Journal of Computer Science

Research Article

Evaluating Machine Translation for Domain Specific Low-Resource Nepali-English Language Pairs: The Impact of Tokenization on Statistical and Neural Techniques

Amit Kumar Roy and Bipul Syam Purkayastha

Department of Computer Science, Assam University, Silchar, Cachar, Assam, India

Article history
Received: 08-04-2025
Revised: 29-06-2025
Accepted: 02-07-2025

Corresponding Author:
Amit Kumar Roy
Department of Computer Science,
Assam University, Silchar, Cachar,
Assam, India
Email: amitroy.cs@gmail.com

Abstract: In the modern era, the field of Machine Translation (MT) has seen a significant shift towards Neural Machine Translation (NMT) techniques, which have surpassed traditional Statistical Machine Translation (SMT) models in terms of the quality of translation. Despite this, the efficacy of these techniques may differ based on the language combination in consideration. While SMT is somewhat more flexible in this regard, NMT often needs sizable parallel corpora to attain high translation accuracy. As a result, a benchmark system capable of offering sufficient translation for languages with limited resources, like Nepali, remains a pipe dream. This paper focuses on translating text using statistical and neural MT techniques for the under-resourced English-Nepali language pair. As a part of this system development, we built a parallel corpus of English-Nepali in the tourism domain. We explore the impact of different tokenization techniques on translation outcomes. A substantial analysis is also done for the performance of both approaches using automatic evaluation metrics, BLEU and TER. This paper aims to provide insights into the applicability of SMT and NMT for the under-resourced English-Nepali language pair in light of two popular epitomes of tokenization and to determine the most effective approach for achieving accurate translations.

Keywords: Statistical MT, Neural MT, Tokenization, Sentence Piece, Low-Resource MT, Nepali Language

Introduction

The development of neural methods has recently given new life to machine translation, bringing the dream of automatic language translation closer to becoming a reality; it has also overshadowed the Statistical approach, the prevailing framework in MT research for almost three decades. However, because of the data-hungriness of the neural approach, there is a major worry that languages with scarce resources may not benefit to the same extent as well-resourced languages. To avoid leaving a low-resource language like Nepali behind in the context of these advancements, we are taking steps to apply NMT and SMT methods to English-Nepali translation and vice versa. To achieve good translation accuracy, NMT generally requires large parallel corpora, whereas SMT is a little more adaptive in this matter. But still, a standard system that can offer appropriate translation for under-resourced languages like Nepali remains an aspiration.

One of the main tasks in all forms of machine translation processes is tokenization, which involves breaking down text sentences into a set of tokens that make up the corpus's vocabulary of distinct tokens. By allowing the model to comprehend and process each word discretely, it aids in improving the precision of machine translation systems. A collection of MT experiments conducted using two distinct tokenizers is presented in the paper.

The Nepali language is the national and official language of the Republic of Nepal and the official language of the northeastern state of Sikkim. Nepali is also added with 21 other languages in the 8th schedule of the Indian Constitution as an official language of India. The Nepali language is an under-resourced language, as there are very few digital resources available till date. Apart from the scarcity of digital resources, the English-Nepali language pair has many notable differences in their script, structure, and morphology (Roy and Purkayastha, 2023).

In this paper, we tried to deduce whether the Neural approach performs equivalently well with respect to the Statistical approach of Machine translation for the language combination being considered. To gain insight, first, we built a bilingual parallel corpus of 29K sentences from monolingual English corpora, which was manually translated with the help of a native language speaker. Then design an NMT system using the standard OpenNMT toolkit and an SMT system using the Moses toolkit to compare translation accuracy for the rarely tested under-resourced language pair, Nepali to English and English to Nepali. This study also examines the impact of preprocessing strategies on SMT and NMT in English-Nepali MT using two prominent tokenization schemes, Moses Tokenizer and Sentence Piece Byte-Pair-Encoding Tokenizer. To check the quality of translations the automatic evaluation metrics, such as BLEU and TER, are used in this research work.

Linguistic Divergence of English and Nepali Languages

The following provides a quick comparison of English and Nepali languages.

Origin

The English language belongs to the Indo-European language family, whereas the Nepali language belongs to the Eastern-Pahari subfamily of the Indo-Aryan family of languages.

Script

The languages also differ in their scripting, where English is written using the Latin-based script, and Nepali, like most of the Indian languages that include Hindi and Sanskrit, use the Devanagari-based script (Roy and Purkayastha, 2023).

Word Order

English fellows Subject-Verb-Object, while Nepali fellows Subject-Object-Verb word structure (Roy and Purkayastha, 2023).

Vowels and Consonants

The English alphabet has 26 letters, out of which 5 are vowels and 21 are consonants. In the Nepali language, there are 11 vowels and 33 consonants, giving a total of 44 alphabets (Bal, 2004).

Derivation

Both Nepali and English have a system of derivation, but they differ in the types of affixes used. Nepali has a rich system of derivational affixes that can change the meaning of words, whereas English tends to use prefixes and suffixes to create new words (Bal, 2004).

Agglutination

Nepali is an agglutinative language, whereas English is not. In Nepali, words are formed by adding suffixes to roots or stems, whereas in English, words are formed through a variety of processes, including affixation, compounding, and conversion (Bal, 2004).

Being a rich language in terms of morphology, Nepali is an interesting example as far as the development of machine translation systems is concerned. In the language, tense, aspect, mood, number, gender, and case are encoded by the affixes; this is why there is a great deal of diversity in word-forms: Multiple forms can be surface instantiations of the same root. This inflectional depth also greatly increases lexical coverage and entails an increase in data insufficiency when it comes to low-resource situations. This is because, in the previous architecture of MT where the word-based tokenization is used, all inflected forms are assumed as independent tokens, which leads to the creation of fragmented alignments and hinders the system's generalization.

Related Works

The emergence of the NMT technique shifted the course of machine translation from the traditional SMT technique; however, each technique has its own advantages and disadvantages. Although NMT is more widely used today, there are certain clear challenges when using the NMT approach for MT (Koehn and Knowles, 2017). The impacts of the SMT and NMT systems on the translated result have been investigated in multiple studies or a variety of language pairs. A comparison of SMT and NMT for the low-resource language Khasi is reported by Singh and Hujon (2020). The SMT system performs better than the NMT system for the language pair in the study conducted on the dataset of 7639 bilingual sentences and 13276 monolingual sentences. Another such work done for German-English languages reports that the performance of NMT and SMT are the same when trained with a dataset of 270k segments (Lohar *et al.*, 2019). The NMT system's performance drops significantly when using less training data, but the PBMT system's performance drops slightly. Stasimioti *et al.* (2019) compared SMT and NMT in the English-Greek language pair and demonstrate that the general NMT achieves higher scores than those obtained in SMT using both automatic and human evaluation measures.

The use of tokenization plays a vital role in Machine Translation (MT) systems architecture, as it has a significant impact on performance and translation quality. Researchers have experimented with how different tokenization methods affect the quality of translations in a wide range of language pairs and domains.

Domingo *et al*. (2023) evaluated on an empirical basis, the role of five different tokenizers: Byte-Pair Encoding (BPE), Basic BPE, Joined BPE (JBPE), Char-BPE, and Sentence Piece in the quality of automatic translation of ten linguistic pairs in which no obvious marks that signify a word boundary are available. As per the data, it has been established that tokenization plays an important role in determining the efficacy of Neural Machine Translation, that the best tokenizer is not generalizable across languages, and that it often changes with the particular language pair. Careful selection of tokenization techniques vastly helped to improve translation quality by up to 12 BLEU and 15 TER units. The findings also indicate that the effectiveness of tokenizers does not use the same in a symmetrical manner; it may perform better in one direction and then be worse in the reverse direction. For example, the Sentence Piece model could effectively be used in translating Arabic to English, but it did not show a similar performance when reversing the process.

The authors of the research performed an experimental investigation utilizing four distinct tokenization libraries: Moses, NLTK, OpenNMT, and IndicNLP. In this paper, by combining various tokenizer combinations, 12 distinct NMT models are created and tested. The study concluded that tokenization has a considerable impact on the quality of NMT. Furthermore, the optimum tokenizer may differ based on the language pair. This work sheds light on the importance of tokenization in increasing the performance of NMT systems, particularly for low-resource languages such as Assamese (Ahmed *et al*., 2023).

The Nepali language is poor in resources yet rich in morphology. Understanding which technique is best for the growth of Nepali MT is our primary objective. The following section summarises the progress of MT for the nepali language up to the present day.

The first machine translation project for the Nepali-English pair was Dobhase. Dobhase was a rule-based system that accepted an input string, parsed it, generated the syntax for the target language, and output the translation. It was unable to handle sentences of complex structures (multiple conjunctions, ambiguous words, etc.) and has been discontinued. At present, the system constitutes a bilingual dictionary of 22,000 words (Bista *et al*., 2007).

English to Nepali using SMT was another project that aimed to translate English to the most likely Nepali sentences by applying the SMT (Statistical Machine Translation) approach. This project was able to give an accuracy of 68 % which is 2.7 out of 4 (Acharya and Bal, 2018).

NMT is relatively new for the Nepali–English pair. In 2018, P. Acharya, in his paper, used a small portion of the parallel corpus from the Nepali National Corpus (NNC) collected by Yadava *et al*. They applied SMT and NMT techniques. On their test sets, the highest BLEU scores they obtained were 5.27 and 3.28 in SMT and NMT, respectively (Acharya and Bal, 2018).

The discussion will focus on the NMT model suggested by Laskar *et al*. (2019), which has a transduction attention mechanism to perform the cross-lingual translation in the WMT19 setting (Laskar *et al*., 2019). The parallel corpus used to train was of Hindi and Nepali, and the test and analysis were done for both Hindi and Nepali translations. The official WMT19 evaluation produced a BLEU score of 53.7 (Hindi to Nepali) and 49.1 (Nepali to Hindi) in the case of the contrastive system type. These scores can be attributed to the similarity between the involved languages with regard to their nature of being close as well as sharing the Devanagari script. To explore the possibility of other language families, the authors trained the Transformer NMT (Chaudhary *et al*., 2020) with a small corpus of hand-labelled or aligned Tamang-Nepali pairs of sentences (about 15K) as training data. The Nepali to Tamang and the Tamang to Nepali result scores are mentioned as 27.74 and 23.74, respectively. The very dataset is now used as a benchmark of Tamang-Nepali MT.

Google provides Google Translate, a free service that translates images, speech, text or real-time video, incorporating multiple languages from one to another. Google Translate introduced Nepali language support in the 36th stage, which was launched in December 2013. To increase translation quality, Google Translate has involved native speakers of the languages in the revision and verification of translated output (Devi *et al*., 2023).

The next paper discusses about a bidirectional transformer-based NMT system constructed specifically for English-Nepali legal translation, which uses a custom-built parallel corpus of 125,000 sentences of the legal domain. The arrangement is also compared to Recurrent Neural Networks (RNNs) with LSTM architecture, whose performance is inferior. The researchers used Fairseq tools to train these models. The system achieves the highest BLEU score of 7.98 for Nep-Eng and 6.83 for Eng-Nep in the transformer-based model (Poudel *et al*., 2024).

In the study by Roy *et al*. (2024), a self-made bilingual parallel corpus of 17,000 sentences was subjected to the statistical approach of machine translation. The author's method translated both ways, from English to Nepali and Nepali to English, obtaining BLEU scores of 21.13 and 22.26, respectively. When comparing their findings with Google Translate, the authors discovered that the proposed method works better than Google Translate when translating from English to Nepali in terms of automatic evaluation metrics scores.

Despite the advancements and comparative studies between SMT and NMT, several research gaps persist in the domain of MT, particularly concerning the Nepali language. Although NMT has shown promising results, its performance still falls short of SMT in certain scenarios, especially with low-resource languages like

Nepali. The limited research on Nepali MT systems underscores the need for more in-depth investigations into the specific challenges posed by its complex morphology and limited linguistic resources. Comprehensive studies exploring the most effective tokenization techniques for Nepali and other morphologically rich, low-resource languages are lacking. Given the variability in tokenizer performance based on translation direction, further research is required to identify and optimize tokenization methods for Nepali-English and English-Nepali translation. The performance of both SMT and NMT systems is heavily influenced by the size and quality of the training corpus. Studies such as those by Acharya and Bal (2018); Laskar *et al.* (2019) emphasize the need for larger, high-quality parallel corpora for Nepali. To address these research gaps we design bidirectional NMT and SMT systems that employ two prominent tokenization techniques for the low-resource English-Nepali pair, and their results are evaluated over the standard automatic evaluation metrics. This will not only enhance the quality of Nepali machine translation systems but also contribute to the broader understanding and development of MT for other low-resource languages.

Methods

Corpus and Preprocessing

A bilingual corpus was created through a processing of 29,127 English sentences; monolingual tourism texts that can be found in the NPLT (National Platform for Language Technology) and conversational utterances that are common in the same domain. In order to simplify this process, native Nepali speakers translated the English text into the Nepali language and thus created a parallel bilingual corpus. The data were cleaned in various computational cleansing processes before being fed to further Machine Translation (MT) experiments: Splitting files, tokenisation, truecasing and cleaning.

The quality and the format of the data are important determinants of Machine Translation (MT) system effectiveness. The bilingual parallel corpus has been arranged in the form of a single .xlsx file, which is then split into two monolingual text files, with one consisting of the sentences written in Nepali and the other one including the English translations of those sentences. These files are used as the input to the next preprocessing steps, as shown in Fig. 1.

A critical pre-processing procedure of Natural Language Processing is the tokenization procedure where the textual data is broken down into small parts, i.e. tokens separated with inter-token spaces. The process allows the system to acquire contextual meaning and record semantic text correctly to be translated later, hence providing more accurate translations.

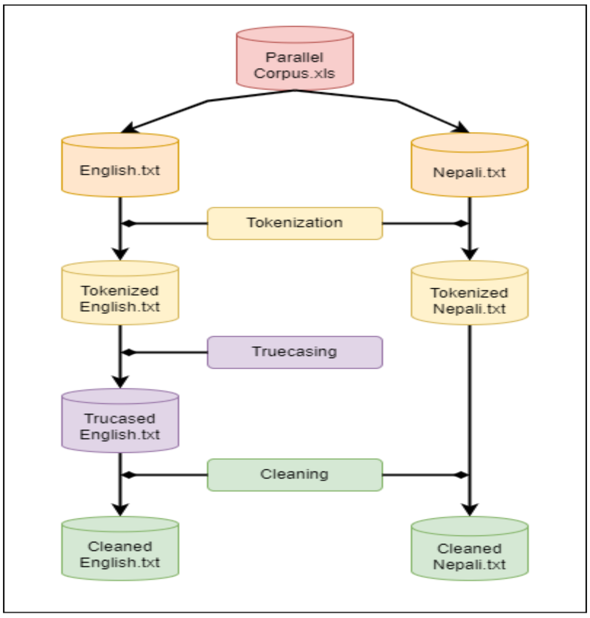

Fig. 1: The steps of preprocessing

In some languages which lack easily perceptible word boundaries, tokenization is decisive, where one token conventionally represents a combination of lexical items. The way such words are tokenized may drastically alter the meaning of the sentence. In this paper, we aim to determine how two popular epitomes of tokenization affect the two prominent MT techniques for the low-resource English-Nepali language combination.

The tokenizer which was employed to evaluate the effect on MT systems is described below.

Moses Tokenizer

The tokenizer comes alongside the Moses toolkit, preserves special tokens like dates and URLs while separating punctuation from words. Additionally, it normalizes characters, including different Unicode variants of quotes. The tokenizer is designed to handle text in any language, ensuring that each token is distinct and standardized for further processing. An example of a sentence tokenized with the Moses tokenizer is shown in Table 1.

Table 1: Moses Tokenization

Language	Text	No. of Tokens
Nepali	यसको सफलता निर◌विवाद छ , जस◌तै यसको लोकप◌रियता पर◌यटक र लन◌डनमा समान छ ।	24
English	Its success is unquestionable , as is its popularity with tourists and Londoners alike	15

Sentence Piece Tokenizer

It is a language-independent, data-driven text tokenizer and detokenizer. It can handle languages without explicit word boundaries since it generates sub word units through unsupervised learning. Sentence Piece can tokenize text into sub words, characters, or other units, making it versatile for various natural language processing tasks (Kudo and Richardson, 2018). It truly is helpful for the processing scripts lacking set word limits or scripts which are complex morphologically. It surely requires the training of specific models. This is for each language. For this purpose, we used the BPE mode as well as set for each corpus's training partition a vocabulary size of 16,000. Table 2 gives an illustration of tokenizing one sentence by using Sentence Piece BPE.

Further, the tokenized data is converted to true cased one, which involves changing uppercase letters to their most probable lowercase counterparts. This conversion reduces the occurrence of sparse data. However, true casing is unnecessary for the Nepali language, as it does not distinguish between capital and lowercase letters. Cleaning the data is another important part of pre-processing, as the task includes removing nonprintable characters, punctuation, long and empty sentences, as well as sentences that are misaligned. These cleaning steps are crucial to ensure the integrity of the training process and prevent interference with the training pipeline. After these pre-processing steps, the corpus is reduced to 26,604 optimal sentences, that is further parted into three files: One for training the MT systems, the next one for tuning or validation of the systems, and the last one for testing the MT systems. The corpus statistics are presented in Fig. 2 and Table 3.

Table 2: Sentence Piece Tokenization

Language	Text	No. of Tokens
Nepali	_यसको _सफलता _निर्विवाद _छ , _जस्तै _यसको _लोकप्रियता _पर्यटक _र _लन्डनमा _समान _छ ।	14
English	_its _success _is _un qu estion able , _as _is _its _popularity _with _tourists _and _Londoners _alike .	15

Table 3: Corpus Description

Language	Type	Size (No. of Sentences)	Size in MB	No. of Tokens
Nepali	Training	23,300	6.60	3,72,753
	Validation	1,757	0.50	27,691
	Testing	1,547	0.40	24,333
English	Training	23,300	2.70	4,60,063
	Validation	1,757	0.21	34,241
	Testing	1,547	0.17	30,041

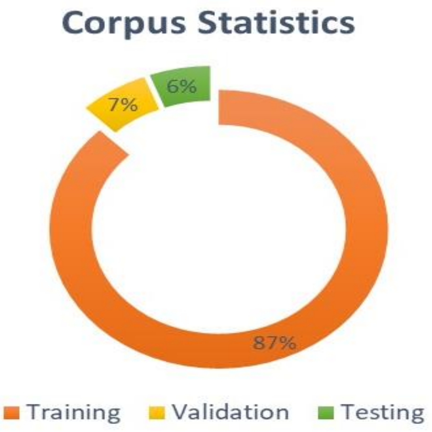

Fig. 2: The corpus statistics

System Setup

SMT Model

The model employs the probability distribution $p(T|S)$ to predict the target language sentence T from a source language sentence S. The distribution is $p(T|S)$ / $(p(S|T) \times p(T))$, where T target is the translation of source S, $p(S|T)$ is the probability and $p(T)$ is the probability of the target word in the language model. The distribution is derived using the Bayesian technique.

The following formula determines the optimal translation, T:

$$p(T|S) = \frac{p(S|T) \times p(T)}{p(S)} \quad (1)$$

$$T = arg\max\left(\frac{p(S|T) \times p(T)}{p(S)}\right) \quad (2)$$

$$T = \arg\max(p(S|T) \times p(T)) \quad (3)$$

The denominator *p(S)* is not present in Eq. 3 and is eliminated due to the constant probability of the source sentence. The translation model supplies *p(S|T)*, while the language model provides *p(T)* (Koehn *et al*., 2007).

With the aid of the open-source toolkit Moses, our SMT system employs the Phrase-based SMT approach (Koehn *et al*., 2007). The parallel source and target sentences are word-aligned using the GIZA++ toolkit after the training corpus has been preprocessed. To construct the language model, the monolingual target language corpus is employed. IRSTLM is being used here, which employs 3-gram modelling. The model architecture is shown in Fig. 3 (Roy *et al*., 2024).

NMT Model

The foundation of NMT systems is built by employing Neural Networks to estimate the conditional probability of the word sequence involving the input source and the output target. The NMT system differs from traditional SMT in the sense that it learns the joint probability without making any assumptions, while typical SMT systems use the Markov assumption to compute the conditional probability. Sequence to sequence Encoder-Decoder based designs are typically used in NMT (Sutskever *et al*., 2014). For a given input source sequence $s = (s_1, \ldots, s_i)$, a basic RNN based model computes a sequence of target outputs $t = (t_1,\ldots, t_j)$:

$$t = arg\,max\,p(t|s) \tag{4}$$

Using the equations given below, the encoding process converts the input sequence s into series of vectors C:

$$h_T = f(s_T\,, h_{T-1}) \tag{5}$$

And:

$$C = q(\{h_1, \ldots, h_{T_x}\}) \tag{6}$$

Here, h_T is the hidden state at any time T, f and q are nonlinear functions, where f is set to LSTM, and C is the context vector formed by the hidden states. The decoder estimates the next word $y_T\,'$ based on context vector C and previously created word sequences ($t_1, \ldots, t_{T\,'-1}$). The equation below calculates the probability of the translation sequence t:

$$p(t) = \prod_{T=1}^{T}(p(t_T \mid \{t_1, \ldots, t_{T-1}\}, C)) \tag{7}$$

$$(p(t_T \mid \{t_1, \ldots, t_{T-1}\}, C) = g(t_{T-1}, s_T, C) \tag{8}$$

Equation 8 is used to compute each conditional probability in Equation 7. Here, g is a nonlinear function that returns the probability of t_T, and s_T is the LSTM RNN's hidden state. It makes use of the SoftMax activation function.

Our NMT system was created using the open-source OpenNMT toolkit (Klein *et al*., 2017). The system uses a Long Short-Term Memory (LSTM) based encoder-decoder technique with two layers of LSTM in both the encoder part and the decoder part, with 500 hidden units. Whether a model is built on a CPU or a GPU will affect how long it takes to train. It took 7.2 hours to train the baseline model on our machine, even after using a 4 GB NVIDIA GeForce GTX 1050Ti GPU to speed up training. For training, the batch size is set to 64, the learning rate and dropout are fixed to 1.0 and 0.3, respectively. The architecture of the proposed model is depicted in Fig. 4 (Roy *et al*., 2024).

Evaluation Metrics

For evaluating our output of machine-generated translation, the standard and widely used automatic evaluation metrics, BLEU and TER, are used.

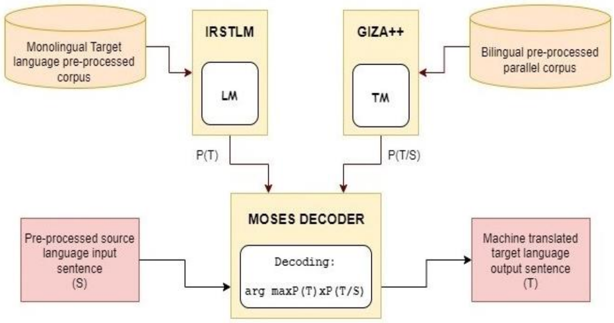
Monolingual Target language pre-processed corpus
IRSTLM
LM
GIZA++
TM
Bilingual pre-processed parallel corpus
P(T)
P(T/S)
MOSES DECODER
Decoding:
arg maxP(T)xP(T/S)
Pre-processed source language input sentence (S)
Machine translated target language output sentence (T)

Fig. 3: The proposed SMT model architecture

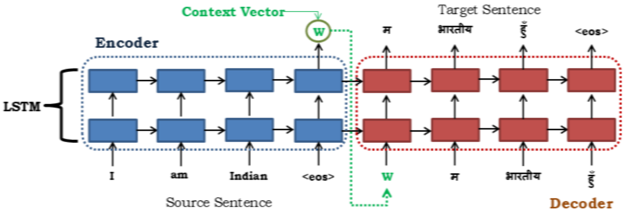

Fig. 4: The proposed NMT model architecture

BLEU Score

The BLEU (Bilingual Evaluation Understudy) score is useful for assessing machine translation precisely because it gives a quantifiable measurement of how well a machine translated text compares to a reference human translation. It assists in determining the quality and correctness of the translation by comparing it to one or more reference translations (Papineni *et al*., 2001). The BLEU score typically ranges from 0 to 1, with 1 representing a perfect match between the translated and reference texts. In practice, it is frequently expressed as a percentage (for example, 0.5847 is equivalent to 58.47%). By applying a brevity penalty, comparing machine translations against reference translations with n-gram precision, and calculating a geometric mean, the BLEU score evaluates the quality of machine translations:

- *N-gram Precision* (p_n) *:* The precision ensures that the machine translation does not receive an inflated score due to repeated words. For each n-gram in the candidate translation, its occurrences are counted and compared to the maximum number of times it appears in any reference translation

$$p_n = \frac{\sum_{C\in Candidates}\sum_{n-gram\in C} Count_{clip}(n-gram)}{\sum_{C\in Candidates}\sum_{n-gram\in C} Count(n-gram)} \tag{9}$$

Where, *Count*$_{clip}$ *(n-gram)* represents the highest number of occurrences of an n-gram in any single reference translation:

- *Brevity Penalty (BP):* The brevity penalty is used to penalize translations that are significantly shorter than the reference:

$$BP = \begin{cases} 1 & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)} & if\ c \leq r \end{cases} \tag{10}$$

Where, *c* is the length of the translated text and *r* is the effective reference length (i.e. length of the closest reference text):

- *Final BLEU Score :* The BLEU score integrates the adjusted precision scores by employing a geometric mean to accommodate various n-gram precisions:

$$BLEU = BP\ \times \exp(\textstyle\sum_{n=1}^{N} w_n \log p_n) \tag{11}$$

Where, p_n is the *n*-gram precision, w_n represents the weight given to each precision score, typically distributed equally (e.g., $w_n = \frac{1}{N}$) and *N* is the highest *n*-gram length considered.

TER Score

The Translation Edit Rate (TER), also known as Translation Error Rate, is a metric used to evaluate the quality of machine translation systems. TER offers a numerical assessment of translation quality, providing an easy method to evaluate how closely a machine translation matches a reference. It precisely measures the edit distance, indicating the effort needed to transform the machine output into a human-quality translation. This is particularly useful in scenarios requiring post-editing, such as professional translation services, where reducing human intervention is essential. TER calculates the number of edits necessary to align a system's output with a reference, including insertions, deletions, substitutions, and shifts. By accounting for these various types of edits, TER captures multiple dimensions of translation errors (Snover *et al*., 2006). This thorough error analysis helps pinpoint specific weaknesses in translation systems, such as problems with word order, vocabulary selection, or missing content. The TER score can be calculated as follows:

$$TER = \frac{Number\ of\ edits}{Average\ number\ of\ reference\ words} \tag{12}$$

Where $Number\ of\ edits$ is the total number of insertions, deletions, substitutions, and shifts needed to convert the translation output to the reference translation, and $Average\ number\ of\ reference\ words$ is the average length of the reference translations.

Results and Discussion

To compare these architectures along with the setups and answer our research questions about which model performs better with this low-resource setting for the Nepali-English language pair, we used automatic evaluation metrics BLEU (Papineni *et al*., 2001) and TER (Snover *et al*., 2006). Here, the SMT_M refer to SMT architecture with Moses tokenizer and SMT_{SP} refers to the same with Sentence Piece tokenizer. Similarly, the NMT_M and NMT_{SP} indicate the NMT architectures with Moses tokenizer and Sentence Piece tokenizer, respectively.

The first four rows of Table 4 designate the scores of English to Nepali translation, while the next four rows signify the scores of Nepali to English translation for these SMT and NMT models respectively. The detailed scores of automatic evaluation matrices on the test sets are elaborated in Fig. 5.

Table 4: Evaluation metrics for Eng-Nep and Nep-Eng Translation models

Translation Direction	Model	BLEU (%)	TER
English to Nepali	SMT_{M}	37.60	51.71
	SMT_{SP}	42.53	64.26
	NMT_{M}	34.89	53.62
	NMT_{SP}	26.80	75.44
Nepali to English	SMT_{M}	23.09	60.58
	SMT_{SP}	50.51	58.27
	NMT_{M}	24.29	58.82
	NMT_{SP}	38.27	77.57

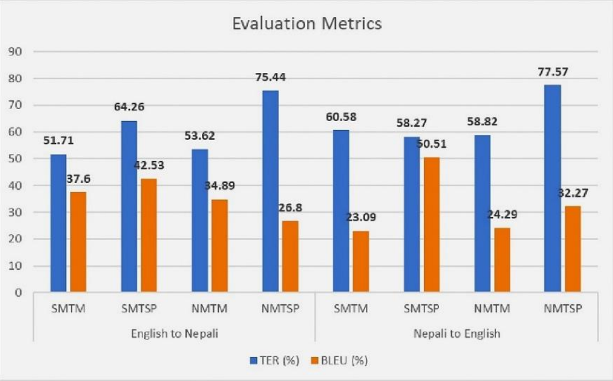

Fig. 5: The scores of automatic evaluation matrices

It presents a comparative analysis of SMT and NMT systems, which we have used Moses and Sentence Piece tokenizers in both English-Nepali and Nepali-English translation directions. The results display that the SMT model with Sentence Piece enhanced (SMTSP) systemically achieves better results: It gets the best results in BLEU index, 42.53 % in English to Nepali and 50.51 % in Nepali to English. Although SMTM achieve a lowest score in the TER index, 51.71% for English-Nepali, but same for Napali-English is again achieved by SMTSP 58.27%. These findings highlight the superiority of subword-based tokenization in improving translation accuracy, especially on languages with rich morphology like the Nepali language. The graph also reveals that

Sentence Piece delivers a tangible improvement against the Moses tokenizer not only outside the SMT framework, but also inside most notably in NMT systems.

Tokenization has a significant impact on translation output, as seen in Tables 5 and 6. The *SMT$_{SP}$* setup outperformed other *SMT$_{M}$*, *NMT$_{M}$* and *NMT$_{SP}$* in terms of BLEU score in English to Nepali translation. For Nepali to English translation *SMT$_{SP}$* produces the best output translation in terms of BLEU score. Regarding the TER score, *SMT$_{M}$* yields the most favourable results for English to Nepali translations, while SMT$_{SP}$ achieves the lowest score for Nepali to English translations, indicating the best performance. We also observed that for both Nepali-English and English-Nepali MT systems, SMT shows significant results in terms of translated output.

Table 5: The sample Nepali to English translation of the Models with source and reference sentences

Moses Tokenizer	
Source Sentence (712)	the temple is biggest in the city and is dedicated to Lord Shiva
Reference Translation	मन ◌ दिर शहरको सबैभन ◌ दा ठूलो हो र भगवान शिवलाई समर ◌ पित छ ।
SMT$_{M}$ Translation	मन ◌ दिर सहरको सबैभन ◌ दा ठूलो हो र भगवान शिवलाई समर ◌ पित छ ।
NMT$_{M}$ Translation	मन ◌ दिर शहरको सबैभन ◌ दा ठूलो हो र बो भगवान शिवलाई समर ◌ पित छ ।
	Sentence Piece Tokenizer
Source Sentence (576)	_it _is _easily _accessible _from _Palai _in _Kottayam _district
Reference Translation	_कोट्टायम _जिल्लाको _पलाईबाट _यो _सजीलो _पहुँचयोग्य _छ ।
SMT$_{SP}$ Translation	_यो _सजीलो _पहुँचयोग्य _छ _कोट्टायम _जिल्लाको _पलाईबाट ।
NMT$_{SP}$ Translation	_कोट्टायम _जिल्लाको _पलाईबाट _सजीलो _पहुँचयोग्य _छ ।

Table 6: The sample English to Nepali translation of the Models with source and reference sentences

Moses Tokenizer	
Source Sentence (263)	*एक शताब्दी अघि डिकेन्सको समयमा, हल अधिक केन्द्र बिन्दु थियो ।*
Reference Translation	*a century ago in dickens & apos; day , the hall was a more focal point*
SMT$_{M}$ Translation	*a century ago in dickens & apos; day , the hall more focal point .*
NMT$_{M}$ Translation	*a century ago in dickens & apos ; , the hall was a more focal point*
SentencePiece Tokenizer	
Source Sentence (1200)	*_सिना गोग _र _चर्च _नजिकै _मन्दिर _र _मस्जिद _छ ।*
Reference Translation	*_near _the _synagogue _and _the _church _ there _is _a _temple and _masque*
SMT$_{SP}$ Translation	*_near _the _synagogue _and _the _church _is _a _temple and _masque*
NMT$_{SP}$ Translation	*_in _heart _and _church _ is _near _the _temple and _masque*

Our experiment of English to Nepali translation revealed some situations where Statistical Machine Translation (SMT) systems performed better compared to their counterparts that are dependent on Neural Machine Translation (NMT). One primary such situation where SMT has been reported to do relatively well in areas where the parallel corpus is hardly available, and this observation is quite contrary to the case with NMT, which requires extensive parallel data to effectively carry on the learning process to represent and consequently generalise. In SMT, phrase-based models, which process sparse data well, are especially resistant to the limitation of the train data. Another important factor is the high complexity of neural architectures, which makes them prone to poor performance in low-resource settings due to noisy performance and overfitting behaviour when presented with low-resource datasets. Conversely, due to the interpretable, modular character of Statistical Machine Translation (SMT) frameworks, a higher level of consistency in the resulting output can be expected. In the case of Machine Translation (MT) systems, model architecture and data availability should be considered, especially for under-resourced languages like Nepali.

The experiment also observed improvement of performance when using Sentence Piece over Moses for tokenization. This is because Moses tokenizer often relies on whitespace and punctuation to tokenize sentences, but Sentence Piece employs a data-driven, subword-based segmentation approach, avoiding language-specific tokeniztion. Due to this independence of preset decomposition principles, the algorithm is far more generic and effective with languages that exhibit morphological richness, in which a single lexeme can take an extensive range of inflected or compounded surface forms, like Nepali, where suffixes and agglutinative structures increase vocabulary size and sparsity if treated at the word level. Sentence Piece's language-free and sub word-based approach offers clear benefits when utilized on SMT in morphologically complex and resource-resource-constrained languages like Nepali. In particular, the sub word units make it possible to find alignment better and generate phrase tables that are much larger and denser, and therefore, make the models more robust to unseen lexicalizations. Thus, translations generated by the systems equipped with Sentence Piece tokenization outperform the ones generated with Moses-tokenized baselines.

Conclusion and Future Work

The article evaluates the performance of SMT and NMT systems using Moses and Sentence Piece tokenizers for translating Nepali-English text that is particular to the tourism domain. The endeavour entails creating a bilingual parallel corpus of size 29k, which is made up of manually translated monolingual English sentences by a native Nepalese speaker. According to our research, the SMT system with Sentence Piece tokenizer performs better translations for both the English to Nepali and Nepali to English setup when trained with identical training and test data than other SMT and NMT setups do in this resource constrained environments respectively. The rich inflectional and morphological features of the Nepali language provide some difficulties and have a negative impact on the performance of our systems. Given that significant effort has not yet been done on this language pair, the results of automatic evaluation metrics are ideal for translations from Nepali to English as well as vice versa.

In the future, extending the corpus size will be required to attain an improved outcome. Comparing other segmentation techniques, like character separation or fixed n-grams, might also be intriguing. Furthermore, we will also focus our research on the verification of whether similar results are obtained when conducting these studies again using the general domain training data utilized for these languages. Additionally, we also intend to investigate many alternative feasible machine translation methods that could enhance the translation accuracy of the chosen low-resource language pair.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

No funding or financial assistance has been received by the authors for this work.

Author's Contributions

Amit Kumar Roy: Conceptualization, Formal analysis, Investigation, Methodology, and Writing original draft.

Bipul Syam Purkayastha: Supervision, Validation, and Writing review.

Ethics

The present study represents an original research effort. The corresponding author confirms that the coauthor has reviewed and approved the manuscript, without raising ethical issues. And guarantees that this specific manuscript hasn't been previously published and that no ethical issues exist.

References

Acharya, P., & Bal, B. K. (2018). A Comparative Study of SMT and NMT: Case Study of English-Nepali Language Pair. *Proceedings of the 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, 90–30. https://doi.org/10.21437/sltu.2018-19

Ahmed, M. A., Kashyap, K., & Sarma, S. K. (2023). Tokenization effect on neural machine translation: an experimental investigation for English-Assamese. *Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–7. https://doi.org/10.1109/icccnt56998.2023.10307971

Bal, B. K. (2004). *Structure of Nepali grammar*.

Bista, S. kumar, bhatta, J., & Keshari, B. (2007). Dobhase: online english-to-nepali machine translation system. *Proceedings of the Innovative Applications of Information Technology for the Developing World*, 330–339. https://doi.org/10.1142/9781860948534_0052

Chaudhary, B. K., Bal, B. B., & Baidar, R. (2020). Efforts towards developing a Tamang-Nepali machine translation system. *Proceedings of the 17th International Conference on Natural Language Processing (ICON 2020)*, 281–286.

Devi, C. S., Roy, A. K., & Purkayastha, B. S. (2023). Parts of speech tagged phrase-based statistical machine translation system for english → mizo language. *SN Computer Science*, *4*(6), 841. https://doi.org/10.1007/s42979-023-02309-8

Domingo, M., García-Martínez, M., Helle, A., Casacuberta, F., & Herranz, M. (2023). How Much Does Tokenization Affect Neural Machine Translation? *Computational Linguistics and Intelligent Text Processing. CICLing 2019*, *13451*, 545–554. https://doi.org/10.1007/978-3-031-24337-0_38

Google. (2025). *Google Translate*. https://translate.google.com/?sl=en&tl=ne&op=translate

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. *Proceedings of the First Workshop on Neural Machine Translation*, 28–39. https://doi.org/10.18653/v1/w17-3204

Koehn, P., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., & Moran, C. (2007). Moses: Open-source toolkit for statistical machine translation. *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions - ACL '07*, 177–180. https://doi.org/10.3115/1557769.1557821

Kudo, T., & Richardson, J. (2018). Sentence Piece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71. https://doi.org/10.18653/v1/d18-2012

Laskar, S. R., Pakray, P., & Bandyopadhyay, S. (2019). Neural Machine Translation: Hindi-Nepali. *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, 202–207. https://doi.org/10.18653/v1/w19-5427

Lohar, P., Maja, P., Haithem, Alfi, & Andy, Way. (2019). A systematic comparison between SMT and NMT on translating user-generated content. *Proceedings of CICLing 2019: 20th International Conference on Computational Linguistics and Intelligent Text Processing*, 7–13.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). BLEU: a Method for Automatic Evaluation of Machine Translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Poudel, S., Bal, Bal Krishna, & Acharya, Praveen. (2024). Bidirectional English-Nepali machine translation (MT) system for legal domain. *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-Resourced Languages (LREC-COLING 2024*, 53–58.

Roy, A. K., Purkayastha, B. S., & Paul, S. (2024). A Bidirectional Statistical Machine Translation System for Exploring the Performance of the Low Resource Language Pair English-Nepali. *IEEE*, 1–6.

Roy, A. K., Purkayastha, & B. S., Devi, C. S (2023). Machine Translation Systems for Official Languages of North-Eastern India: A Review. *INFOCOMP Journal of Computer Science, 23*, 301–315. https://doi.org/10.1007/978-3-031-48879-5_23

Singh, T. D., & Hujon, A. V. (2020). Low resource and Domain Specific English to Khasi SMT and NMT Systems. *Proceeding of the 2020 International Conference on Computational Performance Evaluation (ComPE)*, 733–737. https://doi.org/10.1109/compe49325.2020.9200059

Snover, M. G., Dorr, B. J., Schwartz, R., & Micciulla, L. (2006). TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate. *Association for Machine Translation in the Americas.*, 223–231. https://doi.org/10.1007/s10590-009-9062-9

Stasimioti, M., Sosoni, V., Kermanidis, K., & Mouratidis, D. (2019). Interactive adaptive SMT versus interactive adaptive NMT: a user experience evaluation. *Machine Translation*, 117–134. https://doi.org/10.1007/s10590-019-09230-z

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Proceeding of the Advances in Neural Information Processing Systems 27 (NIPS 2014*, 3104–3112. https://doi.org/https://doi.org/10.48550/arXiv.1409.3215