BERT-Based Nepali Grammatical Error Detection and Correction Leveraging a New Corpus

Sumit Aryal
DOECE
Pulchowk Campus, IOE
Lalitpur, Nepal
sumitaryal310@hotmail.com

Anku Jaiswal
DOECE
Pulchowk Campus, IOE
Lalitpur, Nepal
anku.jaiswal@pcampus.edu.np

***Abstract*—Nepali GEC plays a crucial role in improving the quality of written Nepali text. An annotated corpus of Nepali sentences along with sentences generated by augmenting correct sentences to generate a diverse range of grammatical errors are used for training. The augmentation is done by identifying the Part of Speech tag and root words of verbs and adjectives using Lemmatizer. This study uses BERT models MuRIL and NepBERTa to fine-tune for the GED task of Nepali text. The models performances were assessed using accuracy and training/validation loss, providing a comprehensive assessment of the model's effectiveness in error detection for the Nepali Language which forms the crucial step for GEC. The GEC system developed here, makes use of MLM models of both MuRIL and NepBERTa to predict the mask tokens in input erroneous sentence and thus gives the suggestions which are filtered by the GED model.**

***Index Terms*—Nepali Grammar Correction, Nepali Grammar Error Detection, Nepali GEC Corpus, BERT**

I. Introduction

Nepali, the official language of Nepal is spoken by millions of people as their native tongue. Proper grammar usage is essential for effective communication and written expression in Nepali. However, due to the complexity of Nepali grammar rules, it is common to encounter grammatical errors in written texts, which can hinder comprehension and negatively impact the quality of communication. Nepali is written in the Devanagari script. There are 13 vowels and 36 consonants in the Nepali language.

This research introduces a Nepali Grammatical Error Correction system aimed at improving accuracy and quality of written Nepali, sentence by sentence. Instead of providing immediate corrections, the system initially checks for errors and subsequently offer suggestions and corrections to rectify these issues. This approach assists users in producing grammatically accurate Nepali text by leveraging the usage of masked language models in suggesting correct sentences.

II. Literature Review

The Nepali language is categorized as a "low-resource" language, indicating that there has been limited research conducted in the field of Nepali Language Processing. The amount of work that has been done in Nepali Natural Language Processing is very scarce. The works done include Part-of-speech(POS) tagging [6], morphological analysis [3], sentiment analysis [8] and word embeddings [7]. The amount of work that has been done in the field of Nepali Grammar Correction is minimal. Currently, no previous models exist for Nepali GEC. This research is one of the first attempts at developing models for Nepali GEC using BERT-based architectures. While models do exist for other resource rich languages, there are unique challenges in adapting those models to Nepali due to its low-resource nature and complex grammatical structure.

However, there is a barrier to the development of the Nepali GEC task i.e. the lack of publicly available large-scale parallel corpus for the same task. To this barrier, firstly a large-scale parallel corpus is created for the Nepali GEC task and then propose a method which is solely based on BERT for the Nepali Grammar Correction task.

III. Corpus Creation

The lack of parallel corpus is the main setback for the development of effective Nepali GEC system. In recent years, there has been availability of large parallel corpus for high resource languages like English but it is not the same for languages like Nepali as it is a low resource language. Therefore, the initiative is taken to develop a large scale parallel corpus for Nepali GEC task.

To do so, five different types of Nepali grammatical mistakes are identified which include Verb Inflection, Homophones, Punctuation, Sentence Structure and Sentence Fragments. Furthermore, sentence fragments can be sub divided into three more classes as missing subject, missing main verb and missing auxiliary verb. The process is described as below.

1) Verb Inflection
   Verb inflection refers to the modification of a verb that expresses a different grammatical form of the verb. By undergoing verb inflection, the relationship between the verb and its associated subject will be disrupted which creates an error in the sentence. An example is illustrated in Table I.

979-8-3503-7952-5/24/$31©2024 IEEE

TABLE I
Verb Inflection

Incorrect	Correct
बाबाले सर्प बारे अरू बढी केही बोल्छ ।	बाबाले सर्प बारे अरू बढी केही बोल्नुभएन ।

2) Homophones Error
Homophones are words that sound alike but have different meanings and spellings. They play a significant role in communication often leading to confusion due to the similarity in pronunciation. The use of wrong homophones disrupts the sentence meaning leading to incorrect sentences. An example is illustrated in Table II.

TABLE II
Homophones Error

Incorrect	Correct
दुर्गम क्षेत्रका अरू जनताले पनि उनीहरू बाट पात सिक्नुपर्छ ।	दुर्गम क्षेत्रका अरू जनताले पनि उनीहरू बाट पाठ सिक्नुपर्छ ।

3) Punctuation Error
Punctuations are the symbols that aid in clarity, structure, and comprehension. In Nepali language punctuation marks such as commas(,), full stop( ), question mark(?), exclamation mark(!), and others. The incorrect use of punctuation leads to misunderstanding or ambiguity. They affect the clarity of the sentence. An example is illustrated in Table III.

TABLE III
Punctuation Error

Incorrect	Correct
तर यसका लागि निजी स्कूलहरू मात्र दोषी छैनन् ?	तर यसका लागि निजी स्कूलहरू मात्र दोषी छैनन् ।

4) Sentence Structure
Sentence structure refers to the arrangement of words and phrases to form coherent and meaningful sentences. The incorrect arrangement of words in a sentence results in a grammatically incorrect sentence. The incorrect sentence usually changes the meaning of the sentence and makes it difficult to understand the sentence. An example is illustrated in Table IV.

TABLE IV
Sentence Structure Error

Incorrect	Correct
एकै कोठा मा सुत्ने दाजुभाइ पनि बीच कुराकानी हुन छाडेको छ ।	एकै कोठा मा सुत्ने दाजुभाइ बीच पनि कुराकानी हुन छाडेको छ ।

5) Sentence Fragments
Sentence Fragments are incomplete collections of words that lack a subject or a verb, which doesn't form a complete sentence. When writers omit necessary components, these fragments can cause confusion or ambiguity in communication, potentially leading to misunderstanding. It is further divided into two parts as:

a) Subject Missing: This sentence fragment contains those sentences in which the subject is not included. In case the subject is missing, we cannot remove the noun as it plays a crucial role in conveying the intended message. So only the pronoun can be removed. A pronoun is a word that substitutes for a noun or noun phrase. Pronouns can refer to people, places, things, or ideas previously mentioned or understood in the context of the conversation or text. The absence of the pronoun in a sentence leads to ambiguity or a lack of clarity regarding the subject or object being referenced. An example is illustrated in Table V.

TABLE V
Pronoun Error

Incorrect	Correct
सूचना क्रान्तिको दुनिया मा मख्ख परेर ठूलो भ्रान्ति पालिरहेका छौं ।	हामी सूचना क्रान्तिको दुनिया मा मख्ख परेर ठूलो भ्रान्ति पालिरहेका छौं ।

b) Verb Missing: This sentence fragment contains those sentences in which the verb is missing. It is further divided into two parts which are described as follows.

i) Main verb Missing Error
Main verbs, also known as principal verbs or lexical verbs, are fundamental components of sentences that convey the action or state of being. Unlike auxiliary verbs (helping verbs), which assist the main verb in forming verb phrases, main verbs stand alone and carry the primary meaning in a sentence. An example is shown in Table VI.

TABLE VI
Main Verb Missing Error

Incorrect	Correct
यो टेक्निक पनि प्रभाववाद सँग सम्बद्ध ।	यो टेक्निक पनि प्रभाववाद सँग सम्बद्ध छ ।

ii) Auxiliary Verb Missing Error
Auxiliary verbs, also known as helping verbs, are used alongside main verbs to add functional or grammatical meaning to a sentence. Absence of the auxiliary verb in a sentence results in a loss of information regarding aspects like tense, mood, voice, or aspect, leading to ambiguity or an incomplete expression of the action or state described in the sentence. An example is demonstrated in Table VII.

TABLE VII
Auxiliary Verb Missing Error

Incorrect	Correct
खाद्यान्नकै हक मा पनि सके सम्म खेर गरी खाना नै नबनाए हुने ।	खाद्यान्नकै हक मा पनि सके सम्म खेर जाने गरी खाना नै नबनाए हुने ।

A. Data Sourcing and Preprocessing

The raw data is sourced from a publicly available corpus named "A LARGE SCALE NEPALI TEXT CORPUS" [1]. A data cleaning process was undertaken to refine the collected data. This process involved discarding sentences that did not fall within a 3-20 word count, while also accounting for punctuation marks. Additionally, any characters not conforming to the Devanagari Script were eliminated, and English numerals were converted to Nepali numerals for consistency within the text. Following this, a step was taken to extract unique sentences to remove redundancy. Sentences containing only one parenthesis or single or double quotes were also discarded.

B. Data Augmentation

The different types of errors discussed above are generated on the collected data employing noise injection techniques. Each sentence is regarded as a set of words, denoted by S = $\{W_1, W_2, ..., W_{N-1}, W_N\}$ where N represents the sentence length which is a positive integer. Every word $W_i \in S$ is viewed as a collection of Nepali characters: $W_i = \{C_1, C_2, ..., C_{M-1}, C_M\}$ where M stands for the length of the word which is also a positive integer. It is ensured that each artificially flawed sentence contains only one mistake. However, some sentences may contain multiple words with errors, leading to several flawed versions. Therefore, this process yields one correct sentence alongside multiple incorrect forms.

Firstly, the verbs are extracted from the data using the POS Tagger [6]. After this, the lemma of the verbs are extracted using the hybrid approach of the Lemmatizer [3]. From the verbs and lemma, the suffixes are collected which is as A = $\{a_1, a_2, ..., a_D\}$ where $a_i \in A$ is the i^{th} suffix. The elements within the set A are organized into sub-lists based on the similarity of the suffix. These sub-lists are represented as $D_j = [d_1, d_2, ..., d_F]$ such that $d_i \in A$ and D_j is the j^{th} sub-list. Then, a dictionary is created from the similar groups, which is as D = $\{G_1 : D_1, G_2 : D_2, ..., G_N : D_N\}$ where G_i is the i^{th} group name and D_i is its corresponding list of similar suffixes. Then we iterate each verb of the sentence and determine whether it is found in the dictionary or not. If found the suffix of the verb is replaced with a similar suffix from the dictionary.

For the Homophones error, the website [2] was scraped for the homophones and missing homophones were added manually. A dictionary is created for the homophones such as H = $\{H_1 : P_1, H_2 : P_2, ..., H_i : P_i\}$ where h_j is the j^{th} word and p_j is its respective homophone. To generate the error, we iterate through each word W_i in a sentence S, and if the word is found in the homophones' dictionary key, the word is replaced by its respective value. This creates an erroneous version of the correct sentence.

In order to generate punctuation error, we go through each character C_i of the sentence S, and if a punctuation symbol is found, an error is generated with a random probability. In case of full stop( ), question mark(?) and exclamation mark(!), they are either removed or replaced with the one which has not occurred. In case of other symbols, they are removed with a random probability. Alternatively, we induce errors in sentence structure by randomly swapping the positions of two words within a sentence with a random probability.

In order to generate error for the missing subject and the missing verb, similar approach is used. The use of POS taggers were helpful in generating corresponding POS tags for each word W_i in the sentence S, denoted as $S_t = \{pt_1, pt_2, .. , pt_n\}$ for n words in S. Then, we iterate through the tag set S_t and if the POS tag indicates a pronoun, then the pronoun is removed to generate pronoun missing error. In the POS tagging process, auxiliary verbs and main verbs aren't distinguished. To determine the main and auxiliary verbs in a sentence, a list of verbs within the sentence is extracted. The final verb that completes the sentence structure is considered the main verb, while the preceding verbs are regarded as auxiliary verbs based on specific rules crafted for this purpose. So by removing the auxiliary verb, auxiliary verb missing error is generated and by removing the main verb, main verb missing error is generated.

C. Corpus Statistics

The developed Nepali GEC corpus comprises seven different types of errors. Among these, verb inflection error is the most frequent (39.39%), while pronoun error is the least frequent (3.89%). The prevalence of high instances of verb inflection error is because it also includes errors related to subject-verb agreements, numbers and other cases where some words might end up wrong with the verb inflected. The whole error inflection statistic can be summarized by Table VIII:

The amount of different error types is supported by the fact that none of them were introduced manually; all the instances have been crafted automatically based on the underlying corpus and predefined suffixes which are carefully extracted. Additionally, errors related to word choice is not that common in the corpus which is understandable considering the fact that the Nepali language has a relatively small number of homonyms. Conversely, the most prominent error type in the corpus is related to verb inflection which is expected considering Nepali language has a extensive range of verb inflection suffixes and also due to the fact that the inflection covers a wide scope of errors.

The augmented errors in the dataset were generated using various augmentation techniques as discussed above

TABLE VIII
Statistics of the Nepali GEC Corpus.

Error Types	Number of Instances	Percentage
Verb Inflection	3202676	39.39
Pronouns	316393	3.89
Sentence Structure	1001038	12.31
Auxiliary Verb Missing	1031388	12.69
Main Verb Missing	1031274	12.68
Punctuation Errors	1044203	12.84
Homophones Errors	503524	6.2
Total	8130496	100

which are commonly observed in real-world texts. Although synthetically generated, the errors were designed to replicate typical grammatical mistakes encountered in Nepali, making the corpus diverse and representative.

IV. Methodology

The proposed method is structured around a two-step process which uses a pre-trained BERT model. Leveraging such pre-trained models would reduce the burden of pre-training the model on large Nepali corpus as it takes a lot of computing time and high computational resources. The process of fine-tuning the model on our corpora for Nepali GEC becomes essential. It is then fine-tuned with the generated corpora for correct and erroneous sentences for single sentence classification to identify whether the sentence is grammatically correct or not. This fine-tuning will give us the GED model. We would use the Masked Language Model of the BERT model to come up with alternate sentences and use the fine-tuned GED model to come up with the correct suggestions.

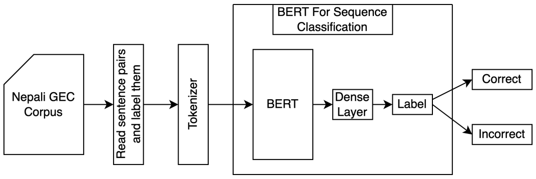
Nepali GEC Corpus
Read sentence pairs and label them
Tokenizer
BERT For Sequence Classification
BERT
Dense Layer
Label
Correct
Incorrect

Fig. 1. Flow of how GED model works.

The constructed corpora is read and labelled as correct and incorrect and passed to the tokenizer which preprocesses and tokenizes the raw text input into a format suitable for input for our BERT model. The model iterates over the corpus, learning its intricacies and generating an intermediate value as BERT is an encoder model. The output is passed through a Dense layer which is a classification layer helping in classifying whether the input sentence is correct or not as illustrated in Figure 1. The model learns from the sentences and their corresponding labels for classification purpose.

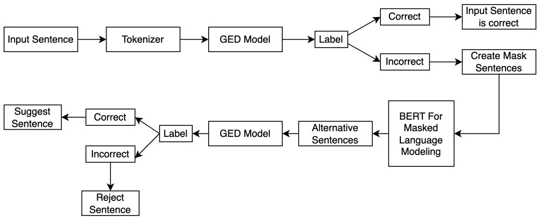

Fig. 2. Flow of how the GEC system works.

The GED model forms the core of our GEC ecosystem. It serves as the first step in identifying whether a sentence contains grammatical errors. Following this, the masked sentences are created where [MASK] token is injected to the parts of the sentence. Then the BERT Masked Language Model is used to generate sentences by predicting the [MASK] tokens in all the sentences. For each incorrect sentence, we inject [MASK] in two different ways i.e. masking each word and adding [MASK] token in each space of the sentence. The generated sentences are yet again passed to the GED model to detect whether the sentences are correct or not. The incorrect labeled sentences are discarded whereas the correct sentences are considered as grammatically correct sentence as shown in Figure 2. This is how Nepali Grammatical Error Correction is achieved. Two BERT based models MuRIL [5] and NepBERTa [3] are used for this pupose.

V. Experimental Analysis

A. Nepali GEC Corpus

The Nepali GEC parallel corpus consists of 8.1M source-target pairs. The dataset is partitioned into two different sets i.e. training and validation sets. This sentence pair was split where 95% of the entire dataset i.e. 7,723,971 pairs were used to construct the training dataset and remaining 5% i.e. 406,525 pairs were used to construct the validation dataset. As the dataset is in the form of pairs of correct sentences and incorrect sentences, ”label 0” and ”label 1” were given to the correct and incorrect sentences respectively. Table IX shows the characteristic of the training and validation dataset.

TABLE IX
Description of Dataset

Split	Number of Correct Sentences	Number of Incorrect Sentences	Total Sentences
Train	2,568,682	7,514,122	10,082,804
Valid	365,606	405,905	771,511

B. Baseline Models

- MuRIL [5]
  MuRIL stands for Multilingual Representations for

Indian Languages. It is a multilingual model developed by Google Research which aims to provide natural language understanding capabilities across a diverse set of Indian languages. By leveraging large-scale pre-training techniques, MuRIL enables applications to process and understand text in multiple Indian languages, which also includes Nepali language. Thus, this model is fine-tuned for the GED model and its MLM variant is also used for generating correct suggestions.

- NepBERTa [3]
  NepBERTa is a variant of the BERT which is specifically tailored for the Nepali language. Built upon the Transformer architecture, NepBERTa is pre-trained on a large corpus of Nepali text data, allowing it to capture contextual language representations effectively. Hence, it is fine-tuned on the GEC dataset to get the GED model. The MLM variant is also used for generating the correct suggestions.

C. Performance Evaluation

- Accuracy
  Accuracy represents the proportion of correctly classified instances over the total number of instances. It is calculated by dividing the number of correctly classified instances by the total number of instances and multiplying the result by 100 to express it as a percentage. Mathematically, accuracy can be expressed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Processing Time
  Calculate how long it takes the system to examine and fix grammar mistakes in a given input. It shows how quickly and effectively the system responds.

D. Hyperparameters

The MuRIL model was fine-tuned for a single epoch with the following hyperparameters: train/valid batch size of 256, Cross Entropy Loss, and AdamW optimizer (learning rate = $5\mathrm{e}^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1\mathrm{e}^{-8}$). The same hyperparameters are used for the NepBERTa model but the model was trained for 2 epochs.

VI. Results

The MuRIL and NepBERTa models were successfully fine-tuned on 2,568,682 correct and 7,514,122 incorrect labeled sentences, and validated on 365,606 correct and 405,905 incorrect labeled sentences respectively. This fine-tuning task was done for GED tasks which forms the core of the GEC engine. The training information for these models are presented in Table X.

The performance shown by MuRIL and NEPBERTa for the GED task is summarized by the Table XI.

TABLE X
Training information of the models.

Model	**Number of Epochs Trained**	**Number of Trainable Parameters**	**Tokenizer Length**
MuRIL	1	237,557,762	197285
NepBERTa	2	109,514,298	30523

TABLE XI
Performance of MuRIL and NEPBERTa.

Model	**Training Loss**	**Validation Loss**	**Accuracy**
MuRIL	0.2427	0.2177	91.15
NEPBERTa	0.2776	0.3446	81.73

Although MuRIL was trained for only one epoch, it outperformed NepBERTa, which was trained for two epochs, in GED tasks. This is because the complexity of MuRIL was much higher than that of NepBERTa as shown by the number of trainable parameters in both the models. Thus, MuRIL effectively demonstrates its ability to learn about the patterns involved in both correct and erroneous sentences. Hence for the GED task which forms the core of our GEC engine, the fine-tuned MuRIL model is used. For predicting the `[MASK]`, either of the two pre-trained model can be used. These models generate possible sentences, which are then checked by the GED model for correctness.

Table XII shows an example of the grammatical error corrected by the system.

TABLE XII
Example Result

Input sentence	नयाँ संविधान कार्यान्वयनको लागि निश्चित समयसीमाभित्रै तीन तहको निर्वाचन गर् ।
Baseline Output	नयाँ संविधान कार्यान्वयनको लागि निश्चित समयसीमाभित्रै तीन तहको निर्वाचन गर्नुपर्नेछ ।
NepBERTa as MLM Output	नयाँ संविधान कार्यान्वयनको लागि निश्चित समयसीमाभित्रै तीन तहको निर्वाचन हुनुपर्छ । नयाँ संविधान कार्यान्वयनको लागि निश्चित समयसीमाभित्रै तीन तहको निर्वाचन गर्यौं ।
MuRIL as MLM Output	नयाँ संविधान कार्यान्वयनको लागि निश्चित समयसीमाभित्रै तीन तहको निर्वाचन हुनेछ । नयाँ संविधान कार्यान्वयनको लागि निश्चित समयसीमाभित्रै तीन तहको निर्वाचन गर्यौं ।

As seen in Table XII, a verb correction was applied in the sentence. “गर्” is an incorrect form of verb for the input sentence and thus, this was changed to correct verb forms such as “गर्नुपर्नेछ, हुनुपर्छ, हुनेछ, गर्यौं” based on the encoded information of the masked sentences. The initial verb “गर्” is a low-honorific verb form, but the sentence context requires a normal level of honorific form. Hence, the verb is changed to its correct honorific form, which reflects the required formality, suitable for the sentence context.

The processing time of the sentences depends upon the length of the sentence. Upon visualizing, it was found that the time taken by the system to generate suggestions is linearly dependent on the sentence length. From the corpus, 10 sentences were randomly selected for each sentence length and the average processing time was calculated for each length. Both MuRIL and NepBERTa were evaluated on this metric which is demonstrated in the Figure 3 and 4 respectively.

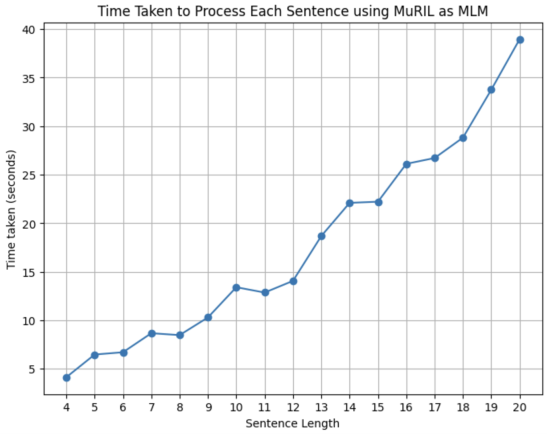

Fig. 3. Processing time for MuRIL as MLM

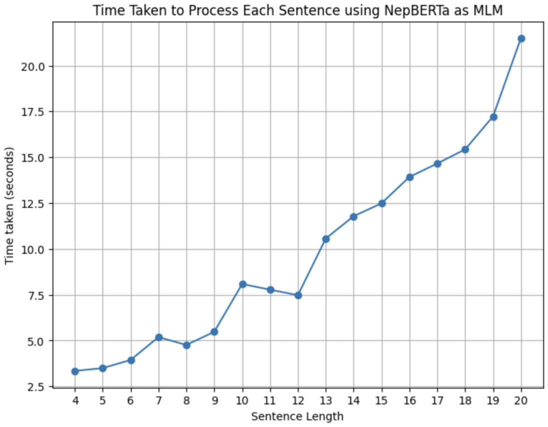

Fig. 4. Processing time for NepBERTa as MLM

Slight variations can be seen as the sentence undergoes the tokenization process. In the tokenization process, a single word can be decomposed into multiple tokens, so more masked sentences are generated which increases the number of sentences to be processed by the BERT MLM model hence fluctuating the processing time.

VII. Conclusion

This study introduces a comprehensive approach to Nepali Grammatical Error Correction (GEC) which employs BERT-based models, MuRIL and NepBERTa. Both models are fine-tuned on a large parallel corpus of Nepali sentences, augmented with diverse grammatical errors. The results demonstrate the effectiveness of these models in detecting and correcting errors in Nepali text, with MuRIL showing superior performance despite being trained for fewer epochs. This work marks a significant step toward improving the quality of written Nepali communication.

Given the absence of publicly available, human-annotated real-world data for Nepali GEC, the evaluation was conducted using synthetically generated grammatical errors. While this data provides a narrow environment for model training and evaluation, further research should focus on testing these models on real-world Nepali texts to ensure their robustness and applicability in practical scenarios.

Additionally, while the augmented corpus offers a wide range of grammatical errors, it was generated synthetically, introducing certain limitations. Automated generation of errors may not be the actual portrayal of human errors, potentially leading to biases in the model's performance.

Despite these limitations, this research lays a foundation for the Nepali Grammatical Error Correction task and provides a baseline for future advancements. By addressing the challenges of low-resource language processing through a creation of large-scale parallel corpus, this study offers promising direction for the further development and enhancement of GEC systems for Nepali language.

Data Availability

Nepali GEC parallel corpus dataset will be made available on request.

References

[1] R. Lamsal, "A Large Scale Nepali Text Corpus," IEEE Dataport, 2020. [Online]. Available: https://dx.doi.org/10.21227/jxrd-d245.

[2] B. BC, "Homophones in Nepali Language," 2020. [Online]. Available: https://bhupendrabc.blogspot.com/2020/11/blog-post_25.html.

[3] D. P. D. L., "Nepali Lemmatizer," 2020. [Online]. Available: https://github.com/dpakpdl/NepaliLemmatizer.

[4] S. Timilsina, M. Gautam, and B. Bhattarai, "NepBERTa: Nepali Language Model Trained in a Large Corpus," in Proc. 2nd Conf. Asia-Pacific Chapter Assoc. Comput. Linguistics and 12th Int. Joint Conf. Nat. Lang. Process. (Vol. 2: Short Papers), Nov. 2022, pp. 273–284. [Online]. Available: https://aclanthology.org/2022.aacl-short.34.

[5] S. Khanuja et al., "MuRIL: Multilingual Representations for Indian Languages," CoRR, vol. abs/2103.10730, 2021. [Online]. Available: https://arxiv.org/abs/2103.10730.

[6] E. 911, "Nepali POS Tagger," 2018. [Online]. Available: https://github.com/e911/Nepali-POS-Tagger.

[7] P. Koirala and N. B. Niraula, "NPVec1: Word Embeddings for Nepali - Construction and Evaluation," in Proc. 6th Workshop Rep. Learn. Nat. Lang. Process. (RepL4NLP-2021), Aug. 2021, pp. 174–184. doi: 10.18653/v1/2021.repl4nlp-1.18.

[8] O. M. Singh, S. Timilsina, B. K. Bal, and A. Joshi, "Aspect Based Abusive Sentiment Detection in Nepali Social Media Texts," in Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Min. (ASONAM), 2020, pp. 301–308, doi: 10.1109/ASONAM49781.2020.9381292.