SA of Social Media Texts in Nepali Using Transformers

A Dissertation Proposal Report

Submitted by:

Regan Maharjan

Roll No. 17/075

Submitted to:

Central Department of Computer Science and Information Technology

Tribhuvan University

Kirtipur, Nepal

Under the Supervision of

Bikash Balami

Co-Supervison

Tej Bahadur Shahi

Table of Contents

Table of tables

List of Figures

List of Abbreviations

MNB	:	Multi-class Naïve Bayes
MLP	:	Multi-layered Perceptron
SVM	:	Support Vector Machine
RBF	:	Radial Basis Function
BERT	:	Bi-directional Encoder Representation Transformer
GPT	:	Generative Pre-Trained Transformer
CNN	:	Convolutional Neural Network
MCNN	:	Multi-channel CNN
RNN	:	Recurrent Neural Network
TF-IDF	:	Term Frequency – Inverse Document Frequency
LSTM	:	Long Short Term Memory
GRU	:	Gated Recurrent Unit
BiLSTM	:	Bidirectional LSTM
DT	:	Decision Tree
RF	:	Random Forest
ML	:	Machine Learning
BPE	:	Byte Pair Encoding
LSA	:	Latent Semantic Analysis

1. Introduction

Sentiment Analysis (SA) is the task of identifying and extracting the polarity or emotion and subjective opinions in natural language texts. It plays a vital role in understanding public opinions and sentiments expressed on social media platforms, e-commerce sites, or any other domain. With the growing reach of the technology and availability of computing machines (such as PC, laptops, and smartphones) as well as the internet even in remote areas, social media usage has become ubiquitous in Nepal [1]. From the young to old, almost everyone has a presence at least in one of the many social media platforms available. The usage of social media and way of expression on these platforms may have evolved from texts to the image to audio-video, however still, the basic form of expressing and interacting remains in large the use of written form.

The world wide web and the Internet have connected people from distant parts of the world. Moreover, it has become a tool that allows us to connect even in the most isolated period, just like the recent pandemic. It also can be taken into account that people are more comfortable sharing their emotions on social media rather than with a person. So, as the number of people who prefer Nepali on digital platforms increases, it is apparent that a proper analysis and sentiment classification of these posts/tweets/comments on digital/social media platforms is necessary.

The young generation, in the majority, prefers the usage of English as the mode of expression. Even so, the actual number of people that use the English language is very minimal, where most prefer to use only English alphabets to converse and express themselves in Nepali. Nonetheless, there are groups of people that express themselves on the internet using Nepali (Devanagari Script). The number of these peoples are growing as there is the availability of typing in Nepali using, may it be Nepali character-labeled keyboards or English-to-Nepali Unicode converter. The growing number can also be attributed to the usage of Nepali by public figures (politicians, actors, and others) and as well as business organizations, as most of the target audience for these business organizations do not understand English, to increase reachability. Also, the online news portals must be attributed as they have long since published news in Nepali script.

In recent years, a state-of-the-art neural network architecture has revolutionized the field of Natural Language Processing (NLP), known as Transformer[2]. Transformers are neural network architectures that rely on attention mechanisms [3] to encode and decode sequential data. They 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 [2]. They were introduced in [2] and have since become the dominant architecture powering many state-of-the-art models. As [4] points out, there has been a paradigm shift in the field of NLP by the use of transformer-based models (like BERT [5], and GPT [6] [7] [8]). In particular, transformer-based models like BERT [5] and its variants have shown superior performance in various natural language processing tasks, including SA. In this thesis, we aim to address problem of SA of social media texts in Nepali using state-of-the-art deep learning techniques, specifically transformer.

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. 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. We tested it by scraping tweets from 2023-06-07 to 2023-06-09, which was above 20,000 for all three dates.

The works which have been done in Nepali SA are mostly done using the RNNs, CNNs, and traditional machine learning algorithms [9] [10] [11] [12] [13]. Some significant work has been done for building Nepali-only pre-trained Language Models, mostly BERT, such as NepBERT [14] and NepBERTa [15]. Some works have been done for text classification of Nepali news texts using transformers, such as in [16] and [17]. 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.

3. Objective

The main objectives of this thesis are:

1. To compare and evaluate different transformer-based models on the available datasets for SA in the Nepali language (Devanagari Script).
2. To study how well the transformer-based models perform compared to other Neural Network Architectures on the premise of Nepali being a low-resource language.
3. To investigate methods to improve the performance of transformer-based models on Nepali SA using techniques such as data augmentation, domain adaptation, etc.

4. 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 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.

Here we discuss some of the works that have been done in the field of SA on Nepali texts. We also take into account the work done in text classification. We do this because many works have been done in the domain of text classification of Nepali texts (not SA but news classification). 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 text classification (news classification) using transformers, some of which we will explore at the end of this section.

In [5], the authors 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 [10], 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 [10] for the SA task, they used BiLSTM, CNN, SVM, and BERT models. They report an 81% F1-score by BiLSTM and CNN. The BERT model performs slightly less with a 79.9% F1 score. The maximum accuracy reported for the BERT model is 81.5%. 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] [14] [15].

In [11], we find SA on Nepali tweets regarding covid-19 using the CNN model. 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. In [11] they make 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 representation performed best with 68.1% accuracy and 58.5% F1 score. Combined, the model achieved 68.7 accuracy and 56.4 F1 score.

We can see the follow-up work on SA on the NepCOV19Tweets dataset in [12] and [13]. We find that the authors focus on the betterment of text representation and propose hybrid feature representations; TF-IDF weighted fastText-based method in [12], and hybrid fastText-based and domain-specific methods in [13]. The effectiveness of respective feature representation was tested using 10 different traditional machine learning algorithms in [11] , and a multi-channel CNN approach is proposed in [13].

The highest achieved performance, as reported in [12], is with the SVM+RBF model with 75.6 F1-score and 70.69% accuracy. We can see 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.

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

Table 1. Tokenization of Nepali texts by the BERT tokenizer.

In [18], we see a news classification task done 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 wword2vec-based feature representation than LSI and TF-IDF.

Similarly, in [19], 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 [20] we see the use of RNN-based models, like LSTM, GRU, and adaptive GRU for Nepali news classification. As other works do, which are referenced so far, [20] 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.

[14] and [15] focuses on training the monolingual BERT language model on a fairly large Nepali corpus. [16] and [17] builds upon the intuitions from [14] and [15] for Nepali news text classification. [14] 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 [16] pre-trained DistilBERT [21] and DeBERTa [22], 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 [14] and [15]. 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. We saw in [18] that SVM with LSI features achieves 93.7% accuracy, however, the f1-score is significantly lower than accuracy. From this, we know that the model fails to identify some of the categories. But since [16] doesn't provide those metrics, we can't make a comparison of the overall performance of the two methods on the news classification task.

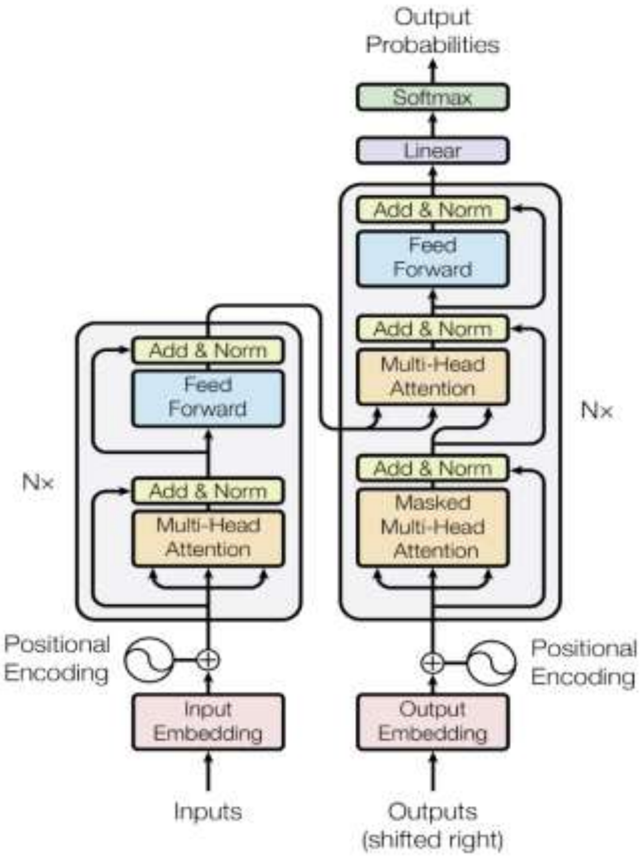

Figure 1. The transformer – model architecture [2]

[17] goes a step further on the use of transformer models for news classification. While [16] used DistilBERT, DeBERTa and two separate pre-trained BERT models, [17] uses BERT, RoBERTa [23], DistilBERT, DeBERTa, mBERT, XLM-RoBERTa [24], and HindiRoBERTa. Along with those models, [17] also uses Bi-LSTM, MNB, RF, and SVM in their study. In their study, [17] 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 [16].

5. Background

5.1. Transformer

Transformers are deep learning architectures that completely rely on self-attention mechanisms [3] to encode and decode sequential data. Transformer is proposed in [2]. 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 is 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. The encoder processes the input sequence, while the decoder generates the output sequence. Normalization is done for each input to the transformer. Here we discuss two components that make transformer unique from other sequence-to-sequence architectures i.e. Self-Attention and Positional Encodings.

5.1.1. 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.

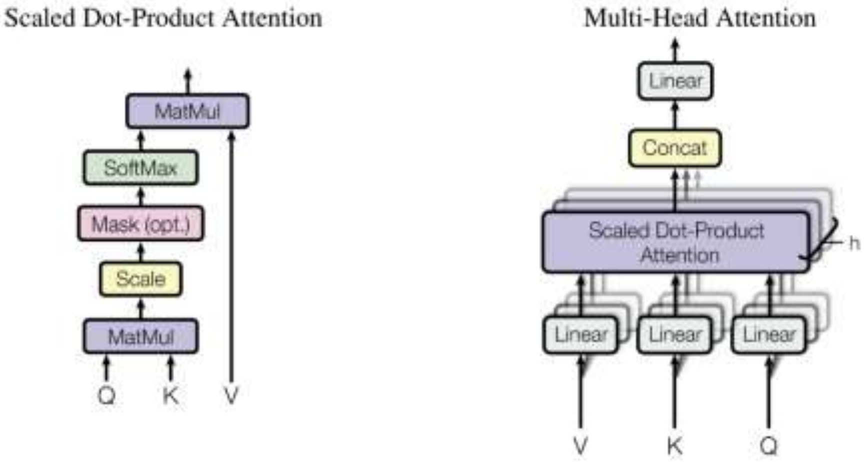

Figure 4 (Left) Scaled dot-product attention. (right) Multi-head attention [2]

Figure 5. (left) Cross Attention Layer, (middle) Global Self-Attention Layer, and (right) Cross Attention LayerFigure 6 (Left) Scaled dot-product attention. (right) Multi-head attention [2]

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

$$Attention(Q, K, V\ ) \ = \ softmax(\frac{QK^T}{\sqrt{d_k}})V \qquad \text{eqn(1)}$$

here,

$Q \ = \ X \ * \ W_k$

$K \ = \ X \ * \ W_k$

$V \ = \ X \ * \ W_k$

Where, X is the input (embedding) sequence and W$_q$, W$_{k,}$ and W$_v$ are learnable weights for, Q(query), K(key), and V(value), respectively. QK^T gives dot-product attention, and it is scaled by $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of K (key).

$$MultiHead(Q,K,V\ ) \ = \ Concat(head_1,\dots,head_h)W_O \qquad \text{eqn(2)}$$

$$head_i \ = \ Attention(QW_i^Q\ ,KW_i^K\ ,\ VW_i^V) \qquad \text{eqn(3)}$$

where, the projections are parameter matrices $W_i^Q \in R^{d_{model}\times d_k}$, $W_i^K \in R^{d_{model}\times d_k}$, $W_i^V \in R^{d_{model}\times d_v}$,and $W^O \in R^{hd_v\times d_{model}}$. h is the number of attention heads. And in case of multi-head attention, $d_k = d_v = d_{model}/h$. d_{model} is dimension of output returned by the model.

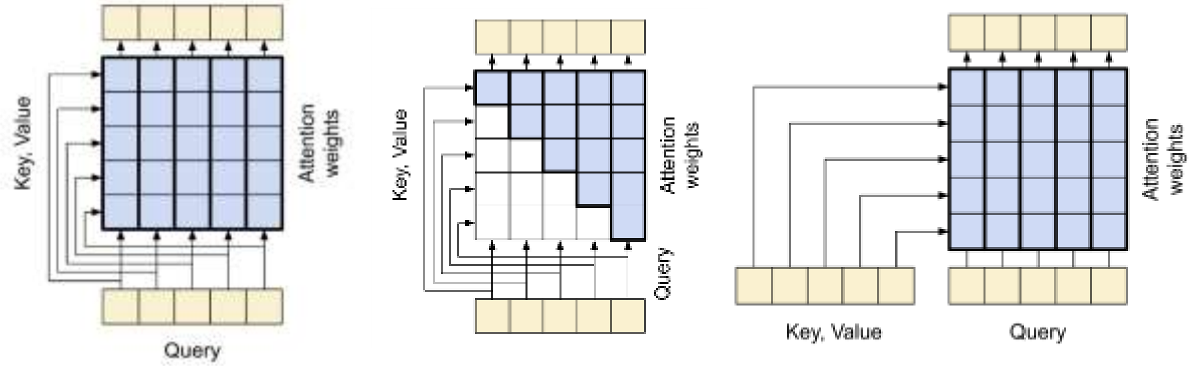

Figure 7. (left) Global Self-Attention Layer, (middle) Causal Attention Layer, and (right) Cross Attention Layer [25]

Though the authors in [2] 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 as (1) global self-attention layer, (2) causal self-attention layer, and (3) cross attention layer, from encoder - to - decoder respectively [25]. These differences can be seen in Figure 3.

5.1.2. Embedding and Positional Encoding

Transformer use learned embedding to convert the input tokens and output tokens to vectors of dimension *d$_{model}$*. The weights of embedding layer is multiplied by $\sqrt{d_{model}}$. These embedding vectors are context-independent. Before we get the final representations out from the transformer, the output of 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. Use of only weight vectors from embedding layer is similar to use of bag of words, thus we lose an important piece of information. To overcome this, the positional encoding is injected to the input embedding vectors. Transformer make use of sine and cosine functions to compute positional encoding:

$$PE_{(pos,2i)} = sin\left(pos/100000^{\frac{2i}{d_{model}}}\right) \qquad \text{eqn(4)}$$

$$PE_{(pos,2i+1)} = cos\left(pos/10000^{\frac{2i}{d_{model}}}\right) \qquad \text{eqn(5)}$$

where, $0 \leq i \leq d_{model}/2$. The sine and cosine encodings are arranged interleaving. The positional encoding vector is then added to input embedding vectors before passing it to / from the transformer encoder-decoder block.

5.2. Decoder Only Transformers - GPT

Decoder-only transformers are a type of transformer-based architecture that use only the decoder part of the original transformer model [2]. 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 [6]. The improved and better performing revision of GPT were later presented in [7] as GPT-2 and [8] as GPT-3, which are massive in size and scale. The GPT model have showcased remarkable abilities in various language-related tasks, including text completion, translation, question-answering, classification, and more. GPT-2 is able to perform these tasks without any task-specific fine-tuning. In addition to that, GPT-3 is able to adapt to new tasks 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 [2], with some major changes listed as follows

i. GPT uses learned positional encoding rather than fixed positional encoding used in original transformer.

ii. GPT uses GELU (Gaussian Error Linear Unit) as activation function in fully connected position-wise feed forward network, whereas original transformer used RELU (Rectified Linear Unit).

GPT-2 and GPT-3 fairly follows the same architecture of GPT but with bigger model size, and 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.

5.3. 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 [5] as BERT. BERTs internal architecture is similar to the encoder part of the original transformer. The pre-training of BERT model is done using MLM (masked language modelling) 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 sentence, sentence pairs, or longer documents.

Many other decoder-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, NepBERTa [15], NepaliBERT [14], DistilBERT [21], RoBERTa [23], DeBERTa [22], XLM-RoBERTa [24].

6. Research Methodology

The research methodology for this thesis will involve the following steps:

6.1. Data Collection:

Research and collect already available representative datasets of Nepali social media text available in open-source data repositories such as Kaggle, GitHub, etc.

a. In [11], authors have created a dataset of Nepali tweets regarding covid-19, called NepCov19, for SA which is publicly available on Kaggle.
b. In [10], authors have created a dataset of YouTube comments in Nepali for SA and aspect term extraction, which is publicly available on GitHub.
c. Data augmentation through transliteration and other approaches.
d. Data collection from one of the popular social media platforms from 2015-04-25 to 2015-05-02, which is a week period of the 2015 earthquake.

6.2. Data Preprocessing:

The collected data will undergo preprocessing steps which include text normalization, tokenization, and cleaning. Since the transformer models use tokenizing algorithms like wordpiece, sentencepiece, and byte-pair encoding (BPE), we assume that traditional preprocessing steps like stemming and lemmatization are not required as they are inherently handled by underlining tokenizer algorithm based on the available corpus and expected vocabulary size. Data cleaning will be done to remove any foreign language characters, including emojis, and will be kept where assumed it will help in the task.

6.3. Model Architecture:

Initially, we will deploy encoder-only transformer models, such as BERT or its variants, for SA. These models have shown significant improvements in capturing contextual information and achieving state-of-the-art results in various NLP tasks. After that, we will use decoder-only transformer models, such as GPT, and compare the efficiency of both models.

6.4. Feature Extraction and Hybrid Approaches:

The representation learned by transformer models is self-sufficient. The learnable weights embedding and positional encoding used by transformers are already robust. In addition to embedding, attention mechanisms used by transformers, self-attention, causal-attention, and cross-attention, are capable of capturing the long-range dependencies and learning contextual representations of words in sentences. Such that, the representations learned by transformer models like BERT are self-sufficient and don't need any other features extracted through traditional methods.

However, we will briefly investigate hybrid feature extraction methods by combining transformer-based representations with other feature extraction techniques like Bhavanakos and Sabdakos, a word2vec and doc2vec method respectively, as created by [9] and using domain-specific and domain-agnostic features as used by [11] [12] [13]. The goal is to boost the model's performance and efficiency by incorporating additional linguistic information specific to Nepali text.

6.5. Pre-training, Fine-tuning, and Training:

We will adapt and fine-tune pre-trained transformer models for SA of Nepali social media texts. We will use existing Nepali language models (e.g. LM made available by [14]) and/or train our own language models on Nepali corpora to initialize the transformer models. We will also explore various techniques such as data augmentation, domain adaptation, cross-lingual transfer learning, etc. to improve the performance of the transformer models.

For model initialization and training, libraries such as Huggingface, Tensorflow, Keras, and Trax will be utilized.

6.6. Evaluation and Performance Metrics:

Fine-tuned SA models will be evaluated on the validation set separated from the collected Nepali social media text dataset [10] [11]. The performance evaluation of SA models will be done using standard evaluation metrics like accuracy, precision, recall, F1-score, etc.

We will compare the performance of our proposed transformer-based model with the existing SA approaches for Nepali, including the referenced papers. This analysis will help assess the effectiveness and potential advantages of our approach.

To evaluate, calculate performance metrics, and visualize data, various libraries such as Scikit-Learn, Matplotlib, Seaborn, and Tensorflow will be utilized.

7. Time Schedule:

A Gantt chart here.

8. References

[1] T. B. Shahi and C. Sitaula, "Natural language processing for Nepali text: a review," *Artificial Intelligence Review, Springer,* vol. 55, pp. 3401-3429, 2021.

[2] 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.

[3] D. Bahdanau, K. Cho and Y. Bengio, "NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE," in *ICLR* , 2015.

[4] 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.

[5] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Google AI Language, 2019.

[6] A. Radford and K. Narasimhan, "Improving Language Understanding by Generative Pre-Training," 2018.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language Models are Unsupervised Multitask Learners," OpenAI.

[8] 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.

[9] C. P. Gupta and B. K. Bal, "Detecting Sentiment in Nepali Texts: A Bootstrap Approach for SA of texts in the Nepali Language".

[10] 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.

[11] C. Sitaula, A. Basnet, A. Mainali and T. B. Shahi, "Deep Learning-Based Methods for SA on Nepali COVID-19-Related Tweets," *Computational Intelligence and Neuroscience,* 2021.

[12] T. B. Shahi, C. Sitaula and N. Paudel, "A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification".

[13] C. Sitaula and T. B. Shahi, "Multi-channel CNN to classify Nepali Covid-19 related tweets," 2022.

[14] 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.

[15] 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.

[16] 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.

[17] S. R. T. a. C. Silpasuwanchai, "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].

[18] K. Kafle, D. Sharma, A. Subedi and A. K. Timalsina, "Improving Nepali Document Classification by Neural Network," in *Proceedings of IOE Graduate Conference,* 2016.

[19] 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.

[20] 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].

[21] 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.

[22] P. He, X. Liu, J. Gao and W. Chen, "DeBERTa: Decoding-enhanced BERT with Disentangled Attention," in *ICLR* , 2021.

[23] 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].

[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] "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].