**Tribhuvan University**

**Institute of Science and Technology**



**A Dissertation Report**

**On**

**“Sentiment Analysis of Social Media Texts in Nepali Using Transformers”**

**In the partial fulfillment of the requirements for the Master’s Degree in Computer Science and Information Technology**

**Submitted by:**

**Regan Maharjan**

**Roll No. 17/075**

**Under the Supervision of**

**Asst. Prof. Bikash Balami**

**Co-Supervision of**

**Asst. Prof. Tej Bahadur Shahi**

**Submitted to:**

**Central Department of Computer Science and Information Technology**

**Tribhuvan University**

**Kirtipur, Nepal**

**August 2023**

Table of Contents

[Table of Contents i](#_Toc143870932)

[List of tables ii](#_Toc143870933)

[List of Figures iii](#_Toc143870934)

[List of Abbreviations iv](#_Toc143870935)

[ABSTRACT v](#_Toc143870936)

[1. Introduction 1](#_Toc143870937)

[**1.1** **Introduction** 1](#_Toc143870938)

[**1.1.1 Sentiment Analysis** 1](#_Toc143870939)

[**1.1.2 Transformer** 1](#_Toc143870940)

[**1.2 Problem Statement** 2](#_Toc143870941)

[**1.3 Objectives** 3](#_Toc143870942)

[**1.4 Scope and Limitation** 3](#_Toc143870943)

[**1.5 Organization of the report** 3](#_Toc143870944)

[2. Literature Review 4](#_Toc143870945)

[2.1 Transformers 4](#_Toc143870946)

[2.2 Sentiment Analysis of Nepali Texts 4](#_Toc143870947)

[3. Methodology 5](#_Toc143870948)

[4. Implementation 5](#_Toc143870949)

[5. Result and Analysis 5](#_Toc143870950)

[6. References 6](#_Toc143870951)

# List of tables

[Table 1 Nepali numerals, consonants, where among 36 consonants, 33 are unique consonants and 3 are combination of some unique consonants, and vowels [2] 1](#_Toc144035725)

# List of Figures

[Figure 1. The transformer – model architecture [2] 2](file:///E:\Repositories-Personal\Thesis%20Reports\Main%20Thesis\Thesis.docx#_Toc143868599)

# List of Abbreviations

BERT : Bi-directional Encoder Representation Transformer

BiLSTM : Bidirectional LSTM

BPE : Byte Pair Encoding

CNN : Convolutional Neural Network

DT : Decision Tree

GPT : Generative Pre-Trained Transformer

GRU : Gated Recurrent Unit

LSA/LSI : Latent Semantic Analysis / Latent Semantic Indexing

LSTM : Long Short Term Memory

MCNN : Multi-channel CNN

ML : Machine Learning

MLP : Multi-layered Perceptron

MNB : Multi-class Naïve Bayes

RBF : Radial Basis Function

RF : Random Forest

RNN : Recurrent Neural Network

SVM : Support Vector Machine

TF-IDF : Term Frequency – Inverse Document Frequency

# ABSTRACT

Sentiment Analysis is the task

# Introduction

## **Introduction**

### **1.1.1 Sentiment Analysis**

Sentiment Analysis (SA) is the automated task of identifying and extracting the polarity or emotion and subjective opinions in natural language texts. These expressions may be categorized as being positive, negative and neutral or more fine-grained as joy, sadness, anger, etc. So, the objective of SA is to classify a given text to a proper sentiment class. SA plays a vital role in understanding public opinions and sentiments expressed on social media platforms, current social media trends, customer feedback on e-commerce sites, etc. [1].

### **1.1.2 Nepali Language and its trend on Social Media**

Table 1 Nepali Numerals, Consonants, and Vowels [2]



Nepali, which is based on Devanagari script, is an official language of Nepal. Other than in Nepal, it is also spoken in India, Myanmar, and Bhutan as well as by Nepalese people spread worldwide [2]. Nepali (Devanagari Script) consists of 10 numerals, 36 consonants, and 13 vowel letters, which can be seen in (Table 1). Along with these characters, Nepali also consists of different modifiers and half-forms.

The usage of social media has been increasing in Nepal with the ubiquity of internet access and smartphones [2]. People have their presence on one or two social media platforms. Along with this growing presence, the number of people who prefer or use Nepali for communication (in written form) is also growing. With the proliferation of Nepali texts on social media platforms, it is apparent that a proper analysis and sentiment classification of these posts/tweets/comments on social media platforms is necessary to understand the attitudes of the people using these platforms. However, very few works have been done in the field of SA of Nepali texts [2].

### **1.1.3 Transformer**

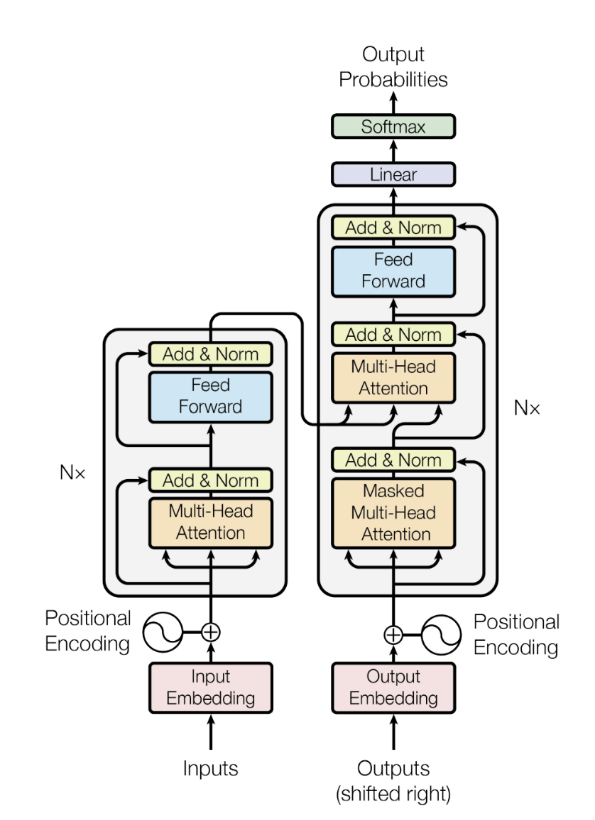


Figure 1. The transformer – model architecture [3]

In recent years, a state-of-the-art neural network architecture has revolutionized the field of Natural Language Processing (NLP), known as Transformer [3]. Transformers are neural network architectures that rely on attention mechanisms to encode and decode sequential data, as seen in Figure 1. The attention mechanism was first proposed by [4] to be used in sequence-to-sequence models for machine translation. Transformer can capture long-range dependencies and learn contextual representations of words and sentences, which had been previously a bottleneck as RNNs couldn’t carry along long-range dependencies [3].

Transformer, since its introduction, has become the dominant architecture powering many state-of-the-art models. Different types of transformer models have been proposed, that follow the architecture of the original transformer (by [3]). Based on their architectures, transformers can be divided into three categories as follows [5] :

1. Auto-Encoding or Encoder only transformer
2. Auto-regressive or Decoder only transformer
3. Sequence-to-sequence or Encoder-Decoder transformer

Auto-Encoding transformers are also known as BERT-based transformers. Similarly, Auto-Regressive transformers are also known as GPT-based transformers. BERT [6] and GPT (GPT-1 [7], GPT-2 [8], GPT-3 [9]) are the first of their kind, which only used either the encoder or decoder part of the original transformer. In the case of the Encoder-Decoder transformer, the original transformer was an encoder-decoder model.

There has been a paradigm shift in the field of NLP by the use of transformer-based models, which are now central components to many NLP systems and research [10]. In particular, transformer-based models, like BERT and its variants, have shown superior performance in various natural language processing tasks, including SA. Whereas, decoder-only models (GPT) have shown a capability to generate text as humans do. On top of that, they have shown the ability to learn human-like language perception, given large corpus, and learn specific tasks, like SA, with few task specific downstream training [8] [9].

## **1.2 Problem Statement**

Transformers have achieved remarkable results in various NLP tasks, including SA. However, most existing SA techniques and tools, which are built on top of this transformer architecture, are primarily developed for widely spoken languages and lack support for languages with limited resources, such as Nepali [10]. The usage of Nepali on the web has continuously risen. Just to put it in perspective, currently, one could scrape more than 20,000 unique tweets that are tweeted on any one specific date, written in Nepali. It is tested by scraping tweets from 2023-06-07 to 2023-06-09, which was above 20,000 for all three dates. Despite these developments, the task of SA is still understudied in Nepali language, as only a handful of works are available [2].

The works which have been done in Nepali SA are mostly done using the RNNs, CNNs, and traditional machine learning algorithms [11] [12] [13] [14] [15]. Some significant work has been done for building Nepali-only pre-trained Language Models, mostly BERT, such as NepaliBERT by [16], NepBERTa by [17] and distilBERT and DeBERTa by [18]. Some works have been done for text classification of Nepali news texts using transformers, such as in [18] and [19]. There hasn’t been much study regarding the use of transformer-based models for SA of Nepali social-media texts. Thus, the goal of this dissertation is to test and analyze the effectiveness of transformer models for SA of Nepali texts on social media.

## **1.3 Objectives**

This thesis aims to address the problem of SA of social media texts in Nepali using state-of-the-art deep learning architecture, known as transformer.

The main objectives of this thesis are:

1. To compare and evaluate different auto-encoder models (BERT, distilBERT [20]) and auto-regressive models (GPT2, distilGPT2) on the available datasets (NepCov19Tweets) for Sentiment Analysis in the Nepali language (Devanagari Script).
2. To study how well the transformer-based models perform, in the task of Sentiment Analysis of Nepali texts, compared to other Neural Network Architectures (CNN, LSTM, GRU).
3. To investigate methods to improve the performance of transformer-based models on Nepali Sentiment Analysis using techniques such as data augmentation and domain adaptation.

## **1.4 Scope and Limitation**

The research work is directed at comparative analysis of transformer models, which are Nepali pre-trained, for the task of sentiment analysis of Nepali social media texts. Here, sentiment analysis is limited to classification of texts to three categories: namely positive, negative and neutral. The use of sentiment classes is the direct consequence of the dataset used for the study i.e. NepCov19Tweets [13].

## **1.5 Organization of the report**

The research work is directed at comparative analysis of transformer m

# Background Study

## **2.1 Transformer**

Transformers are deep learning architectures that completely rely on self-attention mechanisms [4] to encode and decode sequential data. Transformer is proposed by Vaswani et. al. in [3]. Transformer is proposed as a novel architecture for sequence-to-sequence tasks, such as machine translation, without relying on RNNs or CNNs. The transformer uses a self-attention mechanism that allows the model to attend to different parts of the input sequence simultaneously, capturing long-range dependencies more effectively.

Transformer is an encoder-decoder-based architecture. Both encoder and decoder are implemented as a stack of self-attention and point-wise, fully connected layers, as seen in Figure 1. Residual connection is added around each of two sub-layers as {LayerNormalization(x + Sublayer(x))}. This is represented as Add & Norm block inside both the encoder and decoder in Figure 1. The encoder processes the input sequence, while the decoder generates the output sequence. Normalization is done for each input to the transformer.

There are two components that make the transformer unique from other sequence-to-sequence architectures i.e. Self-Attention and Positional Encodings. In the following subsections, these two components are discussed in brief.

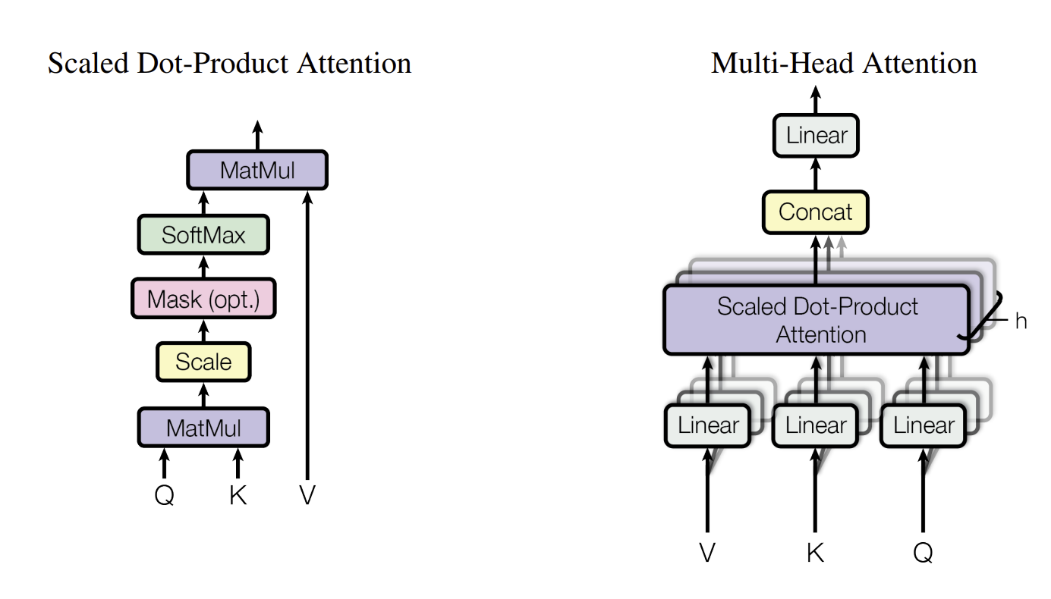


Figure 2 (Left) Scaled dot-product attention. (right) Multi-head attention [2]

### **Self-Attention**

The core idea of the Transformer is the self-attention mechanism, which computes attention weights between different positions in the input sequence to capture the relationships between words. The attention mechanism enables the model to focus on relevant parts of the sequence during encoding and decoding. The transformer employs multiple attention heads to enhance the expressive power of the self-attention mechanism. Each head learns a different representation of the input sequence, allowing the model to capture different types of dependencies and relationships. Moreover, due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.



Figure 3 (left) Global Self-Attention Layer, (middle) Causal Attention Layer, and (right) Cross Attention Layer [21]

Figure 2 shows how attention and multi-head attentions are structured, and the underlying working is shown in eqn(1), eqn(2), and eqn(3).

eqn(1)

here,

Where, X is the input (embedding) sequence and WQ, WK, and WV are learnable weights for, Q(query), K(key), and V(value), respectively. gives dot-product attention, and it is scaled by , where is the dimension of K (key).

eqn(2)

eqn(3)

where, the projections are parameter matrices , , ,and . *h* is the number of attention heads. And in the case of multi-head attention, . is the dimension of output returned by the model.

Though the authors in [3] call all the attention layers self-attention, however, there are differences in how these attentions work depending on where it is used. As seen in Figure 1, there are three different instances of attention layer, which are named (1) global self-attention layer, (2) causal self-attention layer, and (3) cross attention layer, from the encoder - to - decoder respectively [21]. These differences can be seen in Figure 3.

### **Embedding and Positional Encoding**

The transformer model uses embedding (trainable weight) vectors to convert the input tokens and output tokens to vectors of dimension *dmodel*. Initially, those embedding vectors are context-independent and position-independent. Before the final representations come out from the transformer, the embedding is processed by the attention block, and the output of the attention block is passed through a feed-forward layer. And since the transformer doesn’t make use of RNNs and CNNs, the sequence order of the input is not captured. The use of only weight vectors from the embedding layer is similar to the use of a bag of words, thus it loses an important piece of information. To overcome this, positional encoding is injected into the input embedding vectors. the weights of the embedding layer are multiplied by and then positional encoding is added to those weights. Transformer makes use of sine and cosine functions to compute positional encoding:

eqn(4)

eqn(5)

where, . The sine and cosine encodings are arranged in interleaving order. The positional encoding vector is then added to input embedding vectors before passing it to/from the transformer encoder-decoder block.

## **Decoder Only Transformers - GPT**

Decoder-only transformers are a type of transformer-based architecture that uses only the decoder part of the original transformer model. They are mainly used for natural language generation tasks, where the goal is to predict the next token given a sequence of previous tokens, such as text completion, summarization, and dialogue generation. These are also called auto-regressive transformer models.

The first decoder-only transformer, GPT or GPT-1, was introduced in [7]. The improved and better-performing revisions of GPT were later presented in as GPT-2 [8] and [9]as GPT-3, which are massive in size and scale. The GPT model has showcased remarkable abilities in various language-related tasks, including text completion, translation, question-answering, classification, and more. GPT-2 can perform these tasks without any task-specific fine-tuning. In addition to that, GPT-3 can adapt to new tasks by simply providing a few examples or instructions in natural language as part of the input.

The overall structure of GPT is similar to the decoder part of the original transformer [3], with some major changes listed as follows

1. GPT uses learned positional embedding rather than the fixed sinusoidal positional encoding used in the original transformer.
2. GPT uses GELU (Gaussian Error Linear Unit) as an activation function, whereas the original transformer used RELU (Rectified Linear Unit).
3. GPT doesn’t include the cross-attention block from decoder block of original transformer.

GPT-2 and GPT-3 fairly follow the same architecture of GPT but with bigger model size, increased scale, and billions of parameters. There are other decoder-only or auto-regressive transformer models as well, but here we limit ourselves to GPT only. We also use DisitlGPT2, a distilled version of GPT2, which we discuss in sub-section below.

### **2.2.1 DistilGPT2**

DistilGPT2 is a distilled version of GPT2. By the use of knowledge distillation mechanisms, distilGPT2 gives similar performance like GPT2 with much less parameter than GPT2. DistilGPT2 is developed by Huggingface. DistilGPT2 is built by the authors of [20]. In [20], they introduced distilBERT and same approach is applied for the creation of distilGPT2. The model is available in Huggingface-hub.

## **Encoder Only Transformers - BERT**

Encoder-only transformer is a transformer model that uses only the encoder part of the original transformer. The encoder processes the input sequence and generates a representation for each token. Thus, the encoder-only transformers are capable of learning the language structures and contexts within the sentences thoroughly.

The first encoder-only transformer was introduced in [6] as BERT. BERTs' internal architecture is similar to the encoder part of the original transformer. The pre-training of the BERT model is done using MLM (masked language modeling) where some of the words from input are masked randomly and the objective is to predict the original vocabulary id of the masked word based only on its context. Thus, it can learn the left and right context of a token, which makes it bidirectional. This allows BERT to learn rich and deep representations of natural language that can be fine-tuned for various downstream tasks, such as question answering, SA, named entity recognition, and more. BERT can also handle different types of inputs, such as single sentences, sentence pairs, or longer documents.

The overall structure of BERT is similar to the decoder part of the original transformer [3], with some major changes listed as follows

1. BERT uses learned positional embedding rather than the fixed sinusoidal positional encoding used in the original transformer. In addition, BERT also uses segment embeddings, which helps the model to identify one sentence from another in the input sequence.
2. BERT is trained with Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) objectives. Whereas, GPT is trained with Causal Language Modelling (CLM) objective.

Many other encoder-only BERT-based transformer models are now available but here we limit ourselves to BERT only. However, these other models, mostly, are architecturally identical to BERT. To list some of those models, we have multilingual BERT, DistilBERT [20], RoBERTa [22], DeBERTa [23], XLM-RoBERTa [24], etc. NepaliBERT by [16], NepBERTa by [17] are two Nepali-only pre-trained BERT models. In the following sub-sections, we briefly discuss other BERT-based models which are used in this research.

### **2.3.1 Multilingual BERT**

Multilingual BERT (M-BERT) is a BERT model which is pre-trained on more than one language. M-BERT was built by authors of [6] along with the BERT model. M-BERT is pre-trained on 104 languages with MLM objective.

### **2.3.2 DistilBERT**

The DistilBERT is a distilled version of BERT which was introduced in [20]. The DistilBERT is smaller in size (40% smaller than BERT), however, as claimed by authors, it is 60% faster and retains 97% of BERT’s performance. The reduction in size is done by the use of knowledge distillation process. The DistilBERT differs from BERT in two essence (1) The Pooler layer and token-type embeddings are removed and (2) number of layers is halved. Nepali only DistilBERT language model has been trained by [18].

### **2.3.3 DeBERTa**

DeBERTa stands for Decoding-enhanced BERT with disentangled attention, which was proposed by [23]. In DeBERTa, token embedding and position embedding is kept separate, thus for each token there are two vector representations. Attention score is calculated using disentangled matrices based on their contents and relative positions. DeBERTa also makes use of causal mask, such that each token can only attend itself against token on its left. Nepali only DeBERTa language model has been trained by [18].

### **2.3.4 XLM-RoBERTa**

XLM-RoBERTa is a multilingual language model. It was proposed in [24]. This model is pre-trained on 100 languages. It uses RoBERTa way of training a language model. XLM-RoBERTa is trained with the focus on scaling the models capacity across languages. This is done by controlling several parameters like training set size, the size of the shared subword vocabulary, and the rate at which training example is sampled from each language. The authors in [24] show that XLM-RoBERTa perform better than multilingual BERT.

# Literature Review

SA is a text classification problem where the target classes are the sentiments or emotions being conveyed in the given text. These sentiments are categorized as being Positive, Negative, or Neutral. SA is a well-studied problem in Natural Language Processing (NLP) and has been applied to various languages and domains. However, most existing works focus on high-resource languages, such as English, and there is a lack of research and resources for low-resource languages, such as Nepali.

In this section, the works that have been done in the field of SA on Nepali texts are reviewed. Since there are not many works done for Nepali SA, the works done in the Nepali news classification are also reviewed. Many news portals publish news written in Nepali (Devanagari). Moreover, these news portals already categorize the news articles they publish. This provides abundant data. In addition to the data abundance, the reason for our interest in news classification, some works have been conducted for news classification using transformers, some of which are explored at the end of this section.

The authors in [16] claim to be the first to perform SA on Nepali texts. They proceed on the task with two approaches, first, a resource-based approach, and second, an ML-based approach. They used the Naïve Bayes algorithm for the ML-based approach. In the case of the resource-based approach, they use SentiWordNet, a dictionary translated English-to-Nepali SentiWordNet. While the resource-based approach performed poorly they report 77.8% precision and 70.2% recall on the dataset of 20000 sentences. One thing to note here is, they classified two classes, subjective (positive or negative) and objective (neutral).

In [17], SA was performed on data collection of YouTube comments. They study abusive SA and SA over different aspect terms within the sentence. They study four different aspects like profanity, violence, etc. and only use two classes, positive and negative, for sentiment. In [17] for the SA task, they used BiLSTM, CNN, SVM, and BERT models. They report an 81.6% F1-score by BiLSTM and 81.1% F1-score by CNN. The BERT model performs slightly less with a 79.9% F1 score. The maximum accuracy reported for the BERT model is 80% which is also lower than that of BiLSTM and CNN. This should be noted that they used multi-lingual BERT, which is trained in 104 languages including Nepali, rather than using monolingual, only Nepali, pre-trained BERT [5] [11] [10], moreover the dataset is also very small.

In [18], SA on Nepali tweets regarding covid-19 using the CNN model can be found. They provide the most extensive dataset in the domain of SA of Nepali texts. They collected the data and classified the tweets into three polarities; positive, negative, and neutral. The dataset is called NepCOV19Tweets. They propose three different approaches to feature extraction for the representation of the data, namely fastText-based, domain-specific, and domain-agnostic. A separate CNN model is trained using each of the feature representations. Then, a fusion layer is used to make combined decisions based on the result from the three models. The study makes a comparison of the performance of the CNN model with other machine learning models like SVM, DT, RF, etc. When the performance of individual CNN was evaluated, CNN trained using fastText-based feature representation performed best with 68.1% accuracy and 58.5% F1 score. Combined, the model achieved 68.7 accuracy and 56.4 F1 score.

The follow-up work on SA on the NepCOV19Tweets dataset can be observed in [25] and [20]. Also, the authors focus on the betterment of text representation and propose hybrid feature representations; TF-IDF weighted fastText-based method in [25], and hybrid fastText-based and domain-specific methods in [20]. The effectiveness of respective feature representation was tested using 10 different traditional machine learning algorithms in [18], and a multi-channel CNN approach is proposed in [20].

The highest achieved performance, as reported in [25], is with the SVM+RBF model with 75.6 F1-score and 70.69% accuracy. It can be observed that F1-score and accuracy both increased with the use of hybrid text feature representation.

To implement multi-channel CNN, four different CNNs with different kernel sizes (1, 2, 3, and 4 respectively) were initialized. First, each CNN is fine-tuned. Then, each CNN is aggregated using a fusion layer, thus establishing a multi-channel. Then, the model is trained in an end-to-end fashion for the classification. They report the performance of MCNN with a 61.6 F1-score and 71.3 accuracy.

In [23], a news classification task is performed using SVM, and three different feature extraction methods are used in the experiments. They used TF-IDF, word2vec, and LSI-based approaches for feature extraction. LSI-based approach achieved the highest accuracy, 93.7%, followed by word2vec and TF-IDF with 86.3% and 85.4% accuracy, respectively. However, looking at the overall performance, and evaluating all the performance metrics (accuracy, precision, recall, and f1-score), the model performed better with word2vec-based feature representation than LSI and TF-IDF.

Table 2. Tokenization of Nepali texts by the BERT tokenizer.

|  |  |  |
| --- | --- | --- |
|  | Before tokenization | After tokenization |
| Nepali (Devanagari): | फ्लु (फ + ◌् + ल + ◌ु) | फल (फ+ ल) |
| Translation: | Flu | Fruit |

Similarly, in [24], they use SVM along with Naïve Bayes and Multi-layered Perceptron for the task of news classification. A TF-IDF-based feature extraction method is used for feature vector representation. SVM with RBF kernel is shown to achieve the best performance with above 74% baseline across each performance metric (accuracy, precision, recall, and f1-score), specifically 74.65 % accuracy.

In [25], the authors make use of RNN-based models, like LSTM, GRU, and adaptive GRU for Nepali news classification. As other works do, which are referenced so far, [25] also compares the performance of RNN-based models with other traditional ML approaches on the classification task. It makes use of TF-IDF and word2vec-based based feature extraction methods. The GRU model achieved the highest among RNN models with 77.44% accuracy, whereas, a simple perceptron achieved 78.56% accuracy. The author attributes the comparatively lower performance of RNN models (LSTM and GRU) is due to the limited amount of data available for training.

[11] and [10] focuses on training the monolingual BERT language model on a fairly large Nepali corpus. [21] and [22] builds upon the intuitions from [11] and [10] for Nepali news text classification. [11] uses the BERT tokenizer as it is, without taking into account the language difference between English and Nepali leading to incoherent tokenized words, which can be seen in Table 1. Authors of [21] pre-trained DistilBERT [12] and DeBERTa [14], two BERT-based models, and fine-tunes for the task of text classification. Pre-training is done through Masked Language Modelling (MLM). They then compare the performance of their pre-trained LMs with the pre-trained LMs from [11] and [10]. DeBERTa achieves 88.93% accuracy and DistilBERT achieves 88.31% accuracy on the downstream task. This is a significant performance improvement on the text classification task. In [23], SVM with LSI features achieves 93.7% accuracy, however, the f1-score is significantly lower than accuracy. From this, it can be reckoned that the model fails to identify some of the categories. But since [21] doesn’t provide those metrics, it is not possible to make a comparison of the overall performance of the two methods on the news classification task.

Authors of [22] goes a step further on the use of transformer models for news classification. While [21] used DistilBERT, DeBERTa and two separate pre-trained BERT models, [22] uses BERT, RoBERTa [13], DistilBERT, DeBERTa, mBERT, XLM-RoBERTa [15], and HindiRoBERTa. Along with those models, [22] also uses Bi-LSTM, MNB, RF, and SVM in their study. In their study, [22] shows that the transformer models performed better as the size of the dataset was increased. The DistilBERT and DeBERTa achieved the highest accuracy, 87.03% and 86.63% respectively. This result corroborates the finding of [21].

# Methodology

# Implementation

# Result and Analysis

# References

|  |  |
| --- | --- |
| [1] | B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015. |
| [2] | T. B. Shahi and C. Sitaula, "Natural language processing for Nepali text: a review," *Artificial Intelligence Review, Springer,* vol. 55, pp. 3401-3429, 2021. |
| [3] | A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, N. A. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017. |
| [4] | D. Bahdanau, K. Cho and Y. Bengio, "NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE," in *ICLR* , 2015. |
| [5] | Huggingface, "How do Transformers work?," Huggingface, [Online]. Available: https://huggingface.co/learn/nlp-course/chapter1/4. [Accessed 27 8 2023]. |
| [6] | J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Google AI Language, 2019. |
| [7] | A. Radford and K. Narasimhan, "Improving Language Understanding by Generative Pre-Training," 2018. |
| [8] | A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language Models are Unsupervised Multitask Learners". |
| [9] | T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh and Dani, "Language Models are Few-Shot Learners," Open AI, 2020. |
| [10] | R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. v. Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji and others.., "On the Opportunities and Risks of Foundation Models," 2022. |
| [11] | C. P. Gupta and B. K. Bal, "Detecting Sentiment in Nepali Texts: A Bootstrap Approach for Sentiment Analysis of texts in the Nepali Language". |
| [12] | O. M. Singh, S. Timalsina, B. K. Bal and A. Joshi, "Aspect Based Abusive Sentiment Detection in Nepali Social Media Texts," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2020. |
| [13] | C. Sitaula, A. Basnet, A. Mainali and T. B. Shahi, "Deep Learning-Based Methods for Sentiment Analysis on Nepali COVID-19-Related Tweets," *Computational Intelligence and Neuroscience,* 2021. |
| [14] | T. B. Shahi, C. Sitaula and N. Paudel, "A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification". |
| [15] | C. Sitaula and T. B. Shahi, "Multi-channel CNN to classify Nepali Covid-19 related tweets," 2022. |
| [16] | S. Pudasaini, A. Tamang, S. Lamichhane, S. Adhikari, S. Adhikari, S. Thapa and J. Karki, "Pre-training of Masked Language Model in Nepali," in *36th Conference on Neural Information Processing Systems*, 2022. |
| [17] | M. Gautam, S. Timalsina and B. Bhattarai, "NepBERTa: Nepali Language Model Trained in a Large Corpus," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the*, 2022. |
| [18] | U. Maskey, M. Bhatta, S. R. Bhatta, S. Dhungel and B. K. Bal, "Nepali Encoder Transformers: An Analysis of Auto Encoding Transformer Language Models for Nepali Text Classification," in *Proceedings of SIGUL2022 @LREC2022*, 2022. |
| [19] | S. Suwanchai, S. R. Tamrakar and Chaklam, "Comparative Evaluation of Transformer-Based Nepali Language Models," [Online]. Available: https://assets.researchsquare.com/files/rs-2289743/v1/aa3f3ba4a38a880db3d6c5dc.pdf?c=1670229384. [Accessed 19 06 2023]. |
| [20] | V. Sanh, L. Debut, J. Chaumond and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. |
| [21] | "Neural Machine Translation with a Transformer and Keras," Tensorflow, 03 06 2023. [Online]. Available: https://www.tensorflow.org/text/tutorials/transformer. [Accessed 24 06 2023]. |
| [22] | Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 26 7 2019. [Online]. Available: https://arxiv.org/pdf/1907.11692.pdf. [Accessed 23 06 2023]. |
| [23] | P. He, X. Liu, J. Gao and W. Chen, "DeBERTa: Decoding-enhanced BERT with Disentangled Attention," in *ICLR* , 2021. |
| [24] | A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale," arXiv, 2019. |
| [25] | T. B. Shahi and A. K. Pant, "Nepali News Classification using Naive Bayes, Support Vector Machines and Neural Networks," in *International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, 2018. |
| [26] | O. M. Singh, "Nepali Multi-Class Text Classification," 2018. [Online]. Available: https://oya163.github.io/assets/resume/Nepali\_Text\_Classification.pdf. [Accessed 19 6 2023]. |
| [27] | K. Kaﬂe, D. Sharma, A. Subedi and A. K. Timalsina, "Improving Nepali Document Classification by Neural Network," in *Proceedings of IOE Graduate Conference*, 2016. |