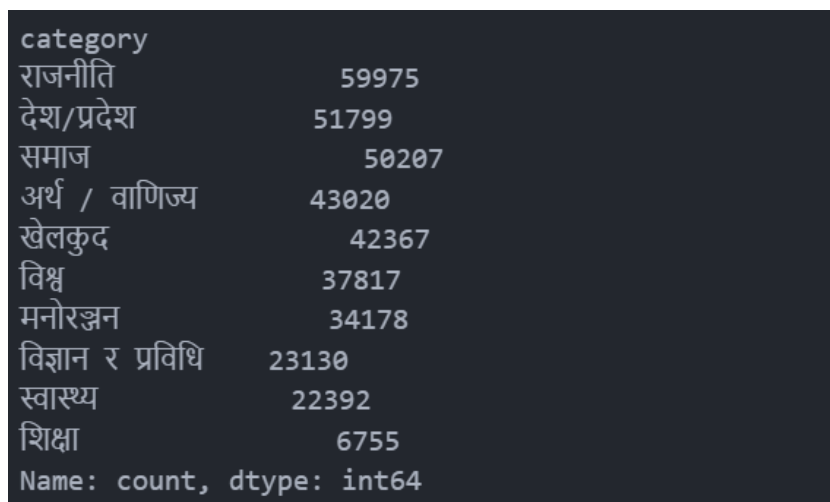


Classification of Nepali News Articles

Dataset:

For the dataset, different online Nepali news portals were scraped and the headlines and news articles across different categories were extracted. The dataset consists of around 372,000 entries.



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: Fig 1: News articles across different categories

Preprocessing:

The cleaning process consists of the following steps:

- Removing extraneous information from the beginning of the news from some of the sources such as Ekantipur, Annapurna, etc. This extra information usually consists of news date, location, author name, etc.
“बाजुरा : बाजुराका विभिन्न ठाउँमा लगाइने मार्से (लट्टे) बालीमा किरा लाग्न थालेपछि किसान ...”
“फुङ्लिङ, मंसिर १२ । ताप्लेजुङको पाथीभरा दर्शन आएकी एक महिला तीर्थयात्रीको शुक्रबार मृत्यु ...”
- Removing special characters. (.,/”;\[{ ...), English characters (Aa-Zz), Arabic Numerals (0-9) and Emojis.
- Removing characters outside of the Devanagari range (U+0900...U+097F).
- Removing stopwords(छ, र, पनि, छन्, लागि, भएको) from news articles.

Here is an example of a sample text after going through the above cleaning process:

Original: काठमाडौं । विप्लवको क्रान्ति ! खसी नदिपछि हत्या गरेका थिए शिक्षक श्रेष्ठलाई

After Cleaning: विप्लवको क्रान्ति खसी नदिपछि हत्या शिक्षक श्रेष्ठलाई

- Then, we removed all the articles that consisted of less than 30 words, and similarly, long news articles were clipped to have a maximum length of 250 and 300
- To handle the imbalance data, all the news related to “*sikshya*” category was removed and undersampling was performed so that each category would have a maximum of 25000 instances.
- I found that the instances related to “*desh/pradesh*” which consisted of news from different provinces of Nepal was similar to news from other categories such as “*samaj*”, “*swastha*” and “*rajniti*”. So, instances related to “*desh/pradesh*” were also removed
- The dataset was then divided into training, validation, and test sets each of which consisted of 80%, 10%, and 10% of the overall data respectively.

Here, we train 3 models, 2 of which are Deep Learning LSTM models while one is a Machine Learning model that uses the Naive Bayes Classifier

For Model 1 (Input sequence length of 250 without news from “*sikshya*”):

Architecture:

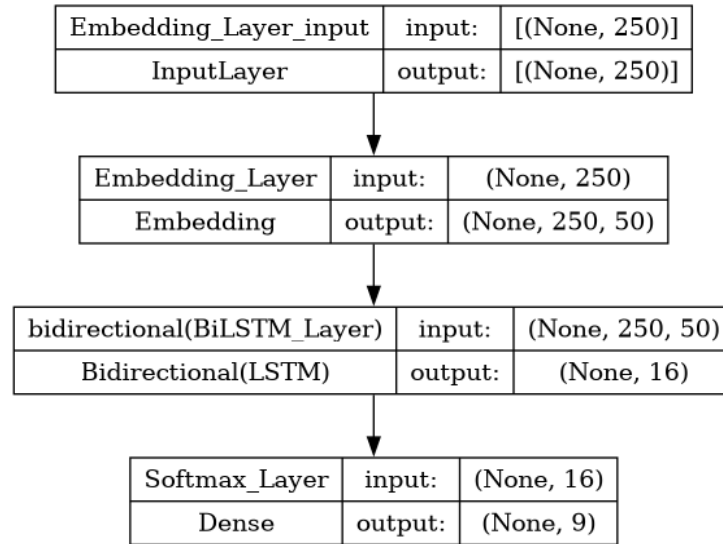


Fig 2: LSTM Architecture for Model 1

The model consists of an Embedding Layer which converts the input sequence into word embeddings. The output of the Embedding Layer is taken by 1 Bi-LSTM layer with a latent dimension of 16. Alongside a dropout of 0.2, a recurrent dropout of 0.2 was also used to prevent overfitting. Finally, the Fully Connected Layer with Softmax Activation produces the probability distribution for 9 classes.

Training:

The model was trained for 16 epochs with a batch size of 512. Additionally, the validation loss was monitored using early stopping with a patience of 3 due to which the training was cut off at 7 epochs.

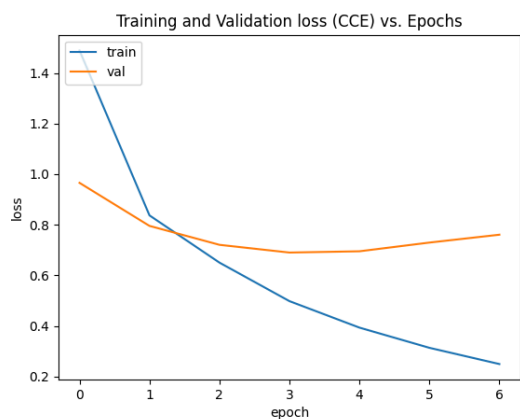


Fig 3: Train and Val Loss (CCE) vs Epoch

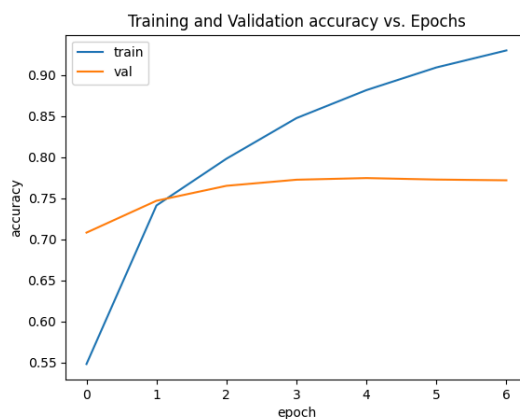


Fig 4: Train and Val Accuracy vs Epoch

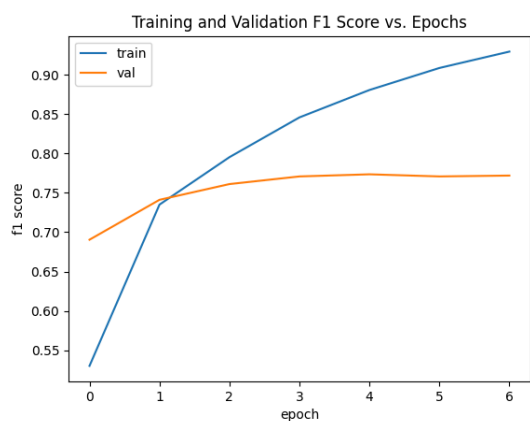


Fig 5: Train and Val F1 Score vs Epoch

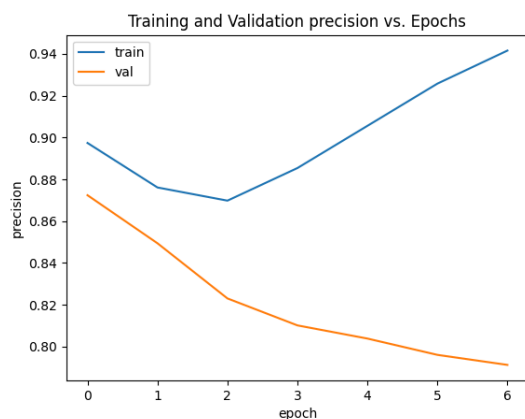


Fig 6: Train and Val Precision vs Epoch



Fig 7: Train and Val Recall vs Epoch

Parameters used in the model:

Parameters	Value
Epochs	16 (cut-off at 7)
Dropout	0.2
Early Stopping	3
Loss	Categorical CrossEntropy Loss (CCE)
Batch size	512
Bi-LSTM Latent Dimension	16
Learning Rate	0.001
Optimization Algorithm	Adam
Sequence Length	250
Vocabulary Size	737756

Model Evaluation:

Metric	Train	Val	Test
CCE Loss	0.2499	0.7609	0.7730
Accuracy	92.97%	77.16%	76.70
F1 Score	92.94%	77.19%	76.81%
Precision	94.16%	79.12%	78.69%
Recall	91.77%	75.64%	75.31%

For Model 2 (Input sequence length of 300 without news from “*sikshya*” and “*desh/pradesh*”):

Architecture:

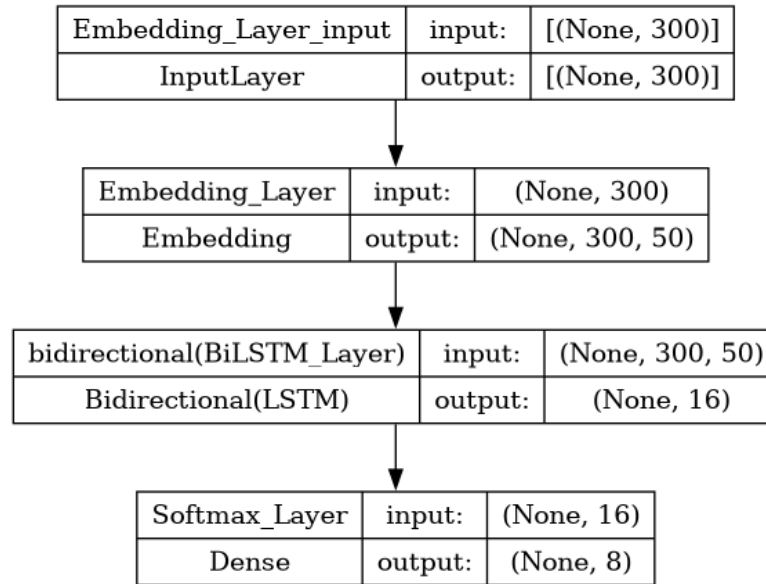


Fig 8: LSTM Architecture for Model 2

The model consists of an Embedding Layer which converts the input sequence into word embeddings. The output of the Embedding Layer is taken by 1 Bi-LSTM layer with a latent dimension of 16. Alongside a dropout of 0.2, a recurrent dropout of 0.2 was also used to prevent overfitting. Finally, the Fully Connected Layer with Softmax Activation produces the probability distribution for 8 classes.

Training:

The model was trained for 16 epochs with a batch size of 512. Additionally, the validation loss was monitored using early stopping with a patience of 3 due to which the training was cut off at 9 epochs.

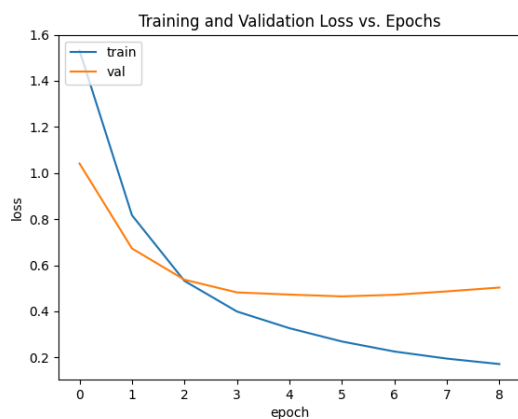


Fig 9: Train and Val Loss (CCE) vs Epoch

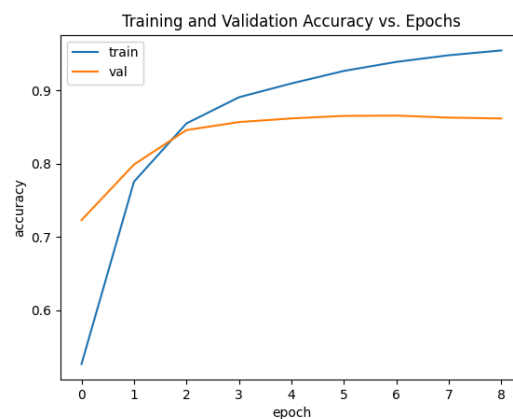


Fig 10: Train and Val Accuracy vs Epoch

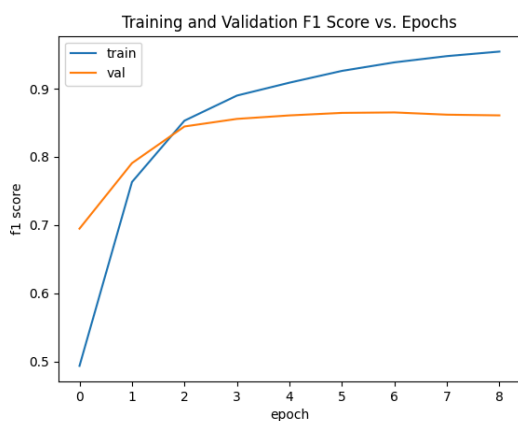


Fig 11: Train and Val F1 Score vs Epoch

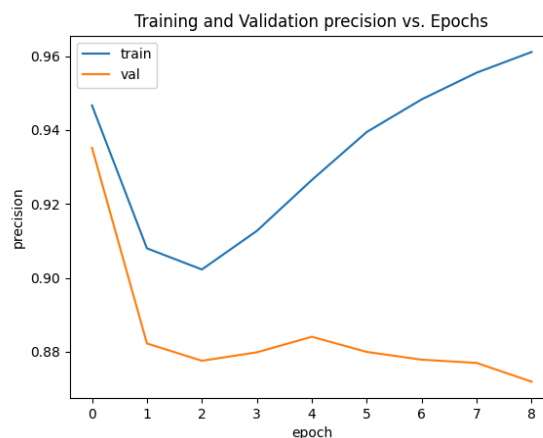


Fig 12: Train and Val Precision vs Epoch

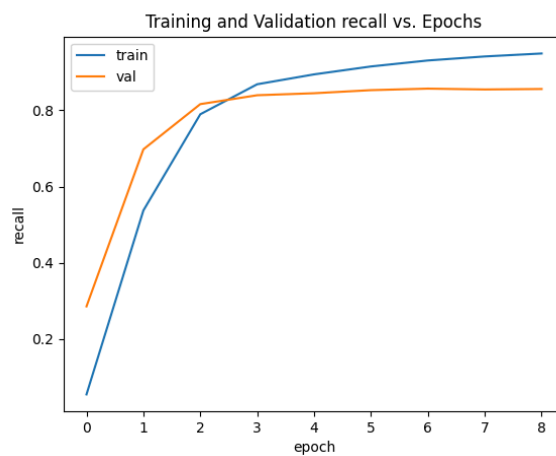


Fig 13: Train and Val Recall vs Epoch

Parameters used in the model:

Parameters	Value
Epochs	16 (cut-off at 9)
Dropout	0.2
Early Stopping	3
Loss	Categorical CrossEntropy Loss (CCE)
Batch size	512
Bi-LSTM Latent Dimension	16
Learning Rate	0.001
Optimization Algorithm	Adam
Sequence Length	300
Vocabulary Size	52000

Model Evaluation:

Metric	Train	Val	Test
CCE Loss	0.1713	0.5033	0.4930
Accuracy	95.47%	86.19%	86.18%
F1 Score	95.43%	86.09%	86.07%
Precision	96.11%	87.19%	87.12%
Recall	94.89%	85.57%	85.38%

For Model 3 (Naive Baye's with input sequence length of 300 without news from “*sikshya*” and “*desh/pradesh*”):

This model uses all the hyperparameter of model 2.

Additionally, we simply use Tf-Idf to represent the text into numbers. The vectorizer considers the individual words as well as the word Bigrams.

	precision	recall	f1-score	support
0	0.92	0.93	0.92	2500
1	0.75	0.77	0.76	2500
2	0.86	0.84	0.85	2500
3	0.98	0.91	0.95	2500
4	0.71	0.64	0.67	2500
5	0.79	0.89	0.84	2500
6	0.84	0.83	0.83	2303
7	0.79	0.83	0.81	2149
accuracy			0.83	19452
macro avg	0.83	0.83	0.83	19452
weighted avg	0.83	0.83	0.83	19452

Result:

Model 2 was found to be the best one among all of the three models as it outperformed the other two models in all of the metrics.