Journal of Computer Science

Original Research Paper

Implementation of Single Candidate Loss Optimization Algorithm for Loss Optimization of Bhojpuri-English Machine Translation Model

[1]Rituraj Dixit, [1]Sarabjeet Singh Bedi, [2]Ibrahim Aljubayri and [2]Mohammad Zubair Khan

[1]Department of Computer Science and IT, MJP Rohilkhand University, Bareilly, India
[2]Department of Computer Science and Information Taibah University, Madinah, Saudi Arabia

Article history
Received: 11-09-2024
Revised: 25-11-2024
Accepted: 07-12-2024

Corresponding Author:
Rituraj Dixit
Department of Computer Science and IT, MJP Rohilkhand University, Bareilly, India
Email: dixit.rituraj@gmail.com

Abstract: Machine translation of low-resource Indian languages is necessary as most of the regions still know and speak their specific dialects and are still not comfortable understanding the English language. Indian languages are morphologically rich, due to which there are two big challenges, Ambiguity and Domain adaption, which are faced by researchers during the translation. Lack of data also increases the challenge for the researchers. In this study, we proposed a novel machine translation model that uses a single candidate optimization algorithm for loss optimization and have proved through results that it is more optimal than traditional gradient-based algorithms. We have used byte pair encoding for tokenization and then BERT is used for contextualized word embedding. The novelty is induced in our model as the traditional transformer model is used with a variation of loss optimization using a single candidate optimization technique during training to refrain from overfitting rather than traditional gradient-based techniques. The results have been compared with other state-of-the-art models and described in tabular form.

Keywords: Low Resource Language, Machine Translation, Byte Pair Encoding, Fine-Tuning, Loss Optimization

Introduction

Natural Language Processing (NLP) is a field of Artificial Intelligence (AI) that focuses on the interaction between computers and humans through natural language. The primary goal of NLP is to enable machines to understand, interpret and generate human language in a way that is both meaningful and contextually relevant. It has drawn attention after the development of machine learning. It is used to digitally characterize and understand language for communication. The artificial neural networks are used to build natural language processing models for different text processing applications which is shown in Fig. (1).

NLP is the study of how to utilize computers to comprehend and modify natural language for additional research and include computer techniques for the automatic analysis and interpretation of different types of human language (Satpute and Agrawal, 2023).

NLP is divided into two parts Natural Language Generation (NLG) and Natural Language Understanding (NLU). In NLU sentiment analysis, text summarization is two tasks and in NLG there is text translation.

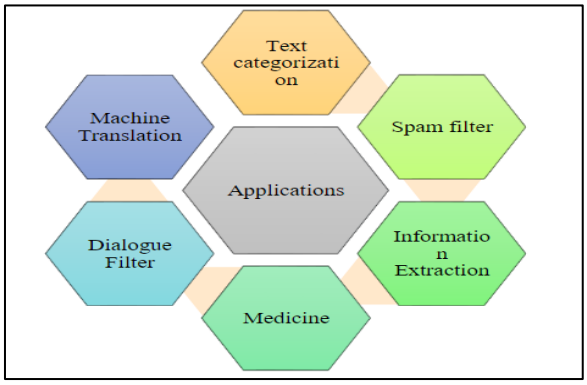

Fig. 1: Application areas of natural language processing (Satpute and Agrawal, 2023)

Traditional MT models have data-driven rules which were created by handwriting rules to map lexical and syntactic structure between two languages. The Statistical Machine Translation (SMT) models have rule-based feature extraction to build NMT where neural networks are used to extract the features directly from data. These models have a limitation in that they need a large corpus

to produce better results. In our proposed work there are two big challenges first, it is a low-resource language and second building a model that can achieve unambiguous translation from low low-resource language is challenging. Our research hypothesis is based on the paper by Philip *et al*. (2019) (Koehn and Knowles, 2017), after referring to the paper we came across ambiguity as one of the major challenges in the translation of low resource languages.

In our proposed model we have used the transformer model which has the following contributions:

1. For the elimination of Out of Vocabulary (OoV) words, training data is being tokenized and segmented using Byte Pair Encoding (BPE) as BPE has the advantage of using it in any corpus with even small units
2. Implementation of Single Candidate Optimization (SCO) for loss optimization of Loss function
3. Six-layer encoder decoder transformer model is implemented in which SCO optimization technique is used
4. It is being proved by experimental results that our model is better than traditional gradient-based loss optimization models like stochastic gradient descent etc.,

Literature Survey

Neural Network-based Machine Translation (NMT) models, which are a new paradigm as illustrated in Bahdanau *et al*. (2014); Bhaduri (1990); Sutskever *et al*. (2014); Vaswani *et al*. (2017), have achieved an improvement in translation performance and significantly reduced the quality gap between machine and human translations between some languages (Hassan *et al*., 2018; Tan *et al*., 2019). NMT frameworks that are available to perform research and development in this field are described in these papers (Junczys-Dowmunt *et al*., 2018; Klein *et al*., 2017; Ott *et al*., 2019; Senellart *et al*., 2018). Unlike statistical machine translation models, which are composed of multiple independently designed sub-components, Neural Machine Translation (NMT) employs a single neural network to directly learn the mapping between source sentences and their translations through an end-to-end training approach.

NMT models aims to learn a conditional language model or the likelihood of the target sentence given the source sentence can auto-decompose and represent millions of translation possibilities and is largely responsible building efficient translation models.

Phrase or rule-based MT, is the end-to-end learning technique of NMT, explicitly models the mapping from source to target language through a posterior probability (Bentivogli *et al*., 2016; Gaido *et al*., 2024). These models depend heavily on the availability of massive parallel data, building efficient NMT systems for languages with limited resources thus becomes a challenging task (Koehn and Knowles, 2017).

Bahdanau *et al*. (2014) have proposed attention models which have proved to be breakthrough in building NMT models and gives good results on sequential models. In this study the context vectors are being multiplied by the weighted score which depend upon the type of word going to be translated (Bahdanau *et al*., 2014).

Luong *et al*. have proposed another work on attention models where they have applied it on two levels the local level and the global level. The terminology local approach where a subset of words is taken into consideration when determining the weighted score for context vectors and in global approach, we have all source words to compute the attention score (Luong *et al*., 2015).

Tan *et al*. have proposed another variant of attention models known as hard and soft attention models which are being computationally less expensive than the previous attention models (Tan *et al*., 2019).

Vaswani *et al*. have developed transformers which have begun an era of parallel computing in NMT and it got adapted very quickly. The biggest advantage of transformers is it run of multiple heads and is not depended on sequential computation (Vaswani *et al*., 2017).

Devlin *et al*. (2018) have proposed Bidirectional Encoding Representation from Transformers (BERT) to generate quality embedding vectors. The BERT has two main components pretraining and fine tuning. In pretraining the models is trained using masking called Mask Language Models (MLM) and then in fine tuning neural networks are used to fine tune the parameters (Devlin *et al*., 2018). After BERT many more versions of BERT have come, we will analyze some of them and try to see their advantages. Devlin m-BERT which is popularly known as BERT is used for multilingual models which can support more than 100 languages.

Status of NMT Research work and Challenges in Indian languages

Hindi is the primary language spoken by the majority of Indians, which is followed by Tamil, Malayalam, Marathi, Telugu, Punjabi, etc. The majorities of people who reside in rural areas do not even speak or understand English (Haddow *et al*., 2022).

This section focuses on research work done on various major Indian languages and their results related to neural machine translation. Dialects of Indian languages due to morphological complexity and diversity makes MT is a difficult problem (Ott *et al*., 2019).

Machine Translation (MT) for Indian languages has already been investigated using rule-based or statistical methods. Statistical Machine Translation (SMT), which is phrase-based, is being replaced by Neural Machine Translation (NMT), which has demonstrated encouraging results for a number of language pairs.

Kunchukuttan *et al.* (., 2014) have proposed a paper which is based on phrase-based statistical machine translation system which has been tested on 110 different language pairs using Indian Language Corpora Initiative (ILCI). This was one of the largest exercises done in terms of both number of language pairs and corpus size (Nakazawa *et al*., 2020; Kunchukuttan *et al*., 2014).

Chakrawarti and Bansal have proposed a SMT model to analyze the ambiguity and translation divergence problem. In their work they have proposed seven module approach to solve the mentioned issue (Chakrawarti and Bansal 2017). They have employed lexicalized reordering and Moses for phrase extraction in statistical phrase-based machine translation for Indian languages but not able to address ambiguity removal solutions for low resource languages (Ghazal, 2015).

Cho *et al*. (2014) have proposed a model where phrase representations are learned using RNN encoder-decoder for statistical machine translation. In this study they two Recurrent Neural Networks (RNNs) makes up the RNN encoder-decoder architecture, a neural network model. In this model one RNN known as encoder converts a sequence of symbols into a fixed length vector representation and the other RNN called the Decoder, decodes the representation into a different sequence of symbols, (Cho *et al*., 2014).

Machine translation of Hindi-English (Hi-En) combination, which has a high resource data, has received a BLEU score of 56 (Gain *et al*., 2022). The results obtained from Hindi to English Translation act as baseline for many Indian languages translation. Indian languages are morphologically rich, making word organization in devnagri script a critical issue. B. Gain *et al*. (2022) have proposed a chat bot in Hindi to English which has achieved a BLEU score of (Gain *et al*., 2022).

Philip *et al.* (2019) have conducted an experiment on six languages and their related translations. In their work on translating six Indian languages to English and from English to six languages the maximum BLEU score of 22 has been achieved (Philip *et al*., 2019).

According to recent studies by Khan *et al*. (2017) the accuracy of machine learning translation models in translating other Indian languages (such as Bengali, Tamil, Punjabi, Urdu and Gujarati, Telugu, Kannada and Malayalam) is just 10%, aside from "hi-en" translation (Khan *et al*., 2017); Choudhary *et al*. have proposed an English to Tamil translation model with BLEU score of 8.33 (Choudhary *et al*. 2018).

Goyal and Sharma (2019) proposed a NMT system of IIIT-H for WMT19 evaluation. In their task they have used an attention model for Gujarati to English news translation and they have achieved a BLEU score of 9.8. The challenge which they have addressed is the low resource data (Dongare, 2024).

Deep *et al*. (2020) proposed a model that claims to provide a BLEU score of 38.30 for Punjabi to English and 36.96 for English to Punjabi translations. The Punjabi language is not as low resource corpus as we have for Bhojpuri or other languages and good efforts have been made in translating English to Punjabi (Deep *et al*., 2020). Singh *et al*. they have used powerful attention models which has given a BLEU score of 24.48 approximately for both type of translation (Singh *et al*., 2018).

A work by Mozammel Haque and Hasan have proposed a model, claiming that the sentences produced in translating English sentence to Bengali by their model are more semantically correct than the sentences produced by google translator (Haque and Hasan, 2018). There are very few studies done in English to Bengali language and many of them are not up to the mark. Their model has produced the accuracy of more than 97%, The real challenge lies in making correct sentences as Bengali is very complicated language, so some specific translating rules are proposed in this study.

Muhammad Aslam Sipra in his study on word borrowing for the English to Urdu translation, demonstrated that there are three ways to accomplish this task: Directly, with little or no change; using a translator to translate from English to Urdu; or thirdly, by combining Urdu and English. In his paper he has not given any specific information about any NMT technique or any methodology regarding embedding vector designing (Haque and Hasan, 2018).

Lingam *et al.* (2014) have proposed a rule-based approach to translate English to Telugu language translation mechanism this is the only work got found during the survey of researches dome in this particular domain (Lingam *et al*., 2014). There MT system claimed to give 92% of efficient translation and other sentences were partially correct. The gap here lies in implementation of more exotic model which is based on NMT concepts which can produce more semantically correct sentences.

Chethan *et al*. (2014) have implemented a MT system for English to Kannada, they have implemented a rule-based MT system which generates target language words from source language morphological information. The major challenges which were faced in implementing English-Kannada MT are the difference in word order of English to Kannada and second one is PNG (Pronoun, Noun and Gender) (Chethan *et al*., 2014).

Kunchukuttan *et al*. (2014) have worked on a project AI4Bharat-IndicNLP corpus in which they have created corpus and embedding vectors for NLP translation of Indian languages (Kunchukuttan *et al*., 2014).

Shukla *et al.* (2023) presented a schema that appraised google translator for translating Bhagavat Gita and Upanishads to Hindi. For this purpose, a manual created corpus was used for training. The schema consists of BERT based language model, then implemented semantic

and sentiment analysis. This analysis contains certain features of language translations such as metaphor, imaginary and contextual significance. Even though the quality evaluation of the model was found to be better, but without knowing the context there were more misinterpretations in the translation.

Kakwani *et al*. (2020) in their work have introduced number of unique datasets and language generation models, including Indic Corp, IndicNLG Suite, IndicGLUE, IndicXtreme (coming soon) and Naamapadam (coming soon). The 8.5 billion words in Indic Corp were compiled from monolingual corpora in 11 Indian languages (Kakwani *et al*., 2020).

For five different language generation tasks spanning 11 Indic languages, the IndicNLG suite includes training and evaluation datasets. One of the biggest collections of multilingual generating datasets exists here. The benchmark for six NLU tasks across 11 Indian languages is provided by IndicGLUE.

The quality of the translated text is being examined on some metrics like the BiLingual Evaluation Understudy (BLEU) score, a BLEU score between 0.6-1.0 is considered to be a good translated score. Metric for Evaluation of Translation with Explicit Ordering (METEOR) is a machine translation metric based on precision and recall it has a range of 0-1 where 1 is best. RIBES is another performance evaluation metric for automated machine translation text having a score in the range of 0-1. GLEU abbreviated as Google-BLEU score is another measure to estimate the quality of the translated text it is based on the concept of precision and recall and has a score range of 0-1.

In this literature survey we have found that very little work is being done on unexplored languages like the Bhojpuri language and the translation process which are available not able to resolve ambiguity, our proposed model has tried to resolve ambiguity and also produced better translation results when finetuned on existing models like Indian Bi-directional and Autoregressive Transformer (Indic-BART) (Dabre *et al*., 2022), Transformers (Vaswani *et al*., 2017), Indian Bidirectional Encoding Representation Techniques (Indic-BERT) which is a multilingual ALBERT Model a BERT class model. The comparative analysis of the above-stated models concerning our proposed model Single Candidate Loss Optimization (SCLO) model, is given in Table (1) for Bi-Lingual Transformer models and Table (2) for Multilingual Transformer models.

Table 1: Comparison of SCLO with bilingual and multilingual models

Model Name	Score (general)	Score (ambiguity)
SCLO Model	0.77(BLEU)	0.61((BLEU))
Indic-BART	0.65(BLEU)	0.51(BLEU)
Transformers	0.61(BLEU)	0.50(BLEU)
Indic-BERT	0.43(GLEU)	0.37(GLEU)

Proposed Model

In this proposed model Lexical Ambiguity problem has been resolved for English-Bhojpuri language translation. There are three components in this model, the Single Candidate Loss Optimization (SCLO) technique is used for loss optimization of the loss function and acts as an optimizer in the transformer translation model, the BERT model for contextualized word embedding and transformers for translation. After every epoch, we pass the updated single candidate loss optimization function and pass the values to the model for learning weight parameters. In this section we have two sub-sections in the first one we will discuss the concept of SCLO and analyze the important hyperparameters, in the second section architecture of the model is being discussed along with the implementation of SCLO in the model.

Single Candidate Loss Optimization Algorithm

Recent development in artificial intelligence and its related technologies like natural language processing in the last decade has made the real-world optimization problem more challenging and it gives motivation for the development of fast and efficient algorithms. In optimizing deep learning algorithms, we are relying on gradient based optimization algorithms which are prone to either vanishing gradient or exploding gradient problems. The proposed solution to this problem is to use single candidate solution-based algorithm. The Single Candidate Optimization (SCO) algorithm (Shami *et al*., 2024) is a single candidate-based optimization algorithm rather than most of the existing algorithms which rely on swarm of particles and we have modified the implementation of SCO for optimizing the loss function in machine translation.

SCLO is a two-phase algorithm combining two well-known meta-heuristic strategy to form single robust algorithm. The purpose of two-phase algorithm is to provide diversity and balance between exploration and exploitation.

The first phase is SCLO terminates when α function evaluations performs and second phase concludes after β evaluations where $T = \alpha + \beta$. In the first phase of SCLO candidate x_i updates its positions by using following set of equations:

$$x_i = \begin{cases} gbest_i + (w|gbest_i|) \; if \; r_1 < 0.5 \\ gbest_i + (w|gbest_i|) \; otherwise \end{cases} \quad (1)$$

where, r_1 is a random variable in the range [0,1] and w is given by:

$$w(t) = exp^{-\left(\frac{b_t}{T}\right)^b} \quad (2)$$

where, b is a constant t is the iteration number, T is the total number of iterations or function evaluations.

In the second phase of SCLO, deep greedy search is conducted to explore the space around the best solution and update its solution as per the following equations:

$$x_i = \begin{cases} gbest_i + ((r_2 w(ub_i - lb_i))) \; if r_2 < 0.5 \\ gbest_i - ((r_2 w(ub_i - lb_i))) \; otherwise \end{cases} \quad (3)$$

where, r_2 is another random variable in the range of [0,1], ub_i, lb_i are the upper bound and lower bound of in the domain of x_i , $i \epsilon D$, D denotes the dimension of the candidate vector. This domain iteratively keeps on increasing as number of iterations keep on increasing. In the implementation of SCO for loss optimization of the translation model *w* plays a very important role it is responsible to create a balance between exploration and exploitations. The high value of *w* in the beginning help in searching the space effectively while low values of *w* can be useful in exploitation. The traditional meta-heuristic technique is prone to get trapped in local optima, but SCLO keep on updating the candidate state in the second phase by using different algorithm if after *m* evaluations candidate does not changes its state. In other words, we can say that a counter *c* is being set off after *m* continuous function evaluation candidate does changes improve its fitness then a candidate has to update its state depending upon the Eq. (4).

To regulate the fitness improvement a Boolean variable *P* is being taken which is set to 0 for an improvement and set to 1 for no improvement, the count of c increases whenever *P* is equal to 0, when *c* gets equal to *m*, then x_i improves updates according to Eq. (4):

$$x_i = \begin{cases} gbest_i + ((r_3 w(ub_i - lb_i))) \; if r_3 < 0.5 \\ gbest_i - ((r_3 w(ub_i - lb_i))) \; otherwise \end{cases} \quad (4)$$

where, r_3 is a random variable that can have a value in the range of [0,1].

While updating the value of some variables of *x* can sometimes cause the values to go out of the range of upper bound and lower bound in that case candidate variables are updated as per Eq. (5):

$$x_i = \begin{cases} gbest_i \; if \; x_i > ub_i \\ gbest_i \; if \; x_i < lb_i \end{cases} \quad (5)$$

In SCO the single candidate solution is generated by using loss optimization function to evaluate the losses occurred during training the model for machine translation on low resource data.

The loss value comes after one iteration is being updated according to the steps of the algorithm rather than using traditional gradient based algorithms. As the iteration increases the loss function moves toward global convexity or global optimization. The single candidate value of *x* will come by using the Eq. (6):

$$x = SparseCatgegoricalCrossEntropy(y, \widehat{y}) \quad (6)$$

And then it is iteratively updated in search of a better solution. The steps of the algorithm for loss optimization of the machine translation model are as follows. As we have mentioned the process starts by generating a candidate solution by using Eq. (6), evaluating its fitness and recording this candidate as *g best* (*global best position*) and its fitness as global best fitness.

This is a repetitive process that terminates when it reaches T function evaluations, we can also call it a total number of epochs. The candidate solution updates its position in two phases based on Eqs. (1 and 3), respectively. After updating the candidate position, the fitness of the newly generated candidate solution *f(x)* is evaluated and compared with *f(gbest)*. If *f(x)* is better than *f(best)*, then *gbest* and *f(gbest)* are replaced by *x* and *f(x)* respectively. The learning process is continued until a total number of function evaluations reaches *T*.

The pseudo-code algorithm of single candidate loss optimization is presented in Table (2).

As earlier, the SCLO algorithm has two phases exploitation and exploration. It has been shown in the paper (Shami *et al*., 2024) that SCO has outperformed all other optimization algorithms during bench mark analysis of unimodal functions which are given in Table (3). In exploitation, we do α function evaluations on unimodal functions given below in Table (3) and β evaluations based on the evaluation function mentioned in Table (4), unimodal functions are evaluated on three different values of *D* as given in Table (5) are used to optimize the hyperparameters used in the MT algorithm. The value of α and β are set to 250 each making total function evaluation equal to 500.

To test the effectiveness of the SCLO algorithm we test it on 10 benchmark functions defined in Tables (5-6), which are divided for testing exploitation and exploration by using Unimodal and Multimodal functions respectively. Unimodal functions are used to test the optimization ability of SCLO since they have only one global optimum whereas multimodal function is used to assess the exploration ability as they have multiple local optima. The results are shown in the table for four dimensions 64, 128, 256, 512.

From Table (5) it has been estimated that D = 256 is the best hyperparameter for the size of the loss optimization vector. In Table (6) we have done the function evaluation using a value of D = 256.

Based on the statistical results which we have got from Tables (5-6), the final hyperparameters can be estimated and listed in the Table (7).

Table 2: Algorithm of SCLO

1	Set c = 0, p = 0 and define the values of α, β and m
2	Generate the initial candidate solution based on Eq. (6) and calculate its fitness f(gbest) on seven unimodal test functions described in Table 1 and Table 2 for the first and second phases
3	**while** t < maximum number of function evaluations
4	**if** t < α **then**
5	Update the dimension position based on Eq. (1)
6	**else**
7	**if** p = 0 **then**
8	c = c + 1
9	**end if**
10	**if** c = m **then**
11	reset the counter c = 0
12	Update the dimension position based on Eq. (4)
13	**else**
14	Update the dimension position based on Eq. (3)
15	**end if**
16	Transfer the *gbest* candidate solution to the model for updating weights of the model
17	Calculate the fitness of the new candidate solution *f(x)* generated by Eq. (6)
18	**if** f(x) is better than f(gbest) **then**
19	*gbest = x*
20	*f(gbest) = f(x)*
21	p = 1
22	else
23	p=0
24	**end if**
25	t = t+1
26	**end while**
27	return *gbest*

Table 3: Unimodal test functions for first phase

S. No	Fitness functions	Range	f_{min}
1	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100, 100]	0
2	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
3	$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	[-100, 100]	0
4	$f_4(x) = \max\{\lvert x_i \rvert, 1 \le i \le n\}$	[-100, 100]	0
5	$f_5(x) = \sum_{i=1}^{n} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$	[-30,30]	0
6	$f_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	[-100, 100]	0
7	$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	[-1.28,1.28]	0

Table 4: Multimodal test functions for second phase

S. No	Function	Range	f_{min}
1	$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x \rvert})$	[-500, 500]	-418.9829
2	$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	[-5.12,5.12]	0
3	$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0

Table 5: Results of the average fitness of unimodal and multimodal function for different values of D

S. No	Function	Statistical parameters	D = 64	D = 128	D = 256	D = 512
1	f_1	Mean Std	1.74E-11, 3.42E-11	4.29E-01, 3.62E-01	8.53E-2680	1.18E-01, 1.53E-01
2	f_2	Mean Std	1.18E-06, 1.52E-06	2.83E-01, 4.98E-02	6.21E-2210	2.69E+02, 2.90E+02
3	f_3	Mean Std	1.46E-04, 8.88E-04	6.77E+04, 3.77E+04	2.88E-1670	1.90E+04, 2.02E+04
4	f_4	Mean Std	2.48E-06, 4.18E-06	3.97E+01, 7.28E+00	2.582E-14, 1.43E-13	4.78E+01, 4.18E+00
5	f_5	Mean Std	1.97E+02, 2.45E-02	2.56E+02, 2.99E+01	1.78E+02, 5.29E-02	1.92E+05, 1.04E+04
6	f_6	Mean Std	3.75E+01, 5.00E-01	3.77E+01, 1.60E+00	3.10E+01, 1.31E+00	1.38E+03, 1.74E+02
7	f_7	Mean Std	2.97E-03, 2.50E-03	2.66E-02, 1.27E-02	4.85E-04, 4.10E-04	1.26E+00, 2.88E-01
8	f_8	Mean Std	-9.21E+03, 1.46E+03	2.53E+03, 2.33E+03	-2.698E+04, 2.98E+03	1.17E+01, 1.42E+00
9	f_9	Mean Std	1.88E-11, 4.16E-11	4.88E+00, 5.92E+00	00	1.28E+02, 1.11E+01
10	f_{10}	Mean Std	1.19E-07, 2.98E-07	4.93E-02, 1.96E-02	8.88E-160	4.77E+00, 3.78E+00
Rank			2	3	1	4

Table 6: Statistical result of exploitation and exploration phase for different values of b, α and β = 250

S. No	Func	Statistical parameters	b = 0.5	b = 1.2	b = 1.8	b = 2.4	b = 2.9
1	f_1	Mean Std	3.22E-138 1.36E-135	1.78E-1670	2.25E-222 0	6.98E-2870	00
2	f_2	Mean Std	1.38E-68 6.92E-68	6.66E-47 4.37E-46	2.21E-118 1.25E-107	3.42E-132 1.79E-134	4.28E-1970
3	f_3	Mean Std	2.56E+01 8.90+E02	2.32E+01 .95E-02	1.98E+01 3.86E-02	2.72E+01 1.35E-01	2.58E+01 10.02E-02
4	f_4	Mean Std	1.37E-03 1.28E-03	1.48E-03 1.40E-03	1.44E-03 1.41E-03	00	00
5	f_5	Mean Std	00	00	00	00	2.27E+00 4.84E-01
6	f_6	Mean Std	1.98E+00 1.45E-01	2.68E+00 2.54E-01	2.72E+00 4.35E-01	1.68E+00 5.98E-01	1.94E-02 2.80E-02
7	f_7	Mean Std	7.02E-03 1.57E-02	5.99E-03 2.23E-02	1.77E-02 2.33E-02	5.74E-03 2.38E-02	3.88E-02 3.68E-01
8	f_8	Mean Std	2.89E-03 1.38E-01	5.88E-03 9.97E-03	4.13E-01 1.58E-04	2.86E-01 3.94E-13	2.97E-010
9	f_9	Mean Std	-3.78E+00 7.88E-04	-3.86E+00 7.38E-04	-3.68E+00 1.18E-03	-2.56E+00 1.76E-05	-2.94E+00 7.89E-05
10	f_{10}	Mean Std	-10.52E+00 1.87E+00	-9.22E+00 1.43E+00	-9.35E+00 2.98E+00	-8.89E+00 3.16E+00	-7.98E+00 2.88E+00
Rank			5	4	3	1	2

Table 7: Hyperparameters used in single candidate loss optimization algorithm

S. No	Hyperparameter	Value
1	α	250
2	β	250
3	T(epochs)	500
4	m	15
5	b	2.4
6	D	256

In the next section, we will see the proposed Machine translation model which uses the SCLO algorithm tuned on hyperparameters stated in Table (7).

Proposed Architecture of Machine Translation Model

In the proposed architecture model, we have used a vanilla transformer with six layers of encoder and six layers of decoder, novelty is being introduced by adding an SCLO algorithm for loss optimization rather than traditional gradient-based optimization (Dogo *et al*., 2018) which are prone to vanishing gradient and local optima problems, further Byte-Pair encoding is used for tokenization and text segmentation and BERT is used for contextualized word embeddings. These word embeddings are fed into the transformer model for machine translation. The hyperparameters that are used in SCLO are given in Table (7).

Figure (2) shown below is the proposed context-based machine translation architecture which integrates BPE (Sennrich *et al*., 2016), pre-trained BERT (Devlin *et al*., 2019), SCLO and a transformer (Vaswani *et al*., 2017). The transformer is represented in box form for illustration purposes, it executes all its functionalities as the standard transformer performs.

In the next Table (8), step by step loss optimization process is explained, where SCLO is used as a loss optimization strategy.

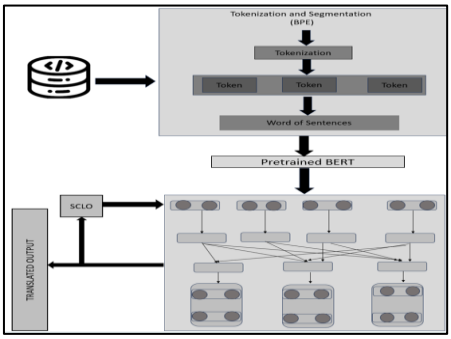

Fig. 2: Proposed model architecture

Table 8: Algorithm of proposed model

1	L: Average Loss of the model
2	E: Total number of epochs
3	$\boldsymbol{V}$: Input embedding vector
4	D: Size of the input sentence
5	For every epoch e in E
6	$L_e = 0$
7	For every i in D
8	$L_i = SparseCategoricalCrossEntropy(y_i, \hat{y}_i)$
9	$\hat{y}_i = SCLO(\boldsymbol{V_i})$
10	$L_e = L_e + L_i$
11	$L = L + \frac{L_e}{D}$
12	$L = \frac{L}{E}$
13	return L

Data Collection

Data collection for low-resource language is a very challenging task, we have manually created data for the experiments as well as some data is being created by freelancers. The dataset is not very big it is 80 K sentences as it is a low-resource language and very little or no data is available for experiments. Low resource data falls into two categories either having little available data or very limited research work being done on that language, our language comes under both categories. The parallel corpus of Bhojpuri and English languages is built in two forms Ambiguous and non-ambiguous which contain ambiguous and non-ambiguous sentences respectively. Ambiguous sentences are 11000 approximately they are divided in the ratio of 7:3, out of which approximately 7600 sentences are mixed with non-ambiguous sentences for training purposes. The remaining 3400 sentences are used for testing purposes. We have approximately 70000 general non-ambiguous parallel corpus sentences which are used for training and testing in the ratio of 7:3.

Materials and Methods

In our research work, the material used is parallel corpus, ambiguous parallel corpus of English-Bhojpuri language, which has been described in the data collection section in detail, the proposed model is illustrated in Fig. (2) and its algorithm in Table (8).

Results and Discussion

The training data is a parallel corpus of the English-Bhojpuri language; to train the proposed mode for ambiguous language, 11000 ambiguous samples have been taken for training and validation in the ratio of 7:3. In the training process the parallel corpus is passed to Byte pair encoding for tokenization and vocabulary creation after the segmented words are based to Pretrained BERT for generation of context-based embedding vectors. These vectors are passed into Transformer models for machine translation. The novelty is that we have used the single candidate loss optimization process for Loss optimization after every epoch or functional evaluation. The total functional evaluations by adding α, β functional evaluation is set to 500. The results have been evaluated based on the Bilingual Evaluation Understudy (BLEU) score, learning i.e., exact optimization time, iteration time in minutes, Accuracy and Precision. The hyperparameters used in designing the model are given in Table (9).

In the result analysis, we have compared our proposed model with some other state-of-the-art models like mBERT, RemBERT, LASER, GEMBA for contextual embedding generation and used those embedding vectors for translation using transformers. These models have used their traditional ADAM optimizer for learning weights rather than novel SCLO.

It is being found in the experiments that our model has outperformed other traditional word embedding models in terms of the quality of embedding vector generation when they are coupled with traditional transformers for machine translation.

We have presented the results in three tables. Tables (10-12) covering all important performance metrics to evaluate the efficiency of our proposed model.

Tables (8-9) represent the results in two parts in the first part we have compared the performance of our proposed model with metrics like Accuracy, Precision, training time per epoch and BLEU score for both variations ambiguous and non-ambiguous. In Table (9) remaining metrics like METEOR and RIBES are also being used to justify the performance of our model These metrics are the standard metrics that ensure the performance of the model and ensure that the novelty that is being used to train the weight parameters refer to Eq. (2) is more effective than traditional gradient-based algorithms. The accuracy of the model has been calculated using the following code snippet for every batch and then the average accuracy has been calculated after every epoch:

→ accuracy-metric = tf. keras. metrics. SparseCategoricalAccuracy()
→ batch-accuracy = accuracy-metric. result()
→ total-accuracu-per-epoch = batch-accuracy
→ accuracy-metric. Reset-states () # reset the accuracy state for the next batch

Table 9: Hyperparameters used in transformer model

S. No	Hyper Parameters	Value
1	Encoder and Decoder	6
2	Encoder Embedding Dimensions	512
3	Decoder Embedding Dimensions	512
4	Encoder Attention heads	2
5	Decoder Attention heads	2
6	Dropout (Feed Forward Layer)	0.2
7	Optimizer	SCLO
8	Number of epochs	500

Table 10: Result analysis part one

Model name	Accuracy	Precision	Learning time per epoch in (sec)	BLEU (general)	BLEU (ambiguity)
SCLO	0.9743	0.96734	2.56	0.77	0.61
mBERT	0.9116	0.92353	4.32	0.56	0.47
RemBERT	0.9209	0.91602	6.28	0.66	0.52
LASER	0.9328	0.93207	5.12	0.59	0.52
GEMBA	0.9308	0.94307	6.16	0.70	0.60

Table 11: Results analysis part two

Model	Gleu (general)	Gleu (ambiguity)	Meteor (general)	Meteor (ambiguity)	Ribes (general)	Ribes (ambiguity)
SCLO	0.48	0.33	0.622	0.584	0.22	0.14
mBERT	0.22	0.71	0.411	0.358	0.19	0.12
RemBERT	0.221	0.175	0.532	0.476	0.17	0.11
LASER	0.188	0.165	0.465	0.377	0.17	0.12
GEMBA	0.42	0.34	0.579	0.451	0.22	0.12

Table 12: Loss optimization results

Model name	Mean loss value (General)	Mean loss value (Ambiguity)
SCLO	0.03114	0.08745
mBERT	0.07589	0.10852
RemBERT	0.07358	0.11365
LASER	0.10589	0.17662
GEMBA	0.05784	0.11022

In Table (9) we have presented a performance evaluation of some other translation metrics and compared the results of our proposed model with other models.

In Table (11) the Optimized Loss value of SCLO algorithm has been shown it is calculated using algorithm designed in Table (6).

The SCLO algorithm is used to provide an effective loss optimization technique than the traditional gradient-based loss optimization techniques, the findings are shown in detail in our result section.

Conclusion

The paper concludes by stating that the translation of low-resource languages is a challenging task, as the data is less. The fewer data induce the problem of overfitting and gradient descent algorithms are not enough to perform the task of training the parallel corpus and solving the ambiguity problems. The machine translation models need ample amounts of data are few data causes overfitting, which is a type of the biggest challenge in machine learning. In our proposed work we have used the optimization algorithms in solving this problem by introducing the singe candidate loss optimization technique to optimize the loss not by gradient-based methods but by constraint-based optimization algorithms. Our results verified that the model is successfully implemented. In the future, we will use this technique to solve domain adaption problems which is another challenge in implementing machine translation models for low-resource languages.

Acknowledgment

I want to sincerely thank Prof. S. S. Bedi, whose important advice, perceptive criticism, and unwavering support have been crucial to the accomplishment of this project. The direction of their work has been greatly influenced by their proficiency in neural machine translation and for his insightful conversations, recommendations, and encouragement during the study process, I am also grateful to my fellow co-authors and especially grateful to Dr. Zubair Khan for creating a cooperative and thought-provoking atmosphere.

Funding Information

The research work was conducted without any funding from any funding agency, private or any government institution.

Author's Contributions

Rituraj Dixit: Came up with the idea for the study, created the methodology, and collected the data

Sarabjeet Singh Bedi and Mohammad Zubair Khan: have helped with the composition and editing of the work. The final text was examined and approved by all authors.

Ibrahim Aljubayri: I trained and assessed the model.

Ethics

There are no human subjects, private information, or delicate ethical issues in this study. Therefore, there was no need for official ethical approval.

References

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv:1409.0473*. https://doi.org/10.48550/arXiv.1409.0473

Bentivogli, L., Bisazza, A., Cettolo, M., & Federico, M. (2016). Neural Versus Phrase-Based Machine Translation Quality: a Case Study. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 257–267. https://doi.org/10.18653/v1/d16-1025

Bhaduri, S. (1990). Evaluation of Different Techniques for Detection of Virulence in Yersinia Enterocolitica. *Journal of Clinical Microbiology*, *28*(4), 828–829. https://doi.org/10.1128/jcm.28.4.828-829.1990

Chakrawarti, R. K., & Bansal, P. (2017). Approaches for Improving Hindi to English Machine Translation System. *Indian Journal of Science and Technology*, *10*(16), 1–8. https://doi.org/10.17485/ijst/2017/v10i16/111895

Chethan, M., Basavaraddi, C. S., & Shashirekha, H. L. (2014). A Typical Machine Translation System for English to Kannada. *International Journal of Scientific & Engineering Research*, *5*(4).

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Choudhary, H., Pathak, A. K., Saha, R. R., & Kumaraguru, P. (2018). Neural Machine Translation for English-Tamil. *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, 770–775. https://doi.org/10.18653/v1/w18-6459

Dabre, R., Shrotriya, H., Kunchukuttan, A., Puduppully, R., Khapra, M., & Kumar, P. (2022). IndicBart: A Pre-Trained Model for Indic Natural Language Generation. *Findings of the Association for Computational Linguistics: ACL 2022*, 1849–1863. https://doi.org/10.18653/v1/2022.findings-acl.145

Deep, K., Kumar, A., & Goyal, V. (2020). Punjabi to English Bidirectional NMT System. *Proceedings of the 17th International Conference on Natural Language Processing (ICON): System Demonstrations*, 7–9.

Devlin, J., Chang, M.-W., Lee, K., & Google, K. T. (2018). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805*. https://doi.org/10.48550/arXiv.1810.04805

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 the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B., & Aigbavboa, C. O. (2018). A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 92–99. https://doi.org/10.1109/ctems.2018.8769211

Dongare, P. (2024). Creating Corpus of Low Resource Indian Languages for Natural Language Processing: Challenges and Opportunities. *Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation*, 54–58. https://aclanthology.org/2024.wildre-1.8/

Gaido, M., Papi, S., Cettolo, M., Cattoni, R., Piergentili, A., Negri, M., & Bentivogli, L. (2024). Automatic Subtitling and Subtitle Compression: FBK at the IWSLT 2024 Subtitling track. *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, 86–96. https://doi.org/10.18653/v1/2024.iwslt-1.13

Gain, B., Appicharla, R., Chennabasavraj, S., Garera, N., Ekbal, A., & Chelliah, M. (2022). Low-Resource Chat Translation: A Benchmark for Hindi-English Language Pair. *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, 83–96.

Ghazal, A. (2015). A Critical Discourse Analysis of SANA and Aljazeera English Channel's Coverage of Syria's 2014-2015 Uprising. *International Journal of English Linguistics*, *5*(3), 143. https://doi.org/10.5539/ijel.v5n3p143

Goyal, V., & Sharma, D. M. (2019). The IIIT-H Gujarati-English Machine Translation System for WMT19. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, 191–195. https://doi.org/10.18653/v1/w19-5316

Haddow, B., Bawden, R., Barone, A. V. M., Helcl, J., & Birch, A. (2022). Survey of Low-Resource Machine Translation. *Computational Linguistics*, *48*(3), 673–732. https://doi.org/10.1162/coli_a_00446

Haque, M., & Hasan, M. (2018). English to Bengali Machine Translation: An Analysis of Semantically Appropriate Verbs. *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, 217–221. https://doi.org/10.1109/iciset.2018.8745626

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., Liu, S., Liu, T.-Y., Luo, R., Menezes, A., Qin, T., Seide, F., Tan, X., Tian, F., Wu, L., Ming, Z. (2018). Achieving Human Parity on Automatic Chinese to English News Translation. *ArXiv:1803.05567*. https://doi.org/10.48550/arXiv.1803.05567

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., & Birch, A. (2018). Marian: Fast Neural Machine Translation in C++. *Proceedings of ACL 2018, System Demonstrations*, 116–121. https://doi.org/10.18653/v1/p18-4020

Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M. M., & Kumar, P. (2020). IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4948–4961. https://doi.org/10.18653/v1/2020.findings-emnlp.445

Khan, J., Nadeem, W. A., & Durrani, N. (2017). Machine Translation Approaches and Survey for Indian Languages. *ArXiv:1701.04290*. https://doi.org/10.48550/arXiv.1701.04290

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. *Proceedings of ACL 2017, System Demonstrations*, 67–72. https://doi.org/10.18653/v1/p17-4012

Koehn, P., & Knowles, R. (2017). Six Challenges for Neural Machine Translation. *ArXiv:1706.03872*. https://doi.org/10.48550/arXiv.1706.03872

Kunchukuttan, A., Mishra, A., Chatterjee, R., Shah, R., & Bhattacharyya, P. (2014). Sáta-Anuvādak: Tackling Multiway Translation of Indian Languages. *Pan*, *841*(54, 570), 4–135.

Lingam, K., Ramalakshmi, E., & Inturi, S. (2014). English to Telugu Rule based Machine Translation System: A Hybrid Approach. *International Journal of Computer Applications*, *101*(2), 19–24. https://doi.org/10.5120/17659-8474

Luong, 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*, 1412–1421. https://doi.org/10.18653/v1/d15-1166

Nakazawa, T., Nakayama, H., Ding, C., Dabre, R., Higashiyama, S., Mino, H., Goto, I., Pa Pa, W., Kunchukuttan, A., Parida, S., Bojar, O., & Kurohashi, S. (2020). Overview of the 7th Workshop on Asian Translation. *Proceedings of the 7th Workshop on Asian Translation*, 44–1. https://doi.org/10.18653/v1/2020.wat-1.1

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). fairseq: A Fast, Extensible Toolkit for Sequence Modeling. *Proceedings of the 2019 Conference of the North*, 48–53. https://doi.org/10.18653/v1/n19-4009

Philip, J., Namboodiri, V. P., & Jawahar, C. V. (2019). A Baseline Neural Machine Translation System for Indian Languages. *ArXiv:1907.12437*. https://doi.org/10.48550/arXiv.1907.12437

Satpute, M. R. S., & Agrawal, A. (2023). Intelligent Systems AandApplications in A Critical Study of Pragmatic Ambiguity Detection in Natural Language Requirements. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(3s), 249–259.

Senellart, J., Zhang, D., Wang, B., Klein, G., Ramatchandirin, J.-P., Crego, J., & Rush, A. (2018). OpenNMT System Description for WNMT 2018: 800 Words/sec on a Single-Core CPU. *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, 122–128. https://doi.org/10.18653/v1/w18-2715

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 (Volume 1: Long Papers)*, 1715–1725. https://doi.org/10.18653/v1/p16-1162

Shami, T. M., Grace, D., Burr, A., & Mitchell, P. D. (2024). Single Candidate Optimizer: A Novel Optimization Algorithm. *Evolutionary Intelligence*, *17*(2), 863–887. https://doi.org/10.1007/s12065-022-00762-7

Shukla, A., Bansal, C., Badhe, S., Ranjan, M., & Chandra, R. (2023). An Evaluation of Google Translate for Sanskrit to English translation via sentiment and semantic analysis. *Natural Language Processing Journal*, *4*, 100025. https://doi.org/10.1016/j.nlp.2023.100025

Singh, S., Anand Kumar, M., & Soman, K. P. (2018). Attention Based English to Punjabi Neural Machine Translation. *Journal of Intelligent & Fuzzy Systems*, *34*(3), 1551–1559. https://doi.org/10.3233/jifs-169450

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *ArXiv:1409.3215*, *27*. https://doi.org/10.48550/arXiv.1409.3215

Tan, X., Chen, J., He, D., Xia, Y., Qin, T., & Liu, T.-Y. (2019). Multilingual Neural Machine Translation with Language Clustering. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 963–973. https://doi.org/10.18653/v1/d19-1089

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *ArXiv:1706.03762*, *30*. https://doi.org/10.48550/arXiv.1706.03762