International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Ahirani Language Translation with Neural Networks: An English Perspective

Dr. Uday C. Patkar[1*], Dr. Shikha Bhardwaj[2], Dr. U. S. Patil[3] , Dr. Yogita Deepak Sinkar[4], Dr. Ratna Nitin Patil[5], Dr. Neha Ganvir[6], Om A. Bhamare[7]

Submitted: 07/01/2024 **Revised**: 13/02/2024 **Accepted**: 21/02/2024

Abstract: Machine Translation is the process of converting one natural language into another using computer-assisted techniques. The primary objective is to bridge the linguistic divide that exists among individuals, communities, or nations with distinct languages. India, being a multilingual country, features various regional languages spoken across its diverse regions. However, not all Indians are proficient in multiple languages. India recognizes 18 constitutional languages and employs ten widely-used scripts globally. A significant portion of the Indian population, particularly in remote rural areas, lacks proficiency in understanding, reading, or writing English. This underscores the importance of implementing an efficient language translation system.

Machine translation systems capable of translating content from one language to another can significantly contribute to the intellectual and cultural enrichment of the Indian population, irrespective of their native languages. Although Marathi is the most prevalent language in the state of Maharashtra, Ahirani is also widely spoken. Given the universality of the English language and the prevalence of Ahirani among Khandeshi people, we propose the development of an English to Ahirani machine translation system using Recurrent Neural Networks (RNNs)..

Keywords: *Cross-Linguistic Translation, Cross-Language Communication, NLP (Natural Language Processing), Translation Model, Ahirani Language, Translation System, Neural Machine Translation, Language Pair, Deep Neural Networks*

1. Introduction

Machine translation, which has been under continual research and development since the 1940s, remains at the forefront of technological advancement. The trajectory of machine translation continues to ascend, serving as a powerful tool for the conversion of text or speech between natural languages. This capability holds the potential to surmount language barriers and foster effective communication.

Within the realm of computer science, Natural Language Processing (NLP) endeavors to bridge the linguistic gap, with neural machine translation representing a conceptually straightforward approach. Here, the focal point lies in training extensive neural networks capable of generating lengthy linguistic sequences. Notably distinct from conventional machine translation systems, this model deliberately incorporates substantial phrase dictionaries and language models to enhance efficiency.

An early milestone in machine translation history was achieved through a collaborative effort between Georgetown University and IBM in 1965, marking the initial deployment of a machine translation system. The significance of machine translation in multilingual societies is underscored by its social and political relevance, which prompts the exploration of attention mechanisms within the field.

Machine translation, however, grapples with the challenge of addressing translation discrepancies arising from grammatical variations among languages. These disparities manifest as differences in grammar, leading to notable divergence when translating sentences from one language (L1) to another (L2). Several studies have explored this phenomenon, offering a range of techniques to mitigate translation disparities. It becomes imperative for machine translation systems to incorporate mechanisms for identifying and resolving such discrepancies to ensure the precision and dependability of translations.

Translation discrepancies manifest at varying levels of complexity, with their impact on overall translation quality commensurate with their intricacy. While certain types of translational differences are universally present across languages, others are specific to particular language pairs, as evidenced in existing literature. Thus, a comprehensive understanding of translation discrepancies necessitates examination from both an inter-language and language-specific perspective.

This research is centered on the English to Ahirani translation language pair, aiming to uncover language-specific distinctions that contribute to translation

[1] Bharati Vidyapeeth's college of Engineering, lavale, Pune, (MS), India. uday.patkar@bharatividyapeeth.edu**
[2,3,7] Bharati Vidyapeeth's college of Engineering, lavale, Pune, (MS), India
[4] SVPM COE MALEGAON BK Tal, Barmati dist, Pune, India
[5] Galgotias College of Engineering and Technology, Greater Noida, India
[6]KJCOEM&R Pune, India

differences. Given the significant dissimilarities between English and the Ahirani dialect, this pairing offers an ideal platform for investigating translation discrepancies in machine translation. These languages not only differ linguistically but also structurally and socioculturally, warranting comprehensive scrutiny.

The primary objective of this work is to scrutinize select aspects of English and Ahirani grammar that may underpin translation discrepancies observed in English-Ahirani machine translations. We delve into the issue of syntactic and structural divergence within English-Ahirani machine translation, employing back-translation to dissect the nature of divergence in distinct contexts.

To frame our investigation, we employ Dorr's taxonomy of translation differences between English and Ahirani as a foundational framework. The subsequent sections of this paper encompass a review of literature on translation discrepancy classification, elucidation of lexical-semantic and syntactic disparities, a detailed implementation of recurrent neural networks for machine translation, and a comprehensive discussion of our research findings.

2. Review Of Literature

Natural language processing (NLP) plays a pivotal role in text and speech manipulation systems. In recent research endeavors [1-5], a text-to-speech system has been developed, incorporating NLP techniques before engaging in digital signal processing (DSP) to synthesize spoken words. This innovation has yielded a straightforward yet valuable text-to-speech application capable of converting written text into audible speech, subsequently saving the output as an MP3 file.

Efforts to bridge the linguistic gap for the deaf community hold immense potential for enhancing the lives of this often-neglected population. Existing literature on English text-to-speech technology has acknowledged the complexities inherent in this endeavor, with only limited success to date. Those intrigued by the realm of text-to-speech conversion are encouraged to explore research within their local languages, where more promising results may be found.

In a notable study [6], a technique was proposed for converting textual content into speech, bypassing the conventional step of constructing concatenated linguistic data. Instead, this method directly converts from registered language parameters. The effectiveness of this algorithm was compared globally with other synthesizers commonly employed for language learning and communication enhancement. For individuals not fluent in English, this approach allows them to visualize how content would be rendered in their native language by examining an enlarged grid of symbols displayed on a computer screen. This study encompasses data from 53 distinct languages, paving the way for potential applications across a wide array of languages and opening up new research avenues. Furthermore, users can retrieve information from images based on their language preferences [7].

Graphing, in the context of extracting text from images and translating it into multiple languages for viewing, utilizes Google Speech API and Tesseract OCR [Optical Character Recognition] in tandem. This technology proves valuable for travelers seeking relaxation by listening to audio content in their preferred language, including those with visual impairments. It also facilitates the reading of text in its original language.

Building upon effective rule-based text-to-speech synthesis, this work combines natural phonetics and lexical formant approaches [8] to refine the final output, encompassing words and concatenated phrases. Simulations demonstrate that this system outperforms proposed approaches in handling words, phrases, sentences, and paragraphs, particularly in Marathi, with a remarkable 91% accuracy rate. Further refinement is necessary to enhance the detection of stressed syllables and vocal intensity.

The article dissects text into its constituent components and assembles them into spoken language. Voices are stored within a database and retrieved as required. One notable study [9] employed concatenated-input-based speech synthesis on the MATLAB 2010 platform, renowned for its clear and high-quality audio output. This technology is increasingly recognized as an aid for individuals with dyslexia and visual impairments, supporting foreign language skill development and pronunciation improvement. Its applications extend to reducing eye strain during reading (both in print and digital formats), minimizing document production costs through digital text printing, enabling English translation, facilitating digital writing/editing, and enhancing listening comprehension.

Abstract: This research focuses on the translation of simple English sentences into assertive Marathi sentences, leveraging rule-based techniques outlined in a previous study [10]. The algorithm's core functionality lies in accepting a straightforward sentence as input and providing a corresponding lexeme or description of the output situation.

The tokenizer employed in this algorithm identifies each newly formed English letter and cross-references it with entries in the English lexicon. When a lexical item has associated tokens, the algorithm can optionally access the morphological features linked to those tokens. Notably, this approach classifies morphemes based on inherent word features, eliminating the need for pre-extraction, thus saving time.

Additionally, specific syntactic groups established within the algorithm contribute to grammatical correctness in sentence construction. The syntax verification in the context of bottom-up parsing examines the parsing depth. Moreover, the program's search dictionary has been enriched with syntactic tokens specific to Marathi, enabling accurate identification of Marathi words within a sentence. In cases where a required Marathi word cannot be identified word by word and connected to the corresponding English word, the algorithm generates a Marathi sentence by collaborating with someone who can provide an English translation using the remaining text. This research encompasses various aspects of sentence formation principles in English, exploring diverse vocabulary rearrangement techniques, and scrutinizing the intricacies of language proofreading.

In parallel, the authors of this study [11] have verified the applicability of the Marathi language within the research context. Two distinct corpora, each representing the most frequently occurring terms and interpretations of text patterns, have been derived from the original text. This process utilizes SMORDT (Sequential Minimum Optimality using Statistical Similarity Models) and a rules transformation-based decision tree approach. The study emphasizes the significance of feature extraction, demonstrating its importance through experimentation and validation. The proposed method has undergone extensive testing, evaluating its performance based on key criteria, including recall, accuracy, and precision.

As urbanization progresses, the demand for clean air has surged. Technological advancements have expanded data volume exponentially, doubling every two years. Consequently, computers are expected to autonomously acquire, analyse, interpret, and apply data with minimal human intervention. However, certain text segments within source code present challenges in interpretation, often featuring "code mixing" or resembling machine code-like languages, which are notoriously difficult to parse. This research [12] actively contributes to classifying and evaluating such documents within learning systems, including Bag of Words and Human Priors (NB, SVM), and addresses Marathi and Hindi text translation documents. Machine learning algorithms demonstrate promising results, often on par with analytical approaches.

Furthermore, when composing sentences in Marathi, an innovative approach [13] recommends initiating each phrase with two words to clarify word breaks. This is particularly beneficial for native Marathi speakers since Marathi lacks the sentence markers present in English. Distinguishing between sentence beginnings and endings relies on a reliable system that considers a substantial amount of contextual information.

Lastly, translation applications [14-15] tailored for language minorities who seek to engage with Kerala's culture and language enhance machine translation and text-to-speech accuracy. Leveraging combined approaches encompassing machine translation and text-to-speech, these applications enable Malayalam texts to be translated into native languages. The process involves grammar-based translation, morphological analysis, and a Malayalam-Tamil-English dictionary. Iterative improvements have increased translation accuracy to 73% over three stages. To ensure speech output accuracy, the system incorporates naturalness and intelligibility characteristics, with 87% of sentences deemed well-formed during evaluation.

3. RNN for Machine Translation

RNNs are a type of neural network designed to handle sequential data, such as time series data, sentences, or any data where the order of elements matters. They are characterized by their loops or recurrent connections, which allow them to maintain a hidden state that carries information from one step in the sequence to the next.

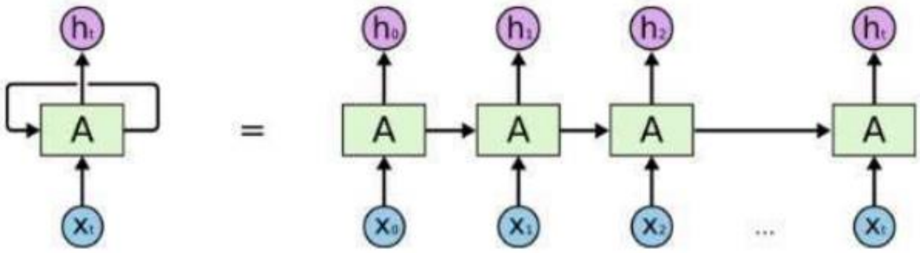

Fig A. An unrolled recurrent neural network.

Sequential Information Handling:

In the context of machine translation, RNNs excel because they can consider the context of a word within a sentence by looking at the words that came before it. This allows them to capture dependencies and relationships between words in a sentence, which is essential for translation tasks.

Repetition of Network:

The notion of "copying and pasting the same network over and over again" refers to the recurrence in RNNs. In each step of the sequence, the same set of weights is used to process the input data, and the network's hidden state is updated. This recurrent structure enables RNNs to maintain a memory of past inputs, which can influence their predictions at each step.

Challenges with RNNs:

A challenge associated with traditional Recurrent Neural Networks (RNNs) is the vanishing gradient problem. During the training of an RNN using gradient-based optimization methods like backpropagation, the gradients may diminish significantly as they propagate backward through the network. Consequently, the earlier layers in terms of the sequence learn at a sluggish pace, making it difficult for the RNN to effectively capture long-range dependencies in the data.

Effectiveness and Sequence Length:

The efficacy of Recurrent Neural Networks (RNNs) may diminish as the temporal gap between relevant information in a sequence expands. This challenge arises because recurrent connections may encounter difficulties in preserving valuable information for extended sequences. Consequently, the network faces challenges in making precise predictions or translations, particularly when dealing with distant words.

To address some of these challenges, more advanced RNN variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been developed. These variants are designed to better handle long-range dependencies and mitigate the vanishing gradient problem.

In the context of machine translation, RNNs have been widely used, but they have been largely supplanted by more advanced architectures like Transformer-based models (e.g., the GPT series and BERT) and sequence-to-sequence models with attention mechanisms (e.g., the Transformer). These models have proven to be highly effective for handling translation tasks and other natural language processing tasks.

An RNN also has different models that it can follow, this study follows a sequence-to-sequence model. Here the RNN receives an input sequence and outputs a sequence.

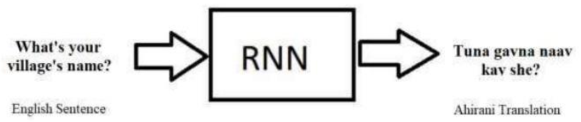

Fig B. Machine Translation: an RNN reads a sentence in one language and then outputs it in another

Configuration of the RNN may vary depending on your application requirements, necessitating adjustments for handling inputs and outputs. In the context of this project, as illustrated below, a many-to-many process is employed. The input consists of a sequence of English words, and the output is a sequence of Japanese words (depicted fourth from the left in the figure below).

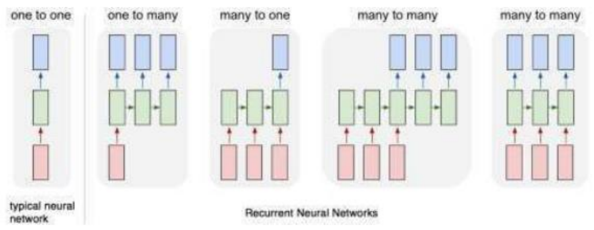

Fig C. Recurrent Neural Network

The following section outlines various preprocessing and modelling steps. Here are the high level steps:

1. **Data Preparation:** This phase involves tasks such as data reading, examination, and data cleaning to ensure data quality. Additionally, it includes tokenization and padding for further processing.
2. **Model Development:** This step encompasses the creation, training, and evaluation of machine translation models.
3. **Evaluation and Prediction:** To assess the model's performance and make predictions, we generate specific translations from English to Ahirani and compare them with the reference or ground truth translations.
4. **Iterative Model Enhancement:** To enhance model performance, we continually iterate on the model architecture and experiment with various configurations to achieve better results.

In this study, we utilize Keras as the frontend and TensorFlow as the backend to achieve our objectives. This combination is selected for its streamlined syntax, which enhances the intuitive creation of model layers. It is essential to mention that while Keras simplifies the process, it might restrict the ability to fine-tune certain aspects. However, for the model developed in this research, this limitation is not a major concern.

PREPROCESSING

Load & Examine Data

We begin by loading and inspecting our dataset. The dataset consists of English input sentences and their corresponding translations into Ahirani. Notably, the dataset's vocabulary is relatively limited, tailored specifically for this project to ensure efficient training.

Cleaning

Fortunately, there is no need for further cleaning at this point. The data has already undergone conversion to lowercase and has been appropriately segmented, with spaces between words and punctuation.

Tokenization

Tokenization involves transforming text into numerical representations. Each word and punctuation mark is assigned a distinct ID number. This process results in a word index that simplifies the conversion of sentences into vector form.

Padding

To input sequences into our model, it is essential to guarantee that all sequences share the same size. Padding is implemented for sequences shorter than the maximum length, ensuring uniformity in size.

Modelling:

To comprehend the architecture of a Recurrent Neural Network (RNN), let's begin with a high-level overview. As illustrated in the diagram above, there are several crucial components of the model that require attention:

Inputs: The input sequence is fed into the model word by word, with each time step representing a different word. Each word is encoded either as a single integer or as a one-hot encoded vector corresponding to the word in the English lexical database.

Embedding Layers: Here, each word undergoes conversion into a vector through embedding. The size of the vector representing the vocabulary is influenced by the complexity of the vocabulary. This process involves an iterative shift, a repetition in the encoder, where the context from the previous time step word vector is applied to the current word vector.

Dense Layers (Decoder): These layers consist of typical fully connected layers utilized to transform the encoded input into the accurate translation sequence, as depicted in the diagram.

Outputs: The output is presented as a series of integers or as a one-hot encoded vector, which can subsequently be mapped to the French record vocabulary after undergoing processing.

Embedding

Embeddings play a pivotal role in establishing straightforward semantic and syntactic relationships among words. Essentially, each word undergoes projection onto an n-dimensional space, resulting in a distinct representation. In this space, words with similar sounds are typically positioned closer to each other, indicating a higher level of similarity in meaning. This spatial arrangement often mirrors a complex network of connections, encompassing aspects such as gender, prepositions, and even strategic associations between proverbs.

Training embeddings from scratch necessitates a significant volume of data and computational resources, rendering it impractical for most users. To circumvent this, we typically rely on pre-trained embedding packages such as Glove or word2vec. When employed in this manner, embeddings can be regarded as a form of transfer learning. However, due to the limited vocabulary and minimal syntactical variations in our project's dataset, we opted to utilize the Keras machine learning framework to train the embeddings directly.

Encoder & Decoder:

In our sequence-to-sequence model, two recurrent networks are linked: an encoder and a decoder, interconnected through a recurrent network. The encoder serves the purpose of summarizing the input and preserving it within a context variable, often referred to as a state variable. Subsequently, the decoding context is derived, enabling the generation of the output sequence.

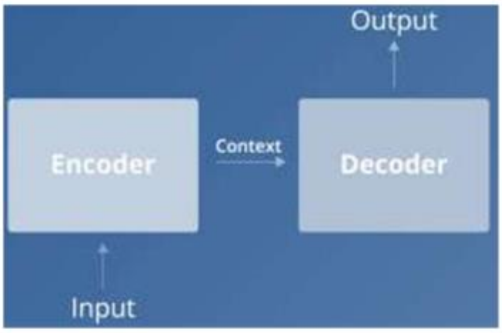

Fig D. Encoder & Decoder

Due to the recurrent nature inherent in both the encoder and decoder, loops are integrated into their structures to handle specific segments of the sequence at various temporal points. To gain a comprehensive understanding of this phenomenon, it is most effective to unroll the network and analyse the operations occurring at each time step. Illustratively, encoding the entire input sequence necessitates four steps, during each of which the encoder "reads" the input word, applying a transformation to the hidden state linked with that input word. Subsequently, it advances to the subsequent time step, carrying the updated hidden state forward. It's crucial to emphasize that the hidden state encapsulates the relevant context flowing through the network. While a larger hidden state augments the model's learning capacity, it concurrently raises the computational demands.

Our proposed system follows a design similar to that of the Moses SMT system, allowing us to compare phrase scores obtained from the Moses SMT system's phrase table. It operates as an RNN that incrementally processes each symbol in an input sequence referred to as the encoder. As the RNN processes each symbol, the hidden state of the RNN evolves based on Equation (1). After processing the entire input sequence, the RNN's hidden state becomes a summary, denoted as "c," of the entire input sequence, as depicted in the following figure. With knowledge of the hidden state "h," the decoder in our proposed model is another RNN trained to generate the output sequence by predicting the next symbol "yt" when the state is known at time "t." The calculation of the decoder's hidden state at time "t" is determined using a specific formula.

$$h_{(t)} = f(h_{(t-1)}, y_{t-1}, c)$$

And similarly, the conditional distribution of the next symbol is

$$P(y_t|y_{t-1}, y_{t-2}, \ldots, y_1, c) = g(h_{(t)}, y_{t-1}, c)$$

For given activation functions f and g (the latter must produce valid probabilities, e.g. with a SoftMax).

$$\max \frac{1}{N} \sum_{n-1}^{N} \log p\Theta(y_n|x_n)$$

Both components of the suggested RNN encoder-decoder are trained together with the aim of maximizing the conditional log-likelihood.

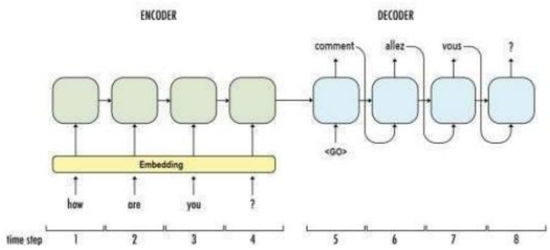

Fig E. RNN encoder-decoder

As both the encoder and decoder are recurrent, they each have loops that handle various parts of the sequence at different time steps. To gain a clearer understanding of this, we can unroll the network and analyse the operations at each time step.

Hidden Layer with Gated Recurrent Unit (GRU)

Determining which information from the hidden state should traverse through the network is a crucial concept. Not all information holds equal relevance, and there are instances where it becomes necessary to discard specific information entirely. Addressing this concern, the Gated Recurrent Unit (GRU) presents a solution. It comprises two pivotal components: the Update Gate (z) and the Reset Gate (r), working collaboratively to regulate the flow of information.

Fundamentally, the update gate (z) assumes the responsibility of determining the quantity of information to be transmitted from the previous time step into the subsequent one. Conversely, the reset gate (r) influences the degree to which preceding information should be reset or erased, thereby serving as a potent mechanism for governing the information flow within the network. For a more extensive elucidation, you can consult the essay by Simeon Kostadinov referenced earlier.

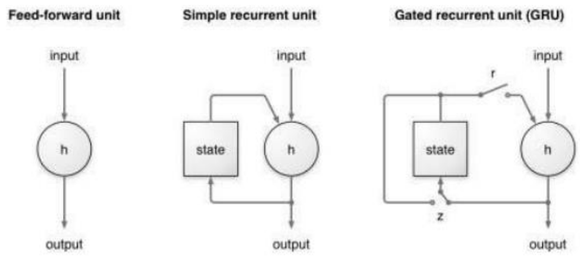

Fig F. Gated Repetition Unit

4. Result Analysis

Evaluating the outcomes of a research study is essential in determining its effectiveness. Nevertheless, prevailing evaluation methodologies frequently concentrate on metrics like precision, recall, time lag, and the volume of queries. Although these metrics hold significance, they occasionally result in evaluations lacking in structure and depth. Hence, there is an increasing demand to establish fresh parameters and approaches to comprehensively assess research outcomes. In this investigation, we employed the Ahirani training corpus, extracted from the Ahirani dictionary available at (https://www.ahirani.in/en/books/ahirani-dictionary). This corpus comprises parallel source-target sentence pairs, with English serving as the source language. Our machine translation system underwent training using this dataset, finding applications across diverse scenarios. However, the utilization of this corpus presented certain constraints, mainly associated with its format. To ensure a fruitful training session, we had to meticulously preprocess the data. Furthermore, validation data, constituting 80% of instances and derived from the training corpus, was utilized to confirm the training convergence of our model.

To assess the translation efficacy of the trained and validated model, we employed a test corpus containing around 1400 English sentences. The implementation process involved fine-tuning various parameters, as detailed in Table 1 below. These parameters played a pivotal role in determining the success of our model.

Table 1.1: RNN Algorithm Tunning Parameters

Number of Epochs	5
RNN Size	128
Batch Size	128
Epoch	2
Learning Rate	0.01
Encoding Embedding Size	128
Decoding Embedding Size	128
Keep Probability	0.8

The training and validation accuracy of the RNN machine translation model was compared against the number of epochs to assess its performance. Table 2 provides a summary of the training and validation accuracy at different epochs, and Figure C visualizes this analysis. Evaluating accuracy over different epochs allows us to understand how the model's performance evolves with training.

Table 2.1: Number of Epoch Vs Training and Validation Accuracy

Epoch	Train Accuracy	Validation Accuracy
1	67.31	67.05
2	76.13	74.36
3	81.29	80.65
4	84.45	83.38
5	87.50	85.65

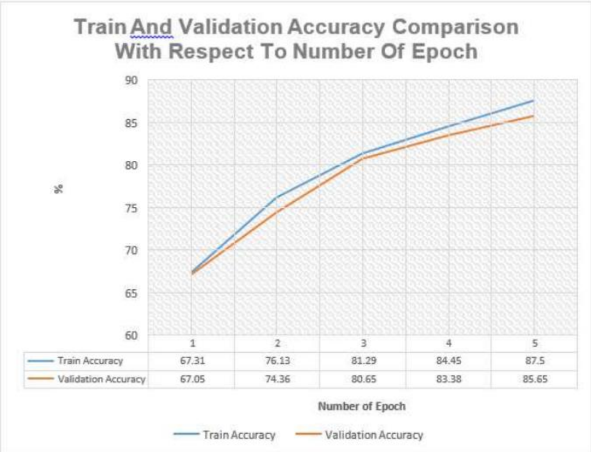

Fig G. Training and Validation Accuracy for RNN Machine Translation Model

The losses occurred during the execution is shown in table 3 and represented in figure D.

Table 3.1. Epochs Vs Training Losses

Epoch	Training Loss
1	0.4356
2	0.3316
3	0.2497
4	0.2070
5	0.1514

Fig H. Training Losses for number of Epochs

5. Conclusion Remarks

In conclusion, this research presents a Neural Machine Translation (NMT) system based on Recurrent Neural Networks (RNN) for predicting translations from English to Ahirani. The study highlights several key advantages of employing NMT in the Indic context, including the system's ability to generate fluent translations, enhanced contextual analysis capabilities, and superior performance compared to Statistical Machine Translation (SMT) systems. These advantages served as strong motivations for investigating NMT in the context of English to Ahirani translation.

The research team conducted comprehensive experiments, exploring various factors such as the number of epochs, the volume of training data, and the length of test phrases to evaluate the translation efficiency of the English-Ahirani MT system. Detailed examination of predicted translations revealed that the MT system consistently produced fluent translations, and its performance improved with larger training datasets and longer test phrases. The analysis of translation performance against the epoch diagram helped in assessing the convergence of the system training.

It's important to note that the effectiveness of NMT is highly reliant on the size of the training corpus. Therefore, increasing the number of training corpus instances is crucial to enhancing the effectiveness of the training data. Additionally, the research emphasizes the significant impact of attention mechanisms and scoring functions on translation success. By modifying the merit function, researchers can enhance the interaction between the source state vector and the current hidden state of the decoder, leading to improved translations.

Furthermore, the research underscores the importance of skillfully selecting system parameters such as the number of epochs, hidden layers, and GPU utilization. These choices play a pivotal role in enhancing the overall quality of translated text.

In summary, this study contributes to the understanding and advancement of Neural Machine Translation for Indic languages like Ahirani. It demonstrates the potential of NMT in generating high-quality translations and emphasizes the need for careful consideration of various factors to optimize translation performance.

References

[1] Pramod Salunkhe, Mrunal Bewoor, Suhas Pati, Shashank Joshi, Aniket kadam, “Summarization and Hybrid Machine Translation System for English to Marathi: A Research Effort in Information Retrieval System (H-Machine Translation” ,Discovery The International journal (ISI thomas retuers indexed).2015.

[2] Pramod Salunkhe , Mrunal Bewoor , Dr.Suhas Patil “A

Research Work on English to Marathi Hybrid Translation System", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3) 2015, 2557-2560.s

[3] S. B. Kulkarni, P. D. Deshmukh "Linguistic Divergence Patterns in English to Marathi Translation International Journal of Computer Applications (09 75 8887) Volume 87 No.4, February 2014.

[4] J. Ramanand, Akshay Ukey, Brahm Kiran Singh, Pushapak Bhattacharyya "Mapping and Structural Analysis of MultilingualWordnets"

[5] Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.

[6] Sreelekha.S, Raj Dabre, Pushpak Bhattacharyya "Comparison of SMT and RBMT, The Requirement of Hybridization for Marathi Hindi MT" [online pdf].

[7] Amruta Godase, Sharvari Govilkar"Survey Of Machine Translation Development for Indian Regional Languages" International Journal of Modern Trends in Engineering and Research IJMTER- 2014.

[8] M.L.Dhore & S.X.Dixit (2011) "English to Devnagari Translation for UI Labels of Commercial web based Interactive Applications, International Journal of Computer Applications (0975 – 8887) Volume 35 – No.10, December 2011.

[9] Devika Pishartoy, Priya, Sayli Wandkar (2012) "Extending capabilities of English to Marathi machine Translator., IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012.

[10] Priyanka-choudhary "A approach for interlingua example based EM Translation, IJCSIT, 2015.

[11] Abhay A, Anita G, Paurnima T, Prajakta G (2013), "Rule based English to Marathi translation of Assertive sentence International Journal of Modern Trends in Engineering and Research 2014.

[12] Aniket Kadam, A.D. Dept. Inf. Tech., BVDUCOEP, Pune, India ; Joshi, S.D.; Medhane, S.P, "Question Answering Search engine short review and road-map to future QA Search Engine", Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 , 10.1109/EESCO.2015.7253949.

[13] Kadam Aniket, Prof.S.D.Joshi, prof.S.P.Medhane, "QAS" International Journal of Application or Innovation in Engineering & Management IJAIEM , Volume 3, Issue 5, May 2014 May 2015.

[14] Kadam Aniket, Prof.S.D.Joshi, Prof.S.P.Medhane, "QAS" International Journal of Application or Innovation in Engineering & Management IJAIEM , Volume 3, Issue 5, May 2014 May 2015.

[15] Kadam Aniket, Prof.S.D.Joshi, Prof.S.P.Medhane "Search Engines to QAS:Explorative Analysis", International Journal of Application or Innovation in Engineering & Management IJAIEM , Volume 3, Issue 5, May 2014 May 2015 IJAIEM May 2015 .

[16] Mishra, Sudha. "Heterogeneous Groups in Competition: Innovative Pedagogy for Teaching English as Second Language." International Journal of Educational Science and Research 4.2014 (2014): 1-6.

[17] Jha, Sanjay Kumar. "Exploring the Causes of Sanskrit's Decline (A Pilot Study)." International Journal of Educational Science and Research (IJESR) 8 (2018): 165-172.

[18] Kumar, Manoj, and Sanjay Kumar. "SPEECH AS TEXT FOR TEACHING: AN ANALYSIS OF LINGUISTIC AND STYLISTIC MANEUVERINGS OF STEVE JOBS" FAMOUS SPEECH STAY HUNGRY STAY FOOLISH AND EXERCISES FOR ENHANCING STUDENTS" LANGUAGE SKILLS." International Journal of English, and Literature (IJEL) 6.1 (2016): 47-58.