SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I5P117

Volume 11 Issue 5, 170-178, May 2024

Original Article

The English to Telugu CLIR for NER Using Bidirectional Encoding and Unidirectional Decoding Using Random Sampling and Beam Search

B. N. V. Narasimha Raju[1], K. V. V. Satyanarayana[2], M. S. V. S. Bhadri Raju[3]

[1,2]*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India.*
[3]*Department of Computer Science and Engineering, SRKR Engineering College, Andhra Pradesh, India.*

[1]*Corresponding Author : buddaraju.narasimharaju@gmail.com*

Received: 20 March 2024 Revised: 23 April 2024 Accepted: 15 May 2024 Published: 31 May 2024

Abstract *- Neural Machine Translation (NMT) systems and the availability of a wide variety of linguistic resources have greatly improved Cross-Lingual Information Retrieval (CLIR) capabilities. When translating English queries into Indian languages, the NMT approach performs well. The NMT will employ a parallel corpus for translations. The translation of English queries into Telugu is the main emphasis of this study. A lack of Telugu-language content makes it challenging to have a sizable parallel corpus. Consequently, NMT encounters issues with Out-Of-Vocabulary (OOV) and Named Entity Recognition (NER). Byte Pair Encoding (BPE) attempts to translate unusual words by breaking them down into subwords in order to overcome the OOV problem. Problems such as NER still have an effect. The system may be trained in both forward and reverse directions to recognize NER effectively. The system is trained to recognize named entities in both directions through bidirectional encoding. Consequently, NER issues can be solved with Bidirectional Long Short-Term Memory (BiLSTM) encoding. Random sampling and beam search decoding with unidirectional LSTM are used to improve the translation output sequence. The approach using BPE and BiLSTM encoding, along with random sampling and beam search decoding with unidirectional LSTM, will help to resolve the OOV and NER problems and improve the output sequence of the translations generated by the NMT system. This approach is evaluated by using the Bilingual Evaluation Understudy (BLEU) score and other metrics like accuracy, perplexity, and cross-entropy, demonstrating that the translation quality of NMT with bidirectional encoding and unidirectional decoding using random sampling and beam search surpasses that of regular encoding and decoding models using LSTM.*

Keywords *- Cross lingual information retrieval, Machine translation, BiLSTM, Random sampling, Beam search.*

1. Introduction

CLIR brings data from a database apart from the user's query language. CLIR allows users to retrieve information in languages they do understand. Language barriers are eliminated, and consumers may now access a far wider range of information. A researcher might utilize CLIR, for instance, to locate studies written in a language they are not familiar with. When users need data in different languages, it helps them find relevant content in other languages. This method translates English questions into Telugu queries. These translations are generated using Machine Translation (MT) techniques. Studies on native content are gaining popularity in countries such as India, where a substantial section of the population still struggles with English. Thus, the CLIR is critical for people who rely on regional content. A 2017 KPMG analysis [1] expects an 18% annual increase in the number of Indians utilizing the internet in their native language.

One of the better translation methods is machine translation, which excels in terms of both speed and volume. It is affordable for basic tasks and translates a ton of information quickly. The majority of MT is done using corpus-based translation. When compared to more conventional Statistical Machine Translation (SMT) techniques, NMT is regarded as the greatest machine translation technology because of its capacity to comprehend complex linguistic patterns and context, producing translations that are more accurate and fluent. NMT is the most often used category in corpus-based translation. This model requires a parallel corpus [2] to be trained. Neural network models and machine learning are both used by the NMT. NMT models are very efficient for a variety of translation jobs since they can handle different language pairs and domains with ease and still produce translations that appear natural.

There are generally multiple stages involved in Neural Machine Translation (NMT). During the data collection phase, parallel corpora are gathered. The NMT model is trained using these corpora. For example, there are a few parallel corpora for Telugu and English. Telugu's rich morphology explains the prevalence of noise and discrepancies in these kinds of data sets. The gathered data is cleaned up during the preprocessing phase. The main stage of NMT, known as encoding, is word-by-word processing while training a neural network model.

An encoder network receives each word after it has been translated into a numerical representation. The encoded representation is obtained by the decoder phase from the encoder. Next, using the source representation and the previously created words in the target language, it predicts the target sentence word-by-word. This process keeps going until a unique "end of sentence" token is produced. The model's performance is assessed on a different validation set following decoding in the evaluation phase in order to check the quality of the translation and pinpoint areas that require improvement. Metrics like BLEU are frequently employed in assessment.

In NMT, OOV problems arise when the model comes across tokens during inference that were absent from the training set. This may make it difficult to translate these OOV tokens precisely. This is how OOV problems appear in NMT. Rare terms, proper nouns, or domain-specific terminology that was sporadically encountered in the training set of data may prove difficult for NMT models to translate. As a result, the output may contain incorrect or untranslated tokens. One way to address OOV concerns [3] is to supplement the training data with extra parallel corpora that contain specialist terminology or a wider vocabulary. As an alternative, methods like word segmentation can divide uncommon or unknown words into more manageable subword units for the model. OOV issues [4, 5] are more noticeable in language pairs with less training data or low-resource languages, where the model might have trouble making meaningful generalizations. The English-Telugu parallel corpus is also resource-poor so the system will face issues with OOV words. These OOV issues can be addressed by using word segmentation like BPE.

Numerous obstacles may lead to NER problems in NMT. Insufficient context in NMT models can make it difficult for them to identify named items correctly, particularly if they use the context that the source language sentence provides. This may result in mistranslated or incorrectly preserved named entities. Named entities frequently display ambiguity, in which a single word or phrase can refer to several different entities or represent different things depending on the situation. It may be difficult for NMT models to distinguish named entities accurately. Errors or omissions in NER [6, 7] may result from NMT models encountering named entities during inference that were absent or inadequately represented in the training data. Accurately maintaining NER annotations between source and target languages presents extra hurdles for NMT models when translating text containing named entities between languages. So, BiLSTM will be helpful in recognizing the named entities.

To translate a sentence from the source language into the target language, NMT employs decoding, an essential step. Word by word, the desired sentence is constructed. The next word is predicted by considering the encoded source text and previously generated target words. A decoder [8, 9] should consider the wider context of the sentence being constructed, guaranteeing that the translated sentence is grammatically accurate, flows naturally, and expresses the meaning that was intended. The quality of the decoding process determines the quality of the final translation because poor choices can make the entire sentence awkward. So, the decoder also plays a vital role in generating better translation accuracy.

2. Related Work

NER to recognize and categorize significant items referenced in the text. These things fall into a number of categories, such as names of people, establishments, locations, and so on. NER essentially assists computers in understanding the actual objects described in the text. In language processing, OOV refers to words or phrases that a system has not encountered in its training.

NER technology has developed and is now widely used using models customized to domain-specific difficulties and entity types in chemistry, food safety, and healthcare, among other disciplines. The advancement of named entity recognition technology is discussed in this study by Xing Liu et al. [10], along with its significance for information extraction and its uses in the future. To improve chemical information extraction operations, Taketomo Isazawa et al. [11] propose a single model that works for both organic and inorganic chemical named entity identification tasks. The difficulty of inaccurately segmenting entity boundaries resulting from the lack of separators between Chinese characters is addressed by Cheng-Yen Lee et al. [12], which focuses on named entity recognition in the Chinese medical sector.

To improve entity name representations, Yi Zhou et al. [13] suggest a Chinese NER model called LEMON, which integrates word and character-level information. Prefix and suffix, two position-dependent features, are incorporated into the model to improve entity name classification. These qualities are produced by lexicon-based memory, which also handles words that are not in the vocabulary. The efficacy of the model was demonstrated by a rise in the F1 score on four popular NER datasets.

Growing accessibility to textual information is fuelling the development of NER systems. Restrictions have been ignored by conventional approaches in favour of changing NER models. F. Zhao et al. [14] restructure popular systems under unconstrained tagging schemes and suggest a dynamic entity-based NER solution. An independent word and entity tagging method and a consistent word and entity labelling system are the two new, unrestricted schemes that are put forth.

The models dynamically handle inputs to ensure that entity-level attributes are incorporated. Tests conducted on datasets in English, German, Dutch, and Spanish demonstrate that the approaches work effectively in linguistic boundaries. An open definition of multilingual named entity definition is made possible by Y. Luo et al. [15] on named entity distribution in a wide word embedding space.

In contrast to earlier, closed, and restricted definitions, this model makes use of a particular geometric structure known as the named entity hypersphere. Mapping the model offers a novel technique to construct a named entity dataset in situations when language resources are scarce; for scenarios involving only one language, the model offers an open description of several named entity types and languages. In general, the suggested paradigm can be used to improve the most advanced named entity recognition systems.

The OOV terms are also frequently used in languages with limited resources, and managing them will improve the quality of translation produced by NMT systems. When working with OOV words, word embeddings can only be used to a limited extent in natural language processing tasks. Methods for interpreting their meaning based on context and morphological structure have been proposed by Zhongyu Zhuang et al. [4]. On the other hand, learning is challenging due to the low frequency of OOV words, and context scarcity is an issue. In order to tackle this issue, the notion of "similar contexts" is presented, drawing from the "distributed hypothesis" found in linguistics and human mechanisms involved in reading comprehension. According to the experimental data, the similar contexts model achieves higher relative scores in both intrinsic and extrinsic assessment tasks, which enhances OOV word embedding learning.

Decoding is a critical stage in machine translation, where the system converts the source text into the destination language using the parameters it has learned from the model. The NER will be recognized with the help of the BiLSTM in encoding. However, conventional unidirectional source-to-target architectures struggle to produce a language-independent representation of text since they rely on specific language pairs. Boyuan Pan et al. [16] propose a Bi-Decoder Augmented Network (BiDAN) for NMT applications. In order to generate the source language sequence during training, BiDAN has an additional decoder. Language-independent semantic space can be produced using this shared encoder. Experiments on multiple NMT benchmark datasets illustrate the usefulness of the proposed method. Therefore, for better translations of the source text, a method that takes into account OOV, NER, and decoding is needed. Analogous research has been conducted on these types of challenges.

3. BPE and Bidirectional Encoding

The NMT approach proposed has preprocessing for cleaning the data set, BPE for handling the OOV words in the data set, BiLSTM encoding to train the system in both directions for recognizing the NER, and unidirectional LSTM with random sampling and beam search to generate a better translation and output sequence. Figure 1 illustrates the architecture for this approach.

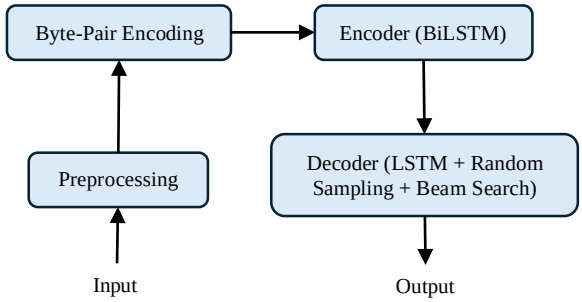

Fig. 1 Architecture for BPE and BiLSTM

3.1. Preprocessing

Preprocessing in NMT is the action taken to get the input data ready before feeding it into the neural network model. For the model to train well and generate correct translations, effective preprocessing is essential. Several data-cleaning procedures may be used during preprocessing to eliminate noise, mistakes, or unnecessary information from the incoming text. This could entail eliminating special characters or uncommon or unnecessary tokens. Cleaning the data increases the model's capacity for generalization and helps it concentrate on discovering significant patterns.

3.2. BPE to Address OOV Words

To address the issues caused by OOV [17, 18], the BPE mechanism is used. NMT uses BPE [19] as a subword tokenization approach. Managing OOV words that the model has not encountered during training is one of the main challenges it addresses in NMT.

BPE Mechanism:
Step 1: Input is the collection of strings is S and the target vocabulary size is n.
Step 2: C is the set of unique characters, and C ϵ S

Step 3: Repeat step 3 when |C| < n.
   a. l and m are the most frequent consecutive bigrams, and l, m ϵ S
   b. Replace l and m by using z.
   c. Now update C by adding the z.
   d. In S, replace each l, m instance by z.
Step 4: Return C.

One effective method for addressing the OOV issue in NMT is byte-pair encoding. BPE divides words into smaller pieces called subwords, as opposed to standard vocabulary, which employs complete words as tokens. If they occur frequently enough, these subwords may consist of single characters, character combinations, or even entire words. The training data's preexisting vocabulary serves as the basis for BPE. The most frequent pair of characters or subwords that follow one another in the text is then determined iteratively. Then, a new single unit is created by combining these frequent pairs. This merging procedure keeps going until the target vocabulary size is reached or for a predetermined number of iterations. In BPE, a word can be represented by a combination of the learned subwords, so the NMT system can still translate it even if the entire term is not in the vocabulary. Consider the scenario where "highest" is OOV. BPE can translate "highest" as the combination of the subwords "high" and "est" if it has acquired these subwords during training.

3.3. Bidirectional Encoder

The process of transforming a sentence in the source language into a fixed-length vector representation that conveys its semantic meaning is called encoding. Because it reduces all the information from the source text into a format that the neural network can use effectively, this encoding is essential.

Bidirectional encoding [20] is significant in NMT; it enables the model to consider both leftward and rightward contexts while generating translations. When encoding a specific token, the encoder can only access information from previous tokens in the sequence since traditional unidirectional encoding processes input text sequentially. Bidirectional encoding, on the other hand, enables the encoder to obtain context from both sides, resulting in a more comprehensive representation of the input sentence. It allows the model to encode a token by capturing both the words that come before and after it. This aids in the model's improved context understanding. By giving the model a broader perspective of the input sentence, it enables it to pick up richer and more detailed representations. Natural language is frequently ambiguous, with words and sentences having several context-dependent meanings. By considering context from both directions, bidirectional encoding lessens the possibility of mistranslations and aids in the model's ability to disambiguate such occurrences. For instance, consider the following two sentences:

- I like Paris very much because it is my favourite tourist place.
- I like Paris very much because he is my best friend.

There is a token known as Paris in both phrases. The leftward context in a unidirectional LSTM model specifies the token that appears next. Accordingly, it might not be possible to tell if the term Paris is the name of the individual or the location. Using both the leftward and rightward contexts, a BiLSTM is able to recognize the tokens in a sentence correctly. Two unidirectional LSTM layers coupled in opposite directions serve as the encoder in BiLSTM. The input values for the forward LSTM are a_1, a_2,..., a_n while a_n, a_{n-1},..., a_1 is the input for the backward LSTM. Two LSTM model outputs can be combined to create the desired result. Figure 2 displays the architectural layout of the BiLTSM.

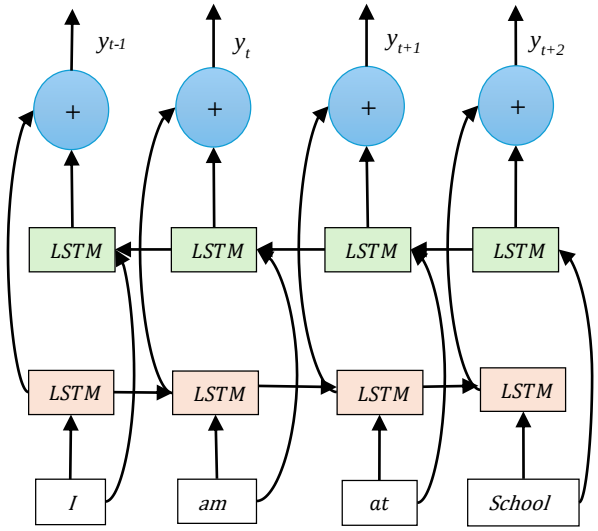

Fig. 2 Architectural design of BiLTSM

3.3.1. Bidirectional Encoding

Step 1: Input to the encoder - In encoding, the forward LSTM takes the input sentence and passes through the following gates, which control the flow of information within the cell and generate the hidden state h_t at each time step t. These hidden states capture the semantics of the input.

a. The input gate n_t decides how much of the new input should be permitted into the cell state, as in equation (1).

$$N_t = \sigma(W_{xn}x_t + W_{hn}h_{t-1} + W_{sn}s_{t-1} + b_n) \quad (1)$$

b. The forget gate r_t decides how much of the prior cell state should be kept, as in equation (2).

$$R_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + W_{sr}s_{t-1} + b_r) \quad (2)$$

c. The output gate m_t determines how much of the cell state should be disclosed as output, as in equation (3).

$$M_t = \sigma(W_{xm}x_t + W_{hm}h_{t-1} + W_{sm}s_{t-1} + b_m) \quad (3)$$

d. The new candidate values that could be added to the cell state are represented by the candidate cell state. $\widetilde{s_t}$, as in equation (4).

$$\widetilde{s_t} = tanh(W_{xs}x_t + W_{hs}h_{t-1} + b_s) \quad (4)$$

e. The cell state update s_t mixes the new candidate values scaled by the input gate and forget gate with the old cell states$_{t-1}$, as in equation (5)

$$s_t = (r_t \odot s_{t-1}) + (i_t \odot \widetilde{s_t}). \quad (5)$$

f. The candidate cell state h_t is the hidden or output state at the time step *t*, as in equation (6).

$$H_t = m_t \odot tanh(s_t) \quad (6)$$

Step 2: The backward LSTM – The equations are like the forward pass but operate in the reverse direction, processing the input sequence from right to left.

Step 3: The output of the BiLSTM – is often expressed as $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$, which is a concatenation of the forward and backward hidden states. A BiLSTM encoder stops when it has processed the entire input sequence.

Step 4: Output generation by the decoder – During decoding, the decoder LSTM generates one output token at a time. The previous output token, the previous hidden state, and maybe some context data are sent by the encoder to the decoder. Until an end-of-sequence token is generated or the maximum length is achieved, these steps are repeated.

In this way, bidirectional encoding and corresponding decoding are performed for the source text. In this way, the encoding is performed to recognize the NER effectively.

3.4. Unidirectional Decoding with Random Sampling and Beam Search

The standard method in NMT for producing the target language sentence is unidirectional decoding [21]. One by one, the target words are predicted by the decoder. It only takes into account the already-created target words and the encoded representation of the entire source text.

Unidirectional Decoding:

Step 1: First, the entire input sentence in the source language is processed by an encoder to create a series of hidden states and a final context vector that captures the semantic essence of the input sentence.

Step 2: The decoder is initialized with the context vector from the encoder. This vector often serves to initialize the decoder's hidden state. The decoder generates words one at a time, beginning with a unique start-of-sequence token. For each step t, the decoder updates its state based on the previous state and the last generated word, as in equation (7).

$$H_t = DecoderRNN(y_{t-1}, h_{t-1}) \quad (7)$$

Step 3: Based on the current state h_t, the decoder determines the next word y_t at each time step, as shown in equation 8.

$$P(y_t \mid y_1, \ldots, y_{t-1}) = softmax(Wh_t + b) \quad (8)$$

Where W and b are weights and bias of the output layer.

Step 4: This process repeats until the decoder produces an end-of-sequence token, signalling the completion of the sentence.

In this case, the unidirectional decoder will perform the translation of the source text. Now, for better output sequence and translation quality, the Random Sampling (RS) and Beam Search (BS) mechanisms are utilized. NMT uses decoding techniques like BS [22] and RS [23] to choose the word sequence that is output. These methods aim to provide translations of the highest quality by navigating a language model in a complex probability environment. By selecting words based on their probability distribution, RS adds stochasticity to the word selection process. A heuristic search strategy called BS expands the most promising nodes in a small collection called the beam to examine a graph.

Random Sampling for Top *k*

Inputs: The encoded source sentence, vocabulary, and the number of top k probabilities to consider for random sampling.

Outputs: A decoded sentence in the target language.

Steps:

Step 1: Model Prediction - Based on the present context, the NMT model provides a probability distribution across the complete vocabulary for the next word in the translation.

Step 2: Top-k Selection - All words in the vocabulary are not taken into consideration; just the top k words with the highest probabilities are kept. A portion of the total vocabulary is made up of these top k terms.

Step 3: Random Sampling - Based on their probabilities, one word is randomly selected from this subset of the top k words. This adds randomness to the decoding process.

Step 4: Sequence *Generation* - The translated sequence is supplemented with the sampled word. The chosen term updates the context of the model. Steps 1-3 are performed until an end-of-sentence token is generated or the maximum sequence length is reached.

Beam Search:

Go through steps 1 through 4 repeatedly, even when the initial state *I* is not equal to empty.

Step 1 - The most optimal node is removed from I and it is denoted as o.

Step 2 - If the desired state is represented by o, go backwards until it is reached, then transmit the path.

Step 3 - Create and assess o successors, add them to I, and provide a parent list.

Step 4 - When the value of | I | exceeds w, which is the beam's width, the best *w* nodes are chosen, and the other nodes are excluded from *I*.

In this way, both the random sampling for top-*k* and beam search algorithms will work. The NMT system combines BS and RS to maximize the advantages of both strategies and generate a wide range of effective translations. RS adds diversity by considering a larger set of word options at each stage, and BS makes sure that the translated text is coherent and fluid. In addition to RS, BS concentrates on identifying high-quality translations by evaluating several theories concurrently and choosing the most likely ones. Combining BS with RS allows you to design a decoding method that produces a wide range of effective translations that are appropriate for different NMT systems.

Metrics, including the BLEU score and other parameters, are used to assess the approach. Accuracy measures the degree of accurate classifications. Conversely, the level of ambiguity signifies the certainty with which the probability model forecasts a sample. Finding the loss function is the goal of cross-entropy. The BLEU score is employed to evaluate the prediction accuracy.

4. Results and Discussion

In the English-Telugu parallel dataset, noise and replications are eliminated during the preprocessing stage. The system might become confused while learning new features from the replicated corpus, which could lead to overfitting of the model with less translation performance. If all these problems have been resolved in the parallel datasets, the parallel corpus will yield translations of higher quality. Therefore, by preprocessing the corpus, the NMT system may produce accurate translations of the original sentence.

Duplicate, noisy, and inconsistent data are removed from the parallel corpus by preprocessing. A comparison is made between the NMT approaches like Unidirectional encoding with RS and BS Decoding (LSTM+RS+BS), Bidirectional encoding with RS and BS Decoding (BiLSTM+RS+BS), Unidirectional encoding along BPE with RS and BS decoding (BPE+LSTM+RS+BS), Bidirectional encoding along BPE with RS and BS decoding (BPE+BiLSTM+RS+BS). The parallel corpus in Telugu and English serves as the input for all models. Unlike the dual-layer architecture seen in bidirectional encoding, the unidirectional encoding model has only one layer. BiLSTM will continue to have a one-layer decoding mechanism and a two-layer encoding method. In both encoding and decoding, LSTM units are utilized. The models will feature an LSTM layer with a 500-size, a 0.01 learning rate, and Adam as the optimizer. With the model decay and dropout rates set at 0.5 and 0.3, respectively, 25000 training steps will be performed. Metrics like accuracy, perplexity, and cross-entropy are used to evaluate the training and validation performance of all the models. Table 1 displays the values for these parameters. Bidirectional encoding along the BPE with RS and BS decoding improves the NMT's performance.

Table 1. Evaluation of the performance of various NMT systems

Parameters	**LSTM + RS + BS**	**BiLSTM + RS + BS**	**BPE+ LSTM + RS + BS**	**BPE+ BiLSTM + RS + BS**
Training Accuracy	97.69	97.41	97.47	98.11
Training Perplexity	1.08	1.09	1.09	1.06
Training Cross-Entropy	0.08	0.09	0.09	0.06
Validation Accuracy	54.81	55.59	54.60	55.41
Validation Perplexity	204.54	190.51	204.73	190.41
Validation Cross-Entropy	5.32	5.25	5.32	5.24

The training graphs for all the NMT systems are shown in Figure 3. The graph in Figures 3(a), 3(b), and 3(c) show how training accuracy is compared, perplexity is shown and cross-entropy comparisons between various NMT systems. The training accuracy of bidirectional encoding along BPE with RS and BS decoding is 98.11, which is higher. It is preferred that the accuracy rates show higher values. The training perplexity of bidirectional encoding along BPE with RS and BS decoding is 1.06, which is lower. The model that has a lower perplexity score is deemed to be superior. The training cross-entropy score of bidirectional encoding along BPE with RS and BS decoding is 0.065, which is lower. It is thought that the model with the lower cross-entropy score is better.

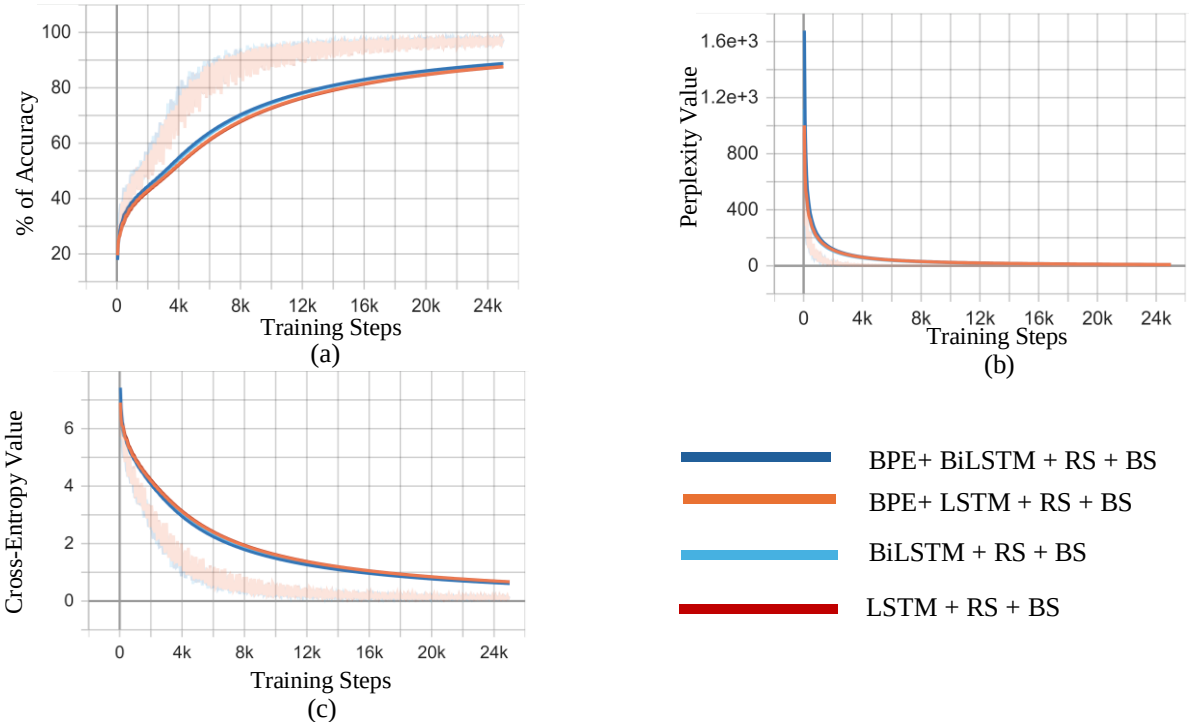

Fig. 3 The training graphs regarding (a) Accuracy, (b) Perplexity, and (c) Cross-entropy.

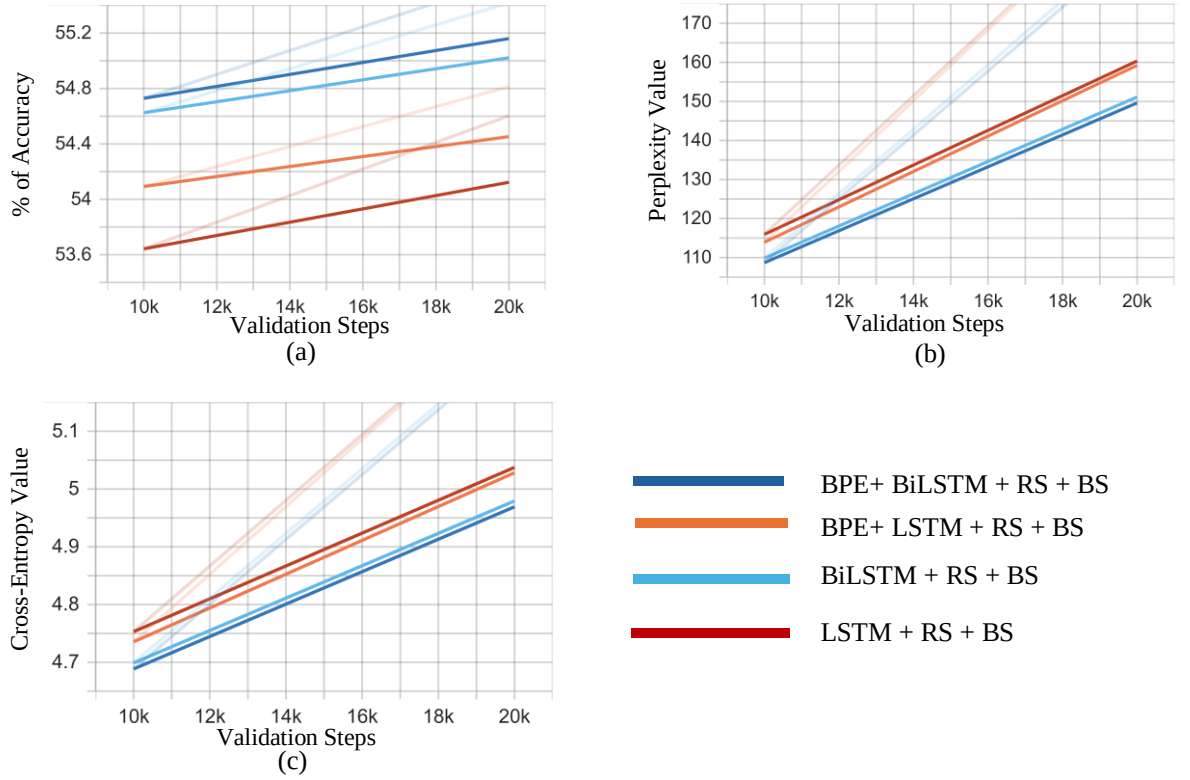

Fig. 4 The validation graphs regarding (a) Accuracy, (b) Perplexity, and (c) Cross-entropy.

The validation graphs for all the NMT systems are shown in Figure 4. The graphs in Figures 4(a), 4(b), and 4(c) show how validation accuracy is compared, perplexity is shown, and cross-entropy comparisons are made between various NMT systems. The validation accuracy of bidirectional encoding along BPE with RS and BS decoding is 55.41, which is higher. It is preferred that the accuracy rates show higher values. The training perplexity of bidirectional encoding along BPE with RS and BS decoding is 190.41, which is lower. It is thought that the model with the lower perplexity score is better. The training cross-entropy score of bidirectional encoding along BPE with RS and BS decoding is 5.24, which is lower. The model that has a low cross-entropy score is considered to be superior. These findings imply that bidirectional encoding along the BPE with RS and BS decoding performs better.

One measure used to evaluate the model's efficacy is the BLEU score. BLEU values are shown in Table 2. It is shown that the use of bidirectional encoding along BPE with RS and BS decoding in NMT yields translations with greater accuracy levels, as indicated by the BLEU score. Preprocessing, BPE, and BiLSTM models help the NMT system generate translations that are more accurate by removing noisy content in a parallel corpus and resolving OOV and NER issues.

Table 2. BLEU scores of NMT approaches

NMT Approaches	BLEU Score
LSTM + RS + BS	16.41
BiLSTM+ RS + BS	17.57
BPE + LSTM + RS + BS	16.64
BPE+ BiLSTM + RS + BS	18.13

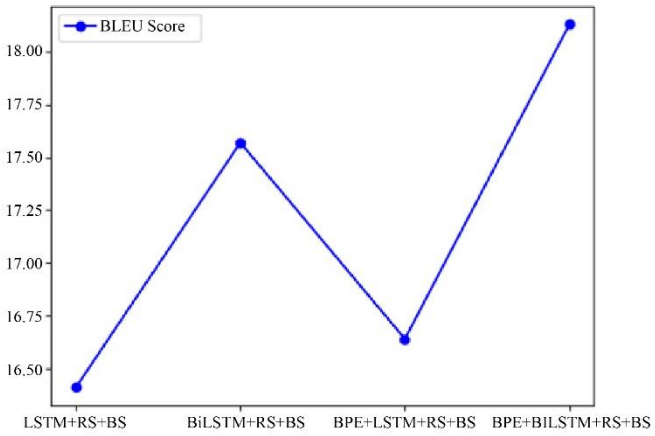

Fig. 5 BLEU score

Figure 5 compares the BLEU scores of all the NMT approaches. The bidirectional encoding along BPE with RS and BS decoding has a value of 18.13, which is the highest among all the NMT approaches. BLEU scores exist for all models. The better model is the one with a higher BLEU score. These results suggest that bidirectional encoding along BPE with RS and BS decoding performs better.

5. Conclusion

The NMT is essential to CLIR because it translates English queries into Indian languages like Telugu. If the translations can be generated by eliminating repetitions, noisy data, and inconsistencies in the corpus, then NMT will function more effectively. These problems are resolved when the parallel corpus is preprocessed. The English-Telugu corpus in the NMT system lacks resources, which contributes to OOV problems like unfamiliar terms in the corpus. BPE is used to tackle these OOV problems by breaking the words down into subword units and attempting a translation. Compared to the use of unidirectional LSTMs, the BiLSTM in the NMT has helped to alleviate some of the NER recognition difficulties. While unidirectional LSTM only uses leftward context, bidirectional LSTM uses both leftward and rightward context in the sentence to identify NER. BiLSTM thus outperforms unidirectional LSTM in terms of translation accuracy.

The preprocessing stage makes use of the English-Telugu parallel corpus, while the encoding stage uses the output as input. Metrics like the BLEU score and other parameters are used to evaluate the quality of translations. The BiLSTM encoding and BPE with unidirectional decoding along RS and BS have performed better when compared with different techniques for MT quality and accuracy. Thus, by adding BPE to the model, some OOV issues have been handled. When OOV problems are resolved, translations in languages with limited resources perform better. Rather than using unidirectional LSTM, the BiLSTM in the NMT has helped to alleviate some of the NER recognition problems.

During the decoding process, RS and BS will generate a translation of superior quality. So, for the parallel corpus of English and Telugu, the NMT that uses BiLSTM encoding and BPE with unidirectional decoding along RS and BS yields better translations. Therefore, it is advised to use this NMT system technique for the parallel Telugu and English corpus. Thus, the CLIR with BiLSTM encoding and BPE with unidirectional decoding along RS and BS are helpful in increasing translation accuracy.

References

[1] Varun Bora, Rahil Bassim, and Sakshi Rai, “Indian Languages - Defining India's Internet,” *Klynveld Peat Marwick Goerdeler*, pp. 1-36, 2017. [Google Scholar] [Publisher Link]

[2] Jianhui Pang et al., "Rethinking the Exploitation of Monolingual Data for Low-Resource Neural Machine Translation," *Computational Linguistics*, vol. 50, no. 1, pp. 25-47, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jonas Waldendorf et al., "Improving Translation of Out of Vocabulary Words Using Bilingual Lexicon Induction in Low-Resource Machine Translation," *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas*, Orlando, USA, vol. 1, pp. 144-156, 2022. [Google Scholar] [Publisher Link]

[4] Zhongyu Zhuang et al., "Out-of-Vocabulary Word Embedding Learning Based on Reading Comprehension Mechanism," *Natural Language Processing Journal*, vol. 5, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Johannes V. Lochter, Renato M. Silva, and Tiago A. Almeida, "Multi-Level Out-of-Vocabulary Words Handling Approach," *Knowledge-Based Systems*, vol. 251, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Hong Ming et al., "Few-Shot Nested Named Entity Recognition," *Knowledge-Based Systems*, vol. 293, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Zhaojian Cui et al, "Language Inference-Based Learning for Low-Resource Chinese Clinical Named Entity Recognition Using Language Model," *Journal of Biomedical Informatics*, vol. 149, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yan Huang, TianYuan Zhang, and Chun Xu, "Learning to Decode to Future Success for Multi-Modal Neural Machine Translation," *Journal of Engineering Research*, vol. 11, no. 2, pp. 1-7, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jinsong Su et al., "Exploiting Reverse Target-Side Contexts for Neural Machine Translation Via Asynchronous Bidirectional Decoding," *Artificial Intelligence*, vol. 277, pp. 1-14, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Xing Liu, Huiqin Chen, and Wangui Xia, "Overview of Named Entity Recognition," *Journal of Contemporary Educational Research*, vol. 6, no. 5, pp. 65-68, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Taketomo Isazawa, and Jacqueline M. Cole, "Single Model for Organic and Inorganic Chemical Named Entity Recognition in ChemDataExtractor," *Journal of Chemical Information and Modeling*, vol. 62, no. 5, pp. 1207-1213, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Cheng-Yen Lee et al., "Named Entity Recognition for Chinese Healthcare Applications," *2023 International Conference on Consumer Electronics - Taiwan*, PingTung, Taiwan, pp. 749-750, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Yi Zhou, Xiao-Qing Zheng, and Xuan-Jing Huang, "Chinese Named Entity Recognition Augmented with Lexicon Memory," *Journal of Computer Science and Technology*, vol. 38, no. 5, pp. 1021-1035, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Feng Zhao et al., "Dynamic Entity-Based Named Entity Recognition Under Unconstrained Tagging Schemes," *IEEE Transactions on Big Data*, vol. 8, no. 4, pp. 1059-1072, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ying Luo et al., "Open Named Entity Modeling From Embedding Distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5472-5483, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Boyuan Pan et al., "Bi-Decoder Augmented Network for Neural Machine Translation," *Neurocomputing*, vol. 387, pp. 188-194, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Aloka Fernando, and Surangika Ranathunga, "Data Augmentation to Address Out of Vocabulary Problem in Low Resource Sinhala English Neural Machine Translation," *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, Shanghai, China, pp. 61-70, 2021. [Google Scholar] [Publisher Link]

[18] Longtu Zhang, and Mamoru Komachi, "Neural Machine Translation of Logographic Language Using Sub-Character Level Information," *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels, Belgium, pp. 17-25, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural Machine Translation of Rare Words with Subword Units," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, vol. 1, pp. 1715-1725, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[20] Martin Sundermeyer et al., "Translation Modeling with Bidirectional Recurrent Neural Networks," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 14-25, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to Sequence Learning with Neural Networks," *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, Canada, vol. 2, pp. 3104-3112, 2014. [Google Scholar] [Publisher Link]

[22] Markus Freitag, and Yaser Al-Onaizan, "Beam Search Strategies for Neural Machine Translation," *Proceedings of the First Workshop on Neural Machine Translation*, Vancouver, pp. 56-60, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Improving Neural Machine Translation Models with Monolingual Data," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, vol. 1, pp. 86-96, 2016. [CrossRef] [Google Scholar] [Publisher Link]