**Proceedings of 10th IOE Graduate Conference**
***Peer Reviewed***
**ISSN: 2350-8914 (Online), 2350-8906 (Print)**
**Year: 2021 Month: October Volume: 10**

# Comparative Analysis of Nepali News Classification using LSTM, Bi-LSTM and Transformer Model

Saroj Sharma Wagle [a], Sharan Thapa [b]

[a, b] *Department of Electronics and Computer Engineering, Pashchimanchal Campus, IOE, Tribhuvan University, Nepal*
**Corresponding Email**: [a] saroj.732534@pasc.tu.edu.np, [b] sharant@ioepas.edu.np

**Abstract**
This paper aims to tackle the task of Nepali News Classification by using different deep learning algorithms and extends previous works on Nepali News Classification by comparing results for these approaches traditional methods. The different algorithms compared are: Long Short-Term Memory, Bidirectional Long Short-Term Memory and Transformer. News classification is done into 17 categories using around 200k news articles scraped from various news portals. The dataset was divided into training and testing sets with a split of 80-20. Furthermore, 20% of the training data was used for validation during training. After training the different models, it was seen that the Transformer obtains the best result for the dataset. The transformer was first pre-trained on all the available training news articles without any labels, using Masked Language Modelling. After the pre-training phase, finetuning gave a result with a training f1-score of 96.54% and a testing f1-score of 95.45%, outperforming all models tried out in this paper.

**Keywords**
BERT, BiLSTM, LSTM, News Classification, News Corpus, Transformer

## 1. Introduction

Text classification is a classic problem in NLP [1, 2]. This problem involves classifying a given piece of text into respective classes. Several algorithms have been used to tackle this problem [2] including Logistic Regression, K-Nearest Neighbors, and Transformers. In the current era, the field of NLP is dominated by Deep Learning based methods. Transformer based architectures are the state-of-the-art models in NLP [3, 4, 5]. News classification is a task that falls under text classification. In this task, given a piece of a news article, a class is predicted for that article based either on the headline or the body of the article. [6] discusses different approaches that have been used for news classification.

News classification is a text classification task, where given a piece of News, a model classifies the article into a list of possible classes. News data is abundantly available on the internet [7, 8] and it makes sense to use this data to build classification models. Also, the news articles can be used to train language models which can be fine-tuned and used in other downstream tasks, not specifically related to news. It also finds application in automated classification of the category of the news article before publishing it in a news portal. The model can automatically decide the category of the news by looking at the contents in the news. This can help the developers build automated pipelines where the content writers can focus on writing the content. Also, the output of the model can serve as feedback for the writer to decide whether a piece they have written is relevant to the category the model is predicting. Inconsistencies between the model predictions and the writer's decided category can provide valuable information both on the part of the writer and ways to make the model better.

Nepali text data is becoming adequately available, especially with the boom of online media in Nepal. Nepali textual data is also available from several social media platforms like Facebook and Twitter. However, these data are unstructured and usually have no particular labels associated with them. An exception to this rule is news websites, where categories are provided by online news platforms to articles. The categories can be used as labels to train different models.

The current research [9, 6, 10, 11, 12, 13, 14, 15] in Nepali News Classification use techniques like ANN

and some use RNN. However, better deep learning models have been developed since then and also the amount of available data has also increased. This paper uses more data than existing works and also makes use of Transformer architecture, which has shown promising in other NLP tasks.

## 2. Related Works

Previous works like [9, 6, 10, 11, 12, 13, 14, 15] explore classification of Nepali news using SVMs and deep neural networks. These works have not explored the application of transformer architectures, which have shown state-of-the-art performance [16] in other languages, to Nepali language. Hence, the contribution of this paper is the inclusion of the transformer model for Nepali News classification and comparison of model performances for different parameter values of Logistic Regression, LSTM, GRU, RNN and Transformer models. For comparison of model performance, f1 score will be used along with precision and recall. An extensive list of baselines is not available for different models. This paper aims to establish baseline performance for comparing the performance of LSTM and BERT for news classification tasks. Transformer architecture-based models have shown state-of-the-art results [16] for other languages. Establishing the performance of a BERT based model for Nepali Language can be a contribution to the field of Nepali NLP as well.

## 3. Methodology

### 3.1 Flow Diagram

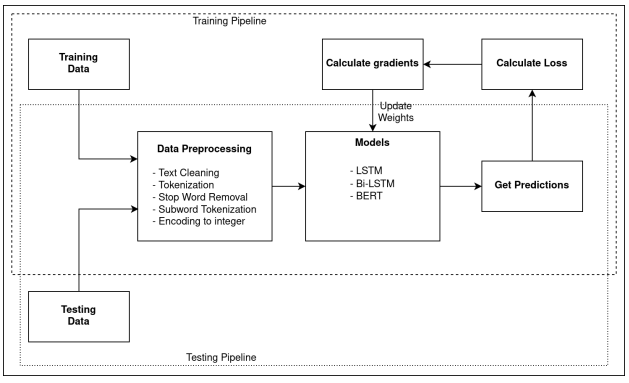


**Figure 1:** The flow diagram

Figure 1 shows the different pipelines for training and testing. For training, we preprocess the news article. In this phase, text cleaning, tokenization, stop word removal, subword tokenization and integer encoding are applied. Then, the data is fed to different models: LSTM, Bi-LSTM, and BERT. Then, the predictions are obtained. From the predictions and the actual label, we can obtain a loss. The loss is used to calculate the gradient. Then, the weights are updated using gradients. The testing pipeline is similar to the training phase. The only difference is that we do not need to calculate the loss and gradients during testing. We only care about getting the predictions.

### 3.2 Data Collection

A dataset to be used for classification is not readily available for Nepali and hence it needed to be scraped from the web. Many news portals are available and the data was scraped from those websites[17, 7] including OnlineKhabar, Ekantipur, NagarikNews, CrimeNewsNepal, Setopati, Ratopati, NepalKhabar. The dataset contains 17 different categories of news, including: Crime, Politics, Sports, Entertainment, etc. The total number of documents scraped was 200k for classification. The exact number of the data was 200,127. Out of the total samples, not all were fit for training the model. 160,101 articles were used for training. 4,172 articles were discarded because they belonged to classes which had less than 200 articles. Training the model on these data tended to hurt the performance of the model so, by preliminary elimination, these were discarded. These data were present because different news portals have different unique categories for some special news. For example, a category that was removed was “World Cup”, which had few articles present in it. After cleaning the data, in the final dataset, there are 17 categories.

### 3.3 Data Preprocessing

#### 3.3.1 Data Cleaning

The data extracted from the news sites is not purely in Nepali script. Traces of English and special characters were present in the data. Other problems include the appearance of other unicode characters in the text as well. The data needs to be cleaned to only keep Nepali Language. Cleaning involved removing english characters as well as special symbols from the data. Unnecessary whitespaces will need to be removed as well. Apart from that the text needs to be tokenized as well. The labels i.e the news categories need to be encoded as well before processing.

| S.No. | Categories | Data |
|---|---|---|
| 1 | Politics | 11963 |
| 2 | World | 11924 |
| 3 | Society | 11860 |
| 4 | Sports | 11829 |
| 5 | Business | 11804 |
| 6 | Crime | 11790 |
| 7 | Technology | 11788 |
| 8 | International | 11787 |
| 9 | Entertainment | 11782 |
| 10 | Diaspora | 11748 |
| 11 | Economy | 11740 |
| 12 | Tourism | 11729 |
| 13 | Automobiles | 11722 |
| 14 | Employment | 11708 |
| 15 | National | 11674 |
| 16 | Health | 11672 |
| 17 | Opinion | 11607 |
| | **Total** | **200127** |

**Table 1:** The class distribution in the dataset.

### 3.4 Data Tokenization

The next step is the tokenization of the article into words. This is done because only after this step we can remove stop words from the article. The articles were broken into words using whitespaces. This yields different words in the sentence.

### 3.5 Stop Word Removal

After the article has been broken down into words, we can select stop words from the data and remove them. These words do not carry any meaning and in our task, add unnecessary information.

### 3.6 Sub-word Tokenization

After the article was tokenized and the stop words were removed, the article was joined as a single string again. And then, subword tokenization was applied to the article. Subword tokenization is the process of breaking down a word into several smaller units called subwords. There can be multiple ways to break a word into multiple units so the optimal way of breaking the words is learned from the data itself. For example, the word **learning** could be broken down into: **learn** and **ing**. Also, during training of the tokenizer only (not the models), the stop words were not removed because they could be broken down into sub tokens which the tokenizer can utilize later. During model training, however, subword tokenization was applied on the article after removing stop words from it.

### 3.7 Models

This section elaborates the parameters of different models used. Multiple hyperparameters were tested and the values shown in this section showed the best results. However, better parameters could possibly be found by using hyperparameter optimization techniques

#### 3.7.1 LSTM

A LSTM was also trained on the data. An input embedding of size 256 was used. The embedding was trained on the data with the LSTM model. The size of the vocabulary for the embedding was 32000. A LSTM with 128 Cells was used. The output of the LSTM was connected to two fully connected networks. Finally, a softmax on the output gave the probabilities for different classes. The summary of the model parameters are:

- Batch size: 64
- Epochs: 12
- Learning Rate: 0.001
- Optimizer: Adam
- Embedding Dimension: 256
- Dropout: 0.5
- Early stopping patience: 5

#### 3.7.2 Bi-LSTM

Bi-LSTM is a bidirectional LSTM, meaning it processes the input sequentially from both ends. It was also trained on the data. It is a non causal system as it has access to future information as well. When the model analyzes the input sequence from both sides, it can have even more context and generally performs better than LSTM. The training configurations for the Bi-RNN are exactly the same as the ones used for the LSTM model. The only difference between LSTM and Bi-LSTM is that a bidirectional LSTM is used in this case. The summary of the model parameters are:

- Batch size: 64
- Epochs: 15
- Learning Rate: 0.001
- Optimizer: Adam
- Embedding Dimension: 256
- Dropout: 0.5
- Early stopping patience: 5

#### 3.7.3 BERT

The transformer is the current state of the art model in various tasks, especially for NLP. Therefore, a transformer model was trained on the data as well. The transformer network is the largest of all the models that have been tried in this thesis. Transformer requires lots of data to train. The transformer was trained in two phases. In the first phase, the language model was trained using Masked Language Modelling. In masked language modelling, the model is trained only using the training data, without the labels. A word is masked in a sentence and the model is asked to predict the word given the context. This forces the model to learn information about the word from the context. After the language model was trained, the model was fine tuned for the actual downstream task. In this case, the downstream task was news classification. The language model was fine tuned by adding a fully connected layer at the end. Finally, a softmax on the output gave the probabilities for different classes. The recurrent networks had about 8M parameters. However, the transformer had 68M parameters, making it the largest model. The summary of the model parameters are:

- Batch size: 64
- Epochs: 13
- Learning Rate: 0.001
- Optimizer: Adam
- Embedding Dimension: 768 (The default from BERT)
- Dropout: 0.5
- Early stopping patience: 5

## 4. Results and Analysis

In this section, the results of the training of different models are elaborated. Different model architectures were tried out in this paper, namely: LSTM, Bi-LSTM and Transformer. LSTM, Bi-LSTM and Transformers are deep learning based models. The algorithms are selected to compare deep learning algorithms performance on news classification dataset. LSTM and Bi-LSTM are compared because it can help assess how much better can the model perform if it can read the input sequence from the reverse as well. Bi-LSTM has extra information and the impact of this extra information can be seen in the metrics. Although many metrics have been calculated for the models, it is reasonable to use a single metric for comparing the models. For comparison, we take the average weighted-f1 score as the metric. The average weighted-f1 score will be referred to as f1-score in this section. F1 score is used because it is a standard metric in machine learning and deep learning literature for balanced as well as unbalanced datasets. F1 score is also derived from both precision and recall, which is another reason F1 score is selected.

### 4.1 Results

#### 4.1.1 LSTM

The output presented in the above table shows the news category business with the highest F-1 Score of 94.32% followed by crime with F-1 Score 94.32% whereas the news category opinion has the lowest F-1 Score 91.19% followed by world with F-1 Score 91.31%. The weighted average score for the model is 92.94%. The model is performing well for the overall dataset.

| | precision | recall | f1 |
|---|---|---|---|
| **politics** | 90.71% | 92.63% | 91.66% |
| **world** | 91.63% | 90.98% | 91.31% |
| **society** | 92.62% | 94.67% | 93.64% |
| **sports** | 94.47% | 90.11% | 92.24% |
| **business** | 94.17% | 94.46% | 94.32% |
| **crime** | 94.63% | 94.01% | 94.32% |
| **technology** | 90.92% | 93.88% | 92.37% |
| **international** | 92.44% | 91.07% | 91.75% |
| **entertainment** | 93.98% | 93.70% | 93.84% |
| **diaspora** | 92.29% | 93.56% | 92.92% |
| **economy** | 93.39% | 90.98% | 92.17% |
| **tourism** | 94.40% | 93.98% | 94.19% |
| **automobiles** | 92.02% | 93.71% | 92.85% |
| **employment** | 94.79% | 94.49% | 91.64% |
| **national** | 94.26% | 93.35% | 93.81% |
| **health** | 93.73% | 91.73% | 92.72% |
| **opinion** | 91.26% | 91.12% | 91.19% |
| | | | |
| **weighted score** | 93.04% | 92.85% | 92.94% |

**Table 2:** The model scores for different classes for LSTM

#### 4.1.2 Bi-LSTM

The output presented in the table shows the news category politics with the highest F-1 Score of 94.78% followed by international with F-1 Score 94.56% whereas the news category crime has the lowest F-1 Score followed by automobiles with F-1 Score 92.51%. The weighted average score for the model is 93.65%. The model is performing well for the overall

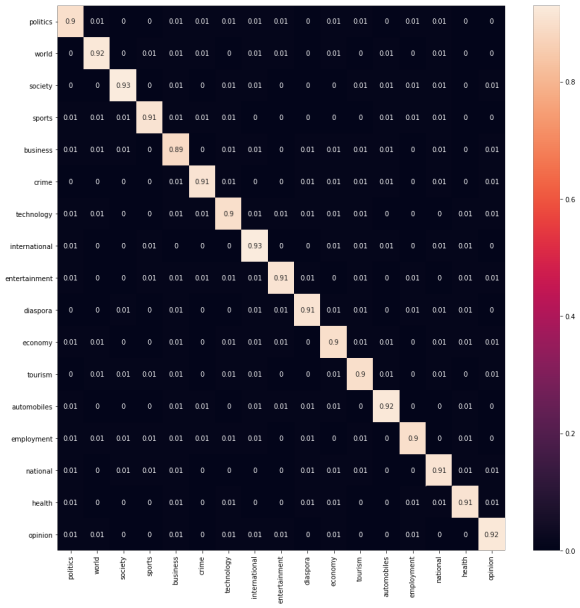


**Figure 2:** The confusion matrix for LSTM

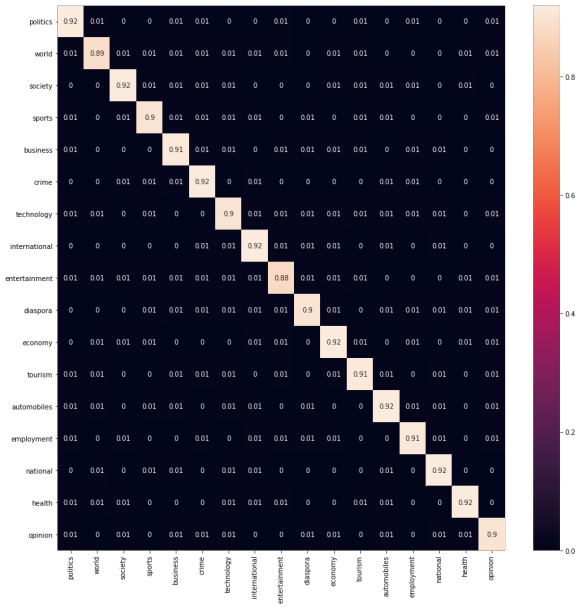


**Figure 3:** The confusion matrix for Bi-LSTM

dataset.

| | precision | recall | f1 |
|---|---|---|---|
| **politics** | 93.52% | 94.04% | 94.78% |
| **world** | 94.49% | 93.50% | 93.99% |
| **society** | 92.25% | 94.52% | 93.38% |
| **sports** | 93.84% | 92.30% | 93.07% |
| **business** | 94.00% | 94.29% | 94.14% |
| **crime** | 92.78% | 92.08% | 92.43% |
| **technology** | 94.47% | 94.45% | 94.46% |
| **international** | 94.48% | 94.63% | 94.56% |
| **entertainment** | 93.71% | 93.71% | 93.71% |
| **diaspora** | 93.69% | 93.85% | 93.77% |
| **economy** | 93.87% | 93.11% | 93.49% |
| **tourism** | 94.74% | 94.25% | 94.50% |
| **automobiles** | 93.19% | 92.09% | 92.64% |
| **employment** | 94.80% | 93.21% | 94.00% |
| **national** | 94.69% | 93.45% | 94.07% |
| **health** | 93.91% | 92.13% | 93.01% |
| **opinion** | 93.13% | 92.85% | 92.99% |
| | | | |
| **weighted score** | 93.86% | 93.44% | 93.65% |

**Table 3:** The model scores for different classes for Bi-LSTM

#### 4.1.3 BERT

The classification model BERT was trained with the dataset using 80-20 for training and testing. The output presented in the table shows the news category economy with the highest F-1 Score of 96.47% followed by crime with F-1 Score 96.01% whereas the news category employment has the lowest F-1 Score 93.62% followed by business with F-1 Score 94.85%. The weighted average score for the model is 93.24%.

| Categories | precision | recall | f1 |
|---|---|---|---|
| **politics** | 96.27% | 94.77% | 95.51% |
| **world** | 95.90% | 95.28% | 95.59% |
| **society** | 96.53% | 94.94% | 95.73% |
| **sports** | 95.73% | 94.45% | 95.09% |
| **business** | 94.61% | 95.10% | 94.85% |
| **crime** | 96.96% | 95.07% | 96.01% |
| **technology** | 94.14% | 96.09% | 95.10% |
| **international** | 94.46% | 96.44% | 95.44% |
| **entertainment** | 95.43% | 94.97% | 95.20% |
| **diaspora** | 94.95% | 95.81% | 95.38% |
| **economy** | 96.25% | 96.68% | 96.47% |
| **tourism** | 94.19% | 94.83% | 94.51% |
| **automobiles** | 94.83% | 94.65% | 94.74% |
| **employment** | 93.70% | 93.54% | 93.62% |
| **national** | 94.60% | 95.74% | 95.16% |
| **health** | 96.50% | 94.02% | 95.24% |
| **opinion** | 95.53% | 96.64% | 96.08% |
| | | | |
| **weighted score** | 95.50% | 95.41% | 95.45% |

**Table 4:** The model scores for different classes for BERT

The confusion matrix is also shown:

### 4.2 Analysis

The LSTM model obtained a training score of 93.55% and a testing score of 92.94%. This is a reasonable score and LSTM can deal with long sequences as well.

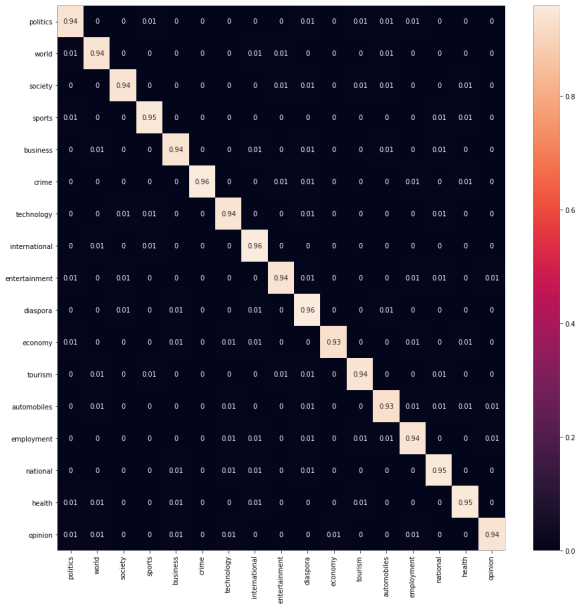


**Figure 4:** The confusion matrix for BERT

Bidirectional variant of the LSTM was trained too. The Bi-LSTM model obtained a training score of 94.58% and a testing score of 93.65%. This is slightly better than the unidirectional version with a slight improvement of around 1%. The parameters have however doubled for the LSTM cell. This is the best result for all the recurrent architectures.

Finally, the Transformer model has a training score of 96.54% and a testing score of 95.45%. The transformer is performing the best on the data. The transformer is however around 8 times larger than the largest recurrent model, the Bi-LSTM. This means that proper training of a transformer architecture requires a lot of training data. Since we are pertaining to the available news corpus, the model is able to learn the representation of different news articles. The language model was further pre-trained using the Nepali Wikipedia corpus dump.

## 5. Conclusion

News classification was performed on data scraped from several online news portals. The data had 200,127 samples out of which 160,101 data were used for training. Some data was discarded because they were from categories which had only a few data, which would not be sufficient for training them. The algorithms LSTM, Bi-LSTM and Transformer were used for classifying news into 17 categories. Out of all the models used, the transformer performed the best with a training f1-score of 96.54% and a testing f1-score of 95.45%. The results illustrate that all the models can be used for Nepali News Classification.

## 6. Future Works

Training deep learning based models requires large amounts of data. More data can be collected and training can be performed on such a large corpus. The original architecture of BERT was trained on a very large corpus. The size of the current dataset is small compared to the original corpus. It is possible to improve the model performance using pre-training using MLM. Also, pretraining language models for LSTM can be done as well. Semi-supervised and Self-supervised methods for deep learning can further be used to leverage unlabeled data. The distribution of the classes is uneven in the dataset before selecting only the 17 categories. In the data collection phase, it was observed that some news categories have little data only. These categories will not get good scores. Techniques like SMOTE can be used to generate synthetic data for these classes. Ensembles of different models can be used to further improve the actual model scores. Alternative architectures can be explored. Algorithms like Neural Architecture Search can be used to find the best models. Hyperparameter tuning can be done using different optimization libraries as well.

## References

[1] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: State of the art, current trends and challenges. *arXiv preprint arXiv:1708.05148*, 2017.

[2] CC Aggarwal and C Zhai. A survey of text classification algorithms. in mining text data 2012 (pp. 163-222).

[3] K Kowsari, K Jafari Meimandi, M Heidarysafa, S Mendu, L Barnes, and D Brown. Text classification algorithms: a survey. information 10 (4): 150. *arXiv preprint arXiv:1904.08067*, 2019.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[5] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2020.

[6] Rohan Katari and Madhu Bala Myneni. A survey on news classification techniques. In *2020 International*

*Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–5. IEEE, 2020.

[7] Online Khabar Nepal. *https://www.onlinekhabar.com/*.

[8] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[9] Sanjeev Subba, Nawaraj Paudel, and Tej Bahadur Shahi. Nepali text document classification using deep neural network. *Tribhuvan University Journal*, 33(1):11–22, 2019.

[10] Tej Bahadur Shahi, Abhimanu Yadav, et al. Mobile sms spam filtering for nepali text using naïve bayesian and support vector machine. *International Journal of Intelligence Science*, 4(01):24–28, 2014.

[11] Oyesh Mann Singh, Ankur Padia, and Anupam Joshi. Named entity recognition for nepali language. In *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*, pages 184–190. IEEE, 2019.

[12] Tej Bahadur Shahi and Ashok Kumar Pant. Nepali news classification using naïve bayes, support vector machines and neural networks. In *2018 International Conference on Communication Information and Computing Technology (ICCICT)*, pages 1–5. IEEE, 2018.

[13] Birat Bade Shrestha and Bal Krishna Bal. Named-entity based sentiment analysis of nepali news media texts. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 114–120, 2020.

[14] Kamal Acharya and Subarna Shakya. An analysis of classification algorithms for nepali news.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[16] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[17] Kantipur Media Group. *https://ekantipur.com/*.