Development of Pre-Trained Transformer-based Models for the Nepali Language

Prajwal Thapa*, **Jinu Nyachhyon***, **Mridul Sharma***, **Bal Krishna Bal**

Information and Language Processing Research Lab (ILPRL), Kathmandu University,
prazzwalthapa87@gmail.com, nyachhyonjinu@gmail.com, mridulsharma3301@gmail.com, bal@ku.edu.np

Abstract

Transformer-based pre-trained language models have dominated the field of Natural Language Processing (NLP) for quite some time now. However, the Nepali language, spoken by approximately 32 million people worldwide, remains significantly underrepresented in this domain. This underrepresentation is primarily attributed to the scarcity of monolingual data corpora and limited available resources for the Nepali language. While existing efforts have predominantly concentrated on basic encoder-based models, there is a notable gap in the exploration of decoder-based architectures. To address this gap, we have collected 27.5 GB of Nepali text data, approximately 2.4x larger than any previously available Nepali language corpus. Leveraging this data, we pre-trained three different models i.e., BERT, RoBERTa, and GPT-2, exclusively for the Nepali Language. Furthermore, we performed instruction tuning and explored its potential for monolingual Nepali data, providing a foundation for future research. Our models outperformed the existing best model by 2 points on Nep-gLUE benchmark, scoring 95.60 and also outperformed existing models on text generation tasks, demonstrating improvements in both understanding and generating Nepali text.

1 Introduction

In recent years, Natural Language Processing (NLP) has undergone a remarkable evolution, transitioning from traditional rule-based to statistical methods to sophisticated deep learning architectures. The initial approaches, such as n-grams (Goodman, 2001) and rule-based systems, laid the groundwork for understanding language. But these methods faced significant limitations in handling the complexities of natural language and general human communication, which often involves subtle nuances, contextual dependencies, and varying linguistic structures.

The introduction of Recurrent Neural Networks (RNNs) (Mikolov et al., 2010) and Long Short-Term Memory networks (LSTMs) (Sundermeyer et al., 2012) for language modeling marked a significant advancement, allowing models to process sequential data more effectively. RNNs and LSTMs brought notable improvements in tasks like language modeling and sequence prediction. However, they still encountered challenges with long-range dependencies and computational efficiency, which limited their scalability and performance. These models require substantial computational resources and face difficulties in maintaining consistent performance across varying lengths of text and contexts. The development of the self-attention mechanism (Vaswani et al., 2017) marked a pivotal moment in NLP, enabling models to capture dependencies in text more effectively. The self-attention mechanism, integral to the Transformer architecture, allows models to weigh the importance of different words in a sequence, facilitating a more nuanced understanding of context. This was further enhanced by the concept of self-supervised model pre-training, where models like ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and GPT (Radford et al., 2018), leveraged vast amounts of unlabeled text data to learn general language representations. These models demonstrated unprecedented performance improvements across a range of NLP tasks.

Further advancements in NLP include instruction tuning (Wei et al., 2021; Wang et al., 2022), which trains models on instruction-output pairs to improve their ability to follow user commands.This method enhances the model's ability to follow specific user commands and adapt to diverse application scenarios. Instruction tuning has proven effective in improving the versatility and responsiveness of models, allowing them to better handle varied tasks and user interactions. Models, through unsupervised pre-training on large corpora, and

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 9–16
January 19, 2025.

instruction-tuning have demonstrated the ability to generalize across various tasks, achieving state-of-the-art results.

Nepali is spoken by over 32 million people worldwide. Syntactically, the Nepali language differs significantly from English. In English, the typical sentence structure follows a Subject-Verb-Object (SVO) order whereas Nepali employs a Subject-Object-Verb (SOV) structure (Timilsina et al., 2022). Nepali language incorporates a complex system of noun, adjective and verb inflections. Nouns have a system of gender, case and number (Bal, 2004). This fundamental difference in syntactic arrangement highlights the unique characteristics of Nepali and underscores the challenges pertinent to natural language processing tasks in the Nepali language.

Our motivation for developing a monolingual language model for Nepali language comes from recent advancements in natural language processing, particularly the success of large-scale pre-trained models. However, the majority of these developments have focused on high-resource languages, leaving a gap in the availability of robust models for low-resource languages like Nepali. To address this disparity, we developed pre-trained monolingual language models for the Nepali language. Our first steps included compiling a dataset, large enough to develop pre-trained language models. We assembled approximately 27.5 GB of text data by scraping the top 99 Nepali news websites, representing the largest Nepali language dataset to date. We also used an instruction-tuning dataset to explore the potential of instruction-tuned models for Nepali, providing a foundation for future advancements.

This paper outlines our methods for developing pre-trained models and presents a thorough evaluation of their performance and comparison with existing models, setting new standards for Nepali NLP and contributing significantly to research on low-resource languages.

2 Related Works

The development of pre-trained language models has been foundational in advancing NLP, and a variety of approaches have emerged over the years. In this section, we briefly review the key methods that have influenced the creation and evolution of language models.

2.1 Unsupervised Pre-training Approaches

Early methods for developing pre-training language models utilized unsupervised learning to create generalized representations, with word embeddings like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) establishing the foundation by learning vector representations of words from large text corpora. These embeddings enhanced performance across various NLP tasks by capturing semantic relationships in a continuous vector space. The advent of contextualized word embeddings marked a significant advancement, exemplified by (Peters et al., 2018), which generated dynamic embeddings based on surrounding text using bidirectional LSTM networks, leading to improved results in benchmarks such as Question Answering and Named Entity Recognition. The introduction of the Transformer architecture (Vaswani et al., 2017) further revolutionized the field, giving rise to models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), DeBERTa (He et al., 2020) and GPT (Radford et al., 2018). BERT and other encoder model's bidirectional training approach allowed it to predict masked words by considering both left and right context, achieving state-of-the-art results across numerous NLP tasks, while GPT's autoregressive method excelled in text generation and completion.

2.2 Multilingual and Monolingual Language Models

The release of multilingual models, such as mBERT (Pires et al., 2019), XLM (Lample and Conneau, 2019) and XLM RoBERTa (Conneau et al., 2020) which includes support for languages like Nepali, has further expanded the accessibility of NLP tools across different languages. While these models offer impressive results for many languages, their performance for languages other than English is still not up to the mark due to limited training data, issues with tokenization, lack of adequate vocabulary, and absence of techniques to handle linguistic diversity.

Recently, numerous powerful monolingual models for languages other than English have emerged showing promising results such as NorBERT (Kutuzov et al., 2021) for Nordic languages, FinBERT (Virtanen et al., 2019) for Finnish language, HerBERT (Mroczkowski et al., 2021) for Polish language, GBERT (Chan et al., 2020) for German

language, Chinese BERT (Cui et al., 2021) for Chinese language, NepBERTa (Timilsina et al., 2022) for Nepali language etc. These models have demonstrated that optimizing tokenizers and architectures for specific languages can lead to substantial improvements in performance.

Few of the models have also focused on the Nepali language. IndicBERT (Doddapaneni et al., 2022) focused on several Indic languages, including Nepali, and demonstrated that language-specific models could outperform their multilingual counterparts on specialized tasks. NepBERTa (Timilsina et al., 2022) introduced a BERT-based language model specifically for the Nepali language, trained on the largest monolingual Nepali corpus with 0.8 billion words collected from various sources. They also established the first Nepali Language Understanding Evaluation benchmark (Nep-gLUE). Similarly, NepaliBERT (Pudasaini et al., 2023) also developed a monolingual BERT model specifically for the Nepali language. These models demonstrate the importance of optimizing tokenizers and architectures for addressing the unique characteristics of individual languages, especially those with complex syntactic and morphological structures like Nepali.

2.3 Instruction Tuning on Low-Resource Language Models

Instruction tuning has recently gained attention as a technique to substantially improve zero-shot performance on unseen tasks (Wei et al., 2021; Wang et al., 2022; Ouyang et al., 2022). This method involves fine-tuning pre-trained models on tasks that require the model to understand and execute explicit instructions, thereby increasing its adaptability and effectiveness across a variety of tasks. While this approach has been widely explored for high-resource languages, its potential in low-resource languages, such as Nepali, remains unexplored.

3 Dataset

This section describes the dataset used in our study, focusing on the methodologies implemented for data collection and preprocessing. Given the necessity for extensive training data in transformer-based language models, we compiled a dataset that is by far the largest one for the Nepali language.

3.1 Dataset Collection

Recently, the rise in digital content in Nepali has led to the increasing number of Nepali-language websites. This has opened doors to creating a comprehensive Nepali language corpus for which we performed web scraping across 99 Nepali news websites. As a result, we were able to gather a dataset totaling 30.4 GB of text data, which is significantly larger than existing resources.

We made a deliberate decision not to include existing datasets, such as the Nepali Wikipedia (Arora, 2020) dataset which is less than 1GB, the OSCAR dataset (Suarez et al., 2019) which is approximately 3GB, and the 12.5 GB dataset from NepBERTa (Timilsina et al., 2022). This choice was based on the fact that all these existing datasets were also scraped from news websites, which overlapped with our sources. To avoid duplication and ensure the uniqueness of our dataset, we opted to scrape all content from scratch.

For Instruction Tuning, we utilized the publicly available Nepali alpaca dataset (Kafley, 2024)) containing 52k rows of instructions. We cleaned the data by removing/translating all the non-Nepali texts. After the cleaning process, we achieved 40k rows of instructions.

3.2 Dataset Preprocessing

Following the data collection phase, we implemented a preprocessing pipeline to enhance the quality of the dataset. First, we implemented a deduplication process to remove redundant content, which was essential due to the extensive nature of our data collection. To address the challenge of multilingual content often found on news websites, we removed or translated the languages other than Nepali depending upon the context. We also developed specialized scripts to remove noise, eliminating non-textual elements such as HTML tags, special characters, and formatting artifacts common in web-scraped data. Additionally, text normalization techniques, including Unicode normalization, were applied to standardize character representation and maintain consistency. After these preprocessing steps, the dataset was refined and reduced to 27.5 GB, ensuring it was clean and well-suited for model training.

3.3 Tokenization

Tokenization is a crucial preprocessing step in natural language processing that breaks text into smaller

units, such as words or subwords, enabling effective processing by language models. Traditional word piece tokenization methods (Wu et al., 2016) rely on splitting text based on spaces and punctuation and often encounter limitations with out-of-vocabulary (OOV) words and morphological variations, leading to potential information loss. In contrast, Byte-Pair Encoding (BPE) (Sennrich et al., 2016) tokenization addresses these limitations by segmenting words into subword units. BPE improves the handling of rare or unseen words by breaking them down into more manageable subword units, which helps in retaining meaningful information and maintaining consistency across different word forms. This method provides optimal balance between vocabulary size and coverage, which is particularly beneficial for morphologically rich languages like Nepali, where word forms can vary significantly.

For our study, we utilized the entire dataset to create two different BPE tokenizers, one with a vocabulary size of 30,522 and another with a vocabulary size of 50,256. These tokenizers were designed to optimize the balance between computational efficiency and linguistic coverage, ensuring that our language models could effectively process and understand Nepali text.

4 BERT & RoBERTa

BERT and RoBERTa are both built upon the transformer encoder architecture, which serves as the foundation for their robust natural language processing capabilities. While they share this common architecture, they differ significantly in their training methodologies and objectives. BERT, introduced by (Devlin et al., 2019), employs two distinct pretraining objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In the MLM task, certain words within a sentence are intentionally masked, and the model's goal is to predict these masked words using the context provided by the surrounding words. Similarly, in the NSP task, the model is presented with pairs of sentences and must determine whether the second sentence logically follows the first or if it is a random, unrelated sentence. In contrast, RoBERTa(Liu et al., 2019) focuses solely on the MLM objective, excluding the NSP task altogether, which has been shown to perform better in various benchmarks.

For our study. we pretrained single BERT (Devlin et al., 2019) variant comprising 110 million parameters using the tokenizer of vocabulary size 30,522. Similarly, we also pretrained the single RoBERTa(Liu et al., 2019) variant, also comprising 110 million parameters, using the tokenizer of vocabulary size 50,257.

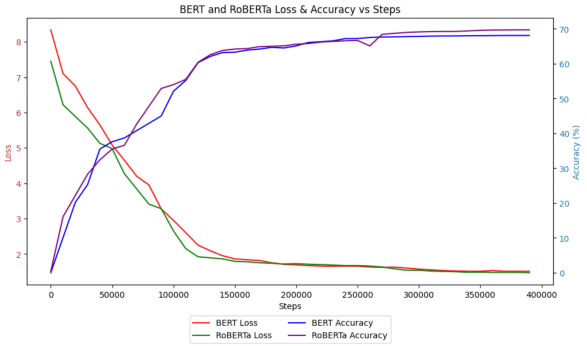

Figure 1: Loss and accuracy of the BERT and RoBERTA model compared with steps

In the case of both BERT (Devlin et al., 2019) (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), we used a batch size of 256 and trained for 400k steps. We chose the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and included an L2 weight decay of 0.01. We also implemented a learning rate warmup for the first 10,000 steps, followed by a linear decay to ensure smooth training. To improve generalization, we set a dropout probability of 0.1 on all layers. For activation, we used the Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel, 2016) function.

The training loss and accuracy trends for BERT and RoBERTa models are illustrated in Figure 1. For BERT, the training loss starts at 8.34 and gradually decreases to 1.51 after 400k steps, while accuracy improves from 3% to 68.11%. In comparison, RoBERTa begins with a training loss of 7.45, which steadily drops to 1.47 by 400k steps, achieving a slightly higher accuracy of 69.72%. The figure underscores RoBERTa's faster convergence and marginally better performance than BERT.

5 GPT-2

In the case of GPT-2 (Radford et al., 2019), we pretrained the 124M parameter model using the causal language modeling (CLM) objective, as described in the original GPT-2 paper (Radford et al., 2019). CLM, as outlined in the original GPT-2 paper (Radford et al., 2019), is designed to predict the next word in a sequence given the preceding con-

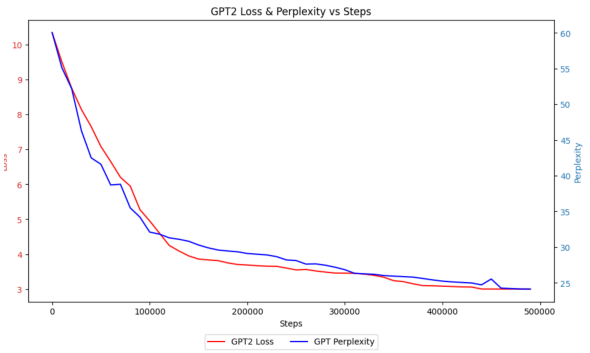

Figure 2: Loss and Perplexity of the GPT2 model compared with steps

text. Unlike BERT's masked language modeling, which predicts masked words from the surrounding context, CLM operates in a left-to-right manner. This means that the model generates text sequentially, using only the preceding words to predict the next word in the sequence. We used a batch size of 256 and trained for 500k steps. We used the Adam optimizer with a learning rate of 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and an L2 weight decay of 0.01. Similar to BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), we implemented a learning rate warmup over the first 10,000 steps, followed by a linear decay. To regularize the model, we set the dropout probability to 0.1 across all layers. This model was also trained using the GELU activation function (Hendrycks and Gimpel, 2016). Furthermore, we performed instruction tuning on the pretrained model using supervised fine-tuning. We used a batch size of 16, the Adam optimizer with a learning rate of 1×10^{-4}, and an attention dropout probability of 0.01. The training loss and perplexity trends for GPT-2 are shown in Figure 2. Initially, the training loss starts at 10.34 and steadily decreases to 3.001 after 500k steps. Similarly, perplexity drops from 60.03 to 24.13 over the same period. The figure illustrates GPT-2's significant improvement in model performance, with both loss and perplexity showing a consistent downward trend, reflecting the model's enhanced ability to predict the next token with greater accuracy as training progresses.

6 Evaluation

We conducted a thorough evaluation across several NLP tasks. Our evaluation includes both Natural Language Understanding (NLU) for encoder models and Natural Language Generation (NLG) tasks for decoder models.

6.1 Evaluating BERT & RoBERTa

For the evaluation of encoder-based models (BERT & RoBERTa) (Devlin et al., 2019; Liu et al., 2019), we used the Nepali Language Evaluation Benchmark, or Nep-gLUE (Timilsina et al., 2022). It consists of four tasks, including Named Entity Recognition (NER), Part-of-Speech (POS) Tagging, text classification, and categorical pair similarity. We used a batch size of 32 and fine-tuned for 3-10 epochs with multiple learning rates (5e-5, 4e-5, 3e-5, 2e-5, and 1e-5) over the data for all Nep-gLUE tasks. For each task, we selected the best-performing model on the test set.

Our models outperformed all existing models across all tasks, scoring 95.60 on Nep-gLUE (Timilsina et al., 2022) benchmark, a result that can be primarily attributed to the large and diverse training corpus we used. The scale of the data allowed the models to generalize better and capture a broader range of linguistic patterns, leading to improved performance.

6.2 Evaluating GPT-2

There were no existing benchmarks for NLG tasks, so we used abstractive summarization for the evaluation of GPT-2 (Radford et al., 2019). We used a publicly available summarization dataset (Bhandari, 2024) and fine-tuned both the GPT-2 model and GPT-2 Instruct model. The dataset consists of 7,258 data points, where we used 5,806 (80%) data points for training and the remaining 1,452 (20%) data points for evaluation. For finetuning, we used a batch size of 8 and trained for 3,000 steps. We used the ROUGE score (Lin, 2004) (Lin and Och, 2004) as our evaluation metric.

One of the things we observed was that the model tends to hallucinate when given very long contexts, and it did not perform well on long inputs, typically those exceeding 400 tokens. A key reason for this behavior can be traced to the model's training. The model was originally trained on sequences of 512 tokens, which limits its ability to handle longer sequences effectively, resulting in average ROGUE scores.

7 Results

We evaluated our pretrained Nepali language models on various Natural Language Processing tasks, comparing their performance with existing models. The results of the evaluation are summarized in table 1 and table 2.

Model	PARAMS	NER	POS	CC	CPS	Nep-GLUE Score
multilingual BERT (Devlin et al., 2019)	172M	85.45	94.65	91.08	93.60	91.19
XLM-Rbase (Conneau et al., 2020)	270M	87.59	94.88	92.33	93.65	92.11
NepBERT (Pudasaini et al., 2023)	110M	79.12	90.63	90.98	91.05	87.94
NepaliBERT (Rajan, 2021)	110M	82.45	91.67	90.10	89.46	88.42
NepBERTa (Timilsina et al., 2022)	110M	91.09	95.56	93.13	94.42	93.55
BERT (Ours)	110M	93.57	96.94	94.47	95.72	95.18
RoBERTa (**Ours**)	125M	**93.74**	**97.52**	**94.68**	**96.49**	**95.60**

Table 1: Nep-gLUE Test Result

Model	PARAMS	ROUGE-1	ROUGE-2	ROUGE-L
distilgpt-nepali (Maskey, 2022)	88.2M	10.16	8.63	9.19
GPT-2 (Ours)	124M	19.66	14.51	16.84
GPT-2-Instruct (**Ours**)	124M	**20.42**	**15.89**	**17.76**

Table 2: Performance comparision on summarization task

For BERT and RoBERTa in table 1, we used the NepGLUE benchmark and evaluated models, against existing monolingual and multilingual models. Both of our models outperformed the previous state-of-the-art, with RoBERTa achieving the highest overall NepGLUE score of 95.60. In particular, our models demonstrated superior performance on every task, reflecting the effectiveness of our models.

In the summarization task in table 2, we compared our GPT-2 models, including an instruction-tuned variant with existing (Maskey, 2022) model. Our GPT-2 models demonstrated substantial improvements across all ROUGE metrics, with the GPT-2-Instruct model achieving the highest scores of 20.42 (ROUGE-1), 15.89 (ROUGE-2), and 17.76 (ROUGE-L).

8 Conclusion

Our study reports significant progress in the field of Natural Language Processing (NLP) for the Nepali language, achieved through the development and evaluation of pre-trained large language models. Our key contributions include the development of by far the largest monolingual corpus for Nepali language and the pretraining of RoBERTa and BERT variants, as well as the introduction of the first GPT-2 model specifically designed for Nepali.

Extensive evaluations conducted on the NepGLUE benchmark and abstractive summarization tasks reveal that our models outperform existing state-of-the-art methods, demonstrating substantial improvements across a range of NLP tasks. By addressing both encoder and decoder architectures, our research emphasizes the potential for optimizing language models tailored to low-resource languages. Our findings not only contribute to the existing body of knowledge but also lay the groundwork for future research and applications in low-resource settings. We anticipate that these insights and benchmarks will inspire further innovations in the field, ultimately resulting in more effective and inclusive NLP research.

9 Acknowledgement

We extend our sincere gratitude to Google's TPU Research Cloud program for granting us free and unlimited access to TPU v4-8 for 30 days and School of Engineering, Kathmandu University for providing us with Nvidia GeForce RTX 3090 GPUs. This research would not have been possible without the unwavering support of the TPU Research Cloud team and School of Engineering, Kathmandu University.

References

Gaurav Arora. 2020. inltk: Natural language toolkit for indic languages. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 66–71. Association for Computational Linguistics.

Bal Krishna Bal. 2004. *Structure of Nepali Grammar*.

Sanjeev Bhandari. 2024. Xlsum-nepali-summerization-dataset.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796. International Committee on Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than text generators. In *International Conference on Learning Representations (ICLR)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116v2*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. *arXiv preprint arXiv:2212.05409*.

Joshua Goodman. 2001. A bit of progress in language modeling. *Expert Systems with Applications*, 41(3):853–860.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units. *arXiv preprint arXiv:1606.08415*.

Saugat Kafley. 2024. alpaca-nepali-sft.

Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for norwegian. *arXiv preprint arXiv:2104.06546*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Utsav Maskey. 2022. Distilgpt2-nepali.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomaš Mikolov, Martin Karafiát, Lukáš Burget, Jan "Honza" Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. Herbert: Efficiently pretrained transformer-based language model for polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27730–27744.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Shushanta Pudasaini, Subarna Shakya, Aakash Tamang, Sajjan Adhikari, Sunil Thapa, and Sagar Lamichhane. 2023. Nepalibert: Pre-training of masked language model in nepali corpus. In *7th International Conference on IoT in Social, Mobile, Analytics and Cloud*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *OpenAI Blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Rajan. 2021. Nepalibert.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Pedro Ortiz Suarez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Interspeech*.

Sulav Timilsina, Milan Gautam, and Binod Bhattarai. 2022. 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 12th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics (ACL).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.