such as mBERT [11] or mT5 [12] with further quality translation even with a limited Kannada- English dataset. Further research into artificial data creation, like back-translation or data augmentation may also be worth-while, as the former would possibly solve data sparsity in low-resource languages. In addition, studying the transfer learning from other Dravidian languages will help tap their similarity of linguistic features to build an enhanced Kannada- English translator. By doing so, it could eventually make way for more inclusive linguistic conditions for Kannada and similar low-resource languages in machine translation and other applications of natural language processing [20].

REFERENCES

1. Chaudhary, J.R., Patel, A.C. (2018). Machine translation using deep learning: A survey. International Journal of Scientific Research in Science, Engineering and Technology, 4(2): 145-150.
2. Sutskever, I., Vinyals, O., Le, Q.V. (2014). Sequence to sequence learn- ing with neural networks. NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2: 3104-3112.
3. Basmatkar, P., Holani, H., Kaushal, S. (2019). Survey on Neural Machine Translation for the multilingual translation system. 3rd International Conference on Computing Methodologies and Communication (Erode,
4. India), pp. 443-448. https://doi.org/10.1109/ICCMC.2019.8819788
5. Bahdanau, D., Cho, K., Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. International Conference on Learning Representations (ICLR).
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30: 5998-6008.
7. Cho, K., Van Merrie¨nboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734.
8. Koehn, P., Knowles, R. (2017). Six challenges for neural machine translation. Proceedings of the First Workshop on Neural Machine Translation, pp. 28-39.
9. Zoph, B., Yuret, D., May, J., Knight, K. (2016). Transfer learning for low-resource neural machine translation. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1568-1575.
10. Sennrich, R., Haddow, B., Birch, A. (2016). Neural machine translation of rare words with subword units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1715- 1725.
11. Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Vie´gas, F., Wattenberg, M., Corrado, G., Hughes, M., Dean,

12. J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. Transactions of the Association for Computational Linguistics, 5: 339-351.
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2019). BERT: Pre- training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of NAACL-HLT, pp. 4171-4186.
14. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C. (2021). mT5: A massively multilingual pre- trained text-to-text transformer. Proceedings of the 2021 Conference of NAACL-HLT, pp. 483-498.
15. Antony, P.J., Soman, K.P. (2012). Machine translation system for Indian languages using computational paninian grammar. International Journal of Computer Applications, 39(1): 1-6.
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J. (2002). BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311-318.
17. Sennrich, R., Haddow, B., Birch, A. (2016). Improving neural machine translation models with monolingual data. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 86-96.
18. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8): 1735-1780.
19. Kingma, D.P., Ba, J. (2015). Adam: A method for stochastic optimiza- tion. International Conference on Learning Representations (ICLR).
20. Luong, M.T., Pham, H., Manning, C.D. (2015). Effective approaches to attention-based neural machine translation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412-1421.
21. Kunchukuttan, A., Mehta, P., Bhattacharyya, P. (2018). The IIT Bombay English-Hindi parallel corpus. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).
22. Ramesh, A., Sanampudi, S.K. (2021). An efficient neural machine translation model for Indian language translation. Journal of King Saud University - Computer and Information Sciences, 33(10): 1214-1223.