

Nepali News Classification

1. Dataset/Corpus

For this project, I have used the same dataset from the Abstractive Summarization project. The dataset consists of around 372,000 entries. Each entry consists of the news article, headline, and news category. For this project, we only use the news article and category columns of the dataset.

| category | |
|---------------------------|-------|
| राजनीति | 59975 |
| देश/प्रदेश | 51799 |
| समाज | 50207 |
| अर्थ / वाणिज्य | 43020 |
| खेलकुद | 42367 |
| विश्व | 37817 |
| मनोरञ्जन | 34178 |
| विज्ञान र प्रविधि | 23130 |
| स्वास्थ्य | 22392 |
| शिक्षा | 6755 |
| Name: count, dtype: int64 | |

Fig 1: News articles across different categories

| | title | news | category |
|---|---|---|----------|
| 0 | एमालेको प्रदेश प्रतिनिधिमा उदयपुरबाट सर्वसम्मत | उदयपुर : नेकपा एमाले कोशी प्रदेश कमिटीको प्रथम... | समाज |
| 1 | गौशालाबाट लुटियो ६ लाख ५० हजार रुपैयाँ | महोत्तरी : गौशाला नगरपालिका-११ भरतपुरका शम्भु ... | समाज |
| 2 | आगलागीमा ४ करोड बढीको क्षति, पीडितलाई तत्काल र... | मुगु : जिल्ला सदरमुकाम गमगढीमा शुक्रबार राति ... | समाज |
| 3 | तलेजुको दर्शन गर्न पूर्वराजा ज्ञानेन्द्र शाह भ... | भक्तपुर : पूर्वराजा ज्ञानेन्द्र शाह भक्तपुर आए... | समाज |
| 4 | वीरगन्जको खेतबाट ८ हजार ९१५ पिस लागूऔषध बरामद | वीरगन्ज : वीरगन्ज महानगरपालिका-२२ मनियारीस्थि... | समाज |

Fig 2: Sample news from the dataset

2. Data Cleaning and Preprocessing:

- Removed special characters. (.,/';\{ ...).
- Removed Arabic and Devanagari numerals.
- Removed stopwords for ML and Bi-LSTM models, while for BERT, they were kept intact.
- Removed English characters (Aa-Zz), emails, HTML elements, Emojis, and other unwanted non-Devanagari characters.
- Removed all the articles with news < 30 words.
- The dataset was highly imbalanced, so random undersampling was performed to balance the dataset.

3. Text Vectorization:

- Machine Learning Models: [TF-IDF](#) (vocabulary=52K)
- Bi-LSTM Model: Word Tokenization(vocabulary=52K) + Embedding
- [mBERT](#): WordPiece Tokenization(vocabulary=110K) + Pretrained mBERT Embeddings

4. Models

4.1. Machine Learning Models

Here, I have experimented with 3 popular classification algorithms

- [Naive Bayes](#)
- [Gradient Boosting](#)
- [XG Boost](#)

4.2. Bi-LSTM

This is a simple neural network with an Embedding, Bi-LSTM, and Feed Forward Layer with [softmax](#) activation.

4.2.1 Hyperparameters

| Hyperparameters | Value |
|-----------------------|--------------------------|
| Vocab Size | 52K |
| Input Sequence Length | 256 |
| Embedding Dimension | 50 |
| LSTM Latent Dimension | 8 |
| Dropout | 0.2 |
| Batch Size | 512 |
| Learning Rate | 1e-3 |
| Epochs | 16 (Early Stopping at 9) |

Table 1: Hyperparameters used in training the Bi-LSTM model

4.2.2 Loss Function

The model uses the [Cross Entropy](#) Loss function.

4.3. [mBERT](#)

Here, I have finetuned the mBERT-base-cased model with 110M parameters. Since the Nepali language has no distinction in upper and lower cases, either a cased or an uncased model could work well. However, for languages that have case distinctions, such as English, the choice could be dependent on the task at hand.

4.3.1 Training Hyperparameters

| Hyperparameters | Value |
|-----------------------|-------|
| Input Sequence Length | 256 |
| Batch Size | 32 |
| Learning Rate | 5e-5 |
| Warmup Steps | 1,000 |

| | |
|-------------------|------|
| Epochs | 3 |
| L2 Regularization | 0.01 |

Table 2: Hyperparameters used in finetuning the mBERT model

4.3.2 Loss Function

The model uses the [Cross Entropy](#) Loss function.

7. Evaluation Metric

- [Accuracy](#)
- [Precision](#)
- [Recall](#)
- [F1-Score](#)

8. Results

| Model | Accuracy | Precision | Recall | F1-Score |
|-------------------|---------------|---------------|---------------|---------------|
| Naive Bayes' | 83% | 83% | 83% | 83% |
| Gradient Boosting | 81% | 81% | 81% | 81% |
| XG Boost | 85% | 85% | 85% | 85% |
| Bi-LSTM | 85.95% | 87.38% | 84.74% | 85.69% |
| mBERT | 88.74% | 88.53% | 88.77% | 88.59% |

Table 3: Results Comparison

References

- [1] See, A., Liu, P. J., & Manning, C. D. (2017, April 14). *Get To The Point: Summarization with Pointer-Generator Networks*. arXiv.org. <https://arxiv.org/abs/1704.04368>
- [2] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018, October 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv.org. <https://arxiv.org/abs/1810.04805>