Fake news detection for Hindi language using BERT and LSTM

A Course Project Report Submitted
in Partial Fulfillment of the Requirements
for the Course

CS-529 Topics and Tools in Social Media Data Mining

by

Darshika Verma, Sandeep Agri, Rohan Jaiswal and Himanshu Sharma
(214101014, 214101047, 214101042, 216101102)

under the guidance of

Dr.Sanasam Ranbir Singh

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

CERTIFICATE

*This is to certify that the work contained in this project "**Fake news detection for Hindi language using BERT and LSTM**" is a project work of **Darshika Verma, Sandeep Agri, Rohan Jaiswal and Himanshu Sharma** (**Roll No. 214101014, 214101047, 214101042, 216101102**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision.*

Supervisor: **Dr.Sanasam Ranbir Singh**

Associate Professor,

Department of Computer Science & Engineering,

Indian Institute of Technology Guwahati, Assam.

May, 2022

Guwahati.

Abstract

With the increase in social networks, more number of people are creating and sharing information than ever before, many of them have no relevance to reality. Due to this, fake news for various political and commercial purposes are spreading quickly. Online newspaper has made it challenging to identify trustworthy news sources. In this work, Hindi news articles from various news sources are collected. Preprocessing, feature extraction, classification and prediction processes are discussed in detail. Different machine learning algorithms such as Feed-Forward Neural Network, Support Vector Machine, BERT, Long Short-Term Memory (LSTM) are used to detect the fake news. The preprocessing step includes data cleaning, stop words removal, tokenizing and stemming ,word embedding is used for feature extraction. Feed-Forward Neural Network, Support Vector, Decision Tree Classifiers are used and compared for fake news detection with probability of truth. It is observed that among these three classifiers, SVM achieved best accuracy of 65%.

Contents

List of Figures iii

List of Tables iv

1 Introduction 1

1.1 Recurrent Neural Networks 1

1.2 LSTM Networks 2

1.3 BERT (Bidirectional Encoder Representations from Transformers) 3

1.4 iNLTK 6

2 Proposed Work 7

2.1 Fake news Data Curation 7

2.2 Fake news detection using BERT 7

2.3 Fake news detection using LSTM 9

3 Results and Discussion 10

3.1 Experimental Scenario-1 (BBC-ner) 10

3.2 Experimental Scenario-2 (BBC-POS) 11

3.3 Experimental Scenario-3 (BBC-Random Merge) 11

4 Conclusion and Future Work 12

List of Figures

1.1 Recurrent Neural Networks have loops. 2
1.2 An unrolled recurrent neural network. 2
1.3 Transformer 3
1.4 Encoder decoder working 4
1.5 Multihead Attention 5
1.6 Multihead Attention 5
1.7 Supported Languages by iNLTK 6

2.1 Fake News detection using BERT 8
2.2 Classifier Used 8
2.3 Fake News detection using LSTM 9

List of Tables

3.1 BBC-ner 10

3.2 BBC-POS 11

3.3 BBC-Random Merge 11

Chapter 1

Introduction

1.1 Recurrent Neural Networks

Humans don't start their thinking from scratch every second. As you read this essay, you understand each word based on your understanding of previous words. You don't throw everything away and start thinking from scratch again. Your thoughts have persistence.

Traditional neural networks can't do this, and it seems like a major shortcoming. For example, imagine you want to classify what kind of event is happening at every point in a movie. It's unclear how a traditional neural network could use its reasoning about previous events in the film to inform later ones.

Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist.

In the above diagram, a chunk of neural network, A, looks at some input xt and outputs a value ht. A loop allows information to be passed from one step of the network to the next.

These loops make recurrent neural networks seem kind of mysterious. However, if you think a bit more, it turns out that they aren't all that different than a normal neural

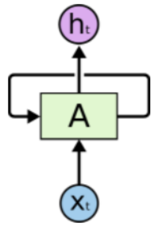

Fig. 1.1 Recurrent Neural Networks have loops.

network. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. Consider what happens if we unroll the loop:

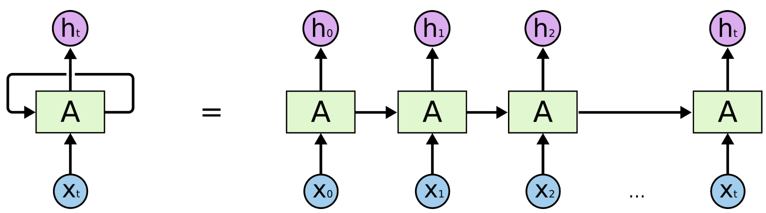

Fig. 1.2 An unrolled recurrent neural network.

1.2 LSTM Networks

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter Schmidhuber (1997), and were refined and popularized by many people in following work.1 They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

The repeating module in a standard RNN contains a single layer. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

1.3 BERT (Bidirectional Encoder Representations from Transformers)

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary.

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).

Fig. 1.3 Transformer

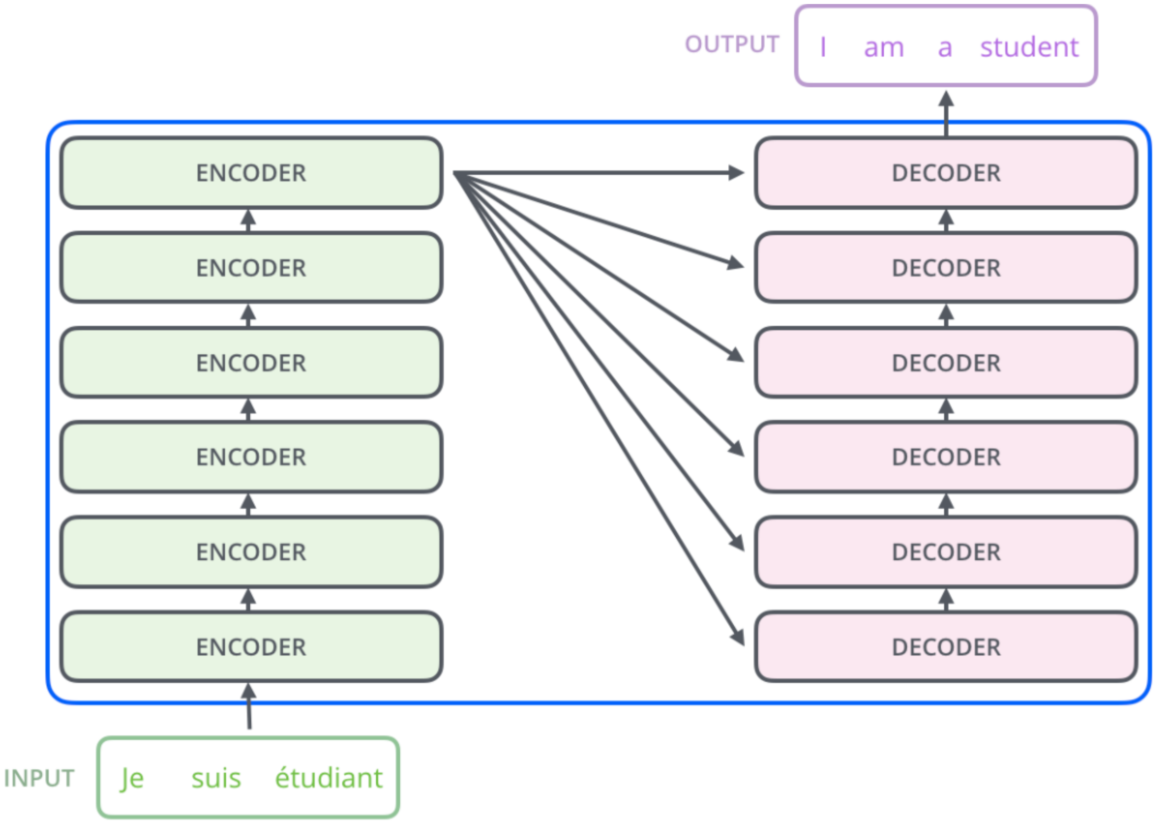

Fig. 1.4 Encoder decoder working

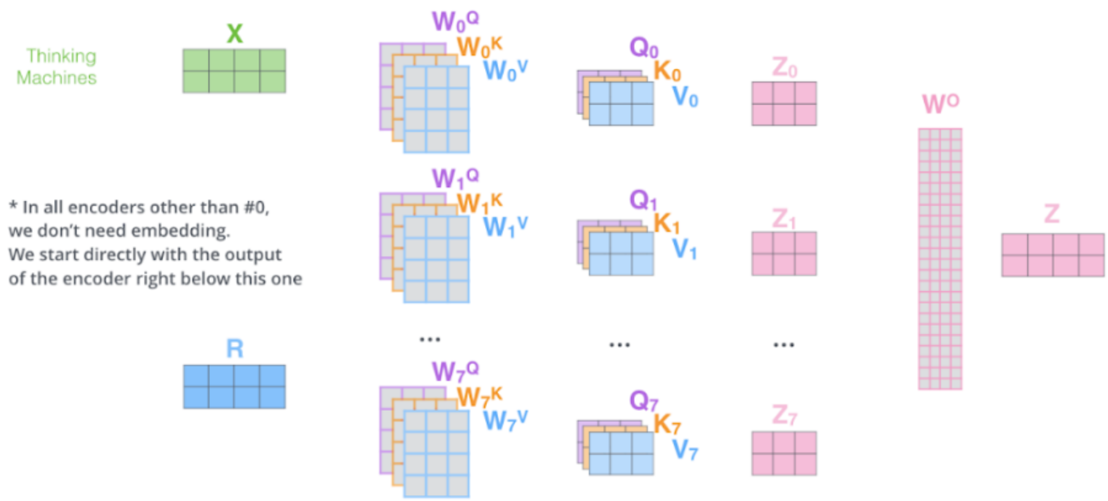

Fig. 1.5 Multihead Attention

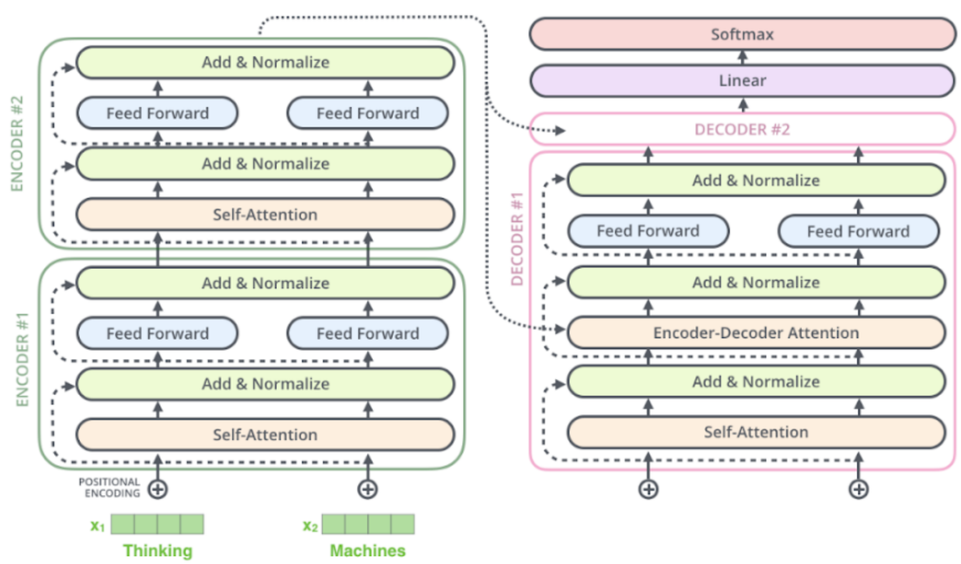

Fig. 1.6 Multihead Attention

1.4 iNLTK

iNLTK, is an open-source NLP library consisting of pre-trained language models and out-of-the-box support for Data Augmentation, Textual Similarity, Sentence Embeddings, Word Embeddings, Tokenization and Text Generation in 13 Indic Languages.

Native languages

Language	Code
Hindi	hi
Punjabi	pa
Gujarati	gu
Kannada	kn
Malayalam	ml
Oriya	or
Marathi	mr
Bengali	bn
Tamil	ta
Urdu	ur
Nepali	ne
Sanskrit	sa
English	en
Telugu	te

Fig. 1.7 Supported Languages by iNLTK

Chapter 2

Proposed Work

2.1 Fake news Data Curation

We have prepared around 300 new Hindi fake news for our analysis. In the new fake news, we embedded the fake news in the body, which is related to the body and headline but in two different scenarios. We have used BBC-ner, BBC- POS and BBC Random merge data sets for fake news detection.

2.2 Fake news detection using BERT

We used the BERT tokenizer to tokenize the heading and body and passed it into the BERT model, which gave us the embedding vector of 768. Then based on that feature vector, we performed the classification using Feed-Forward Neural Network, Decision Tree, and Support Vector Machine.

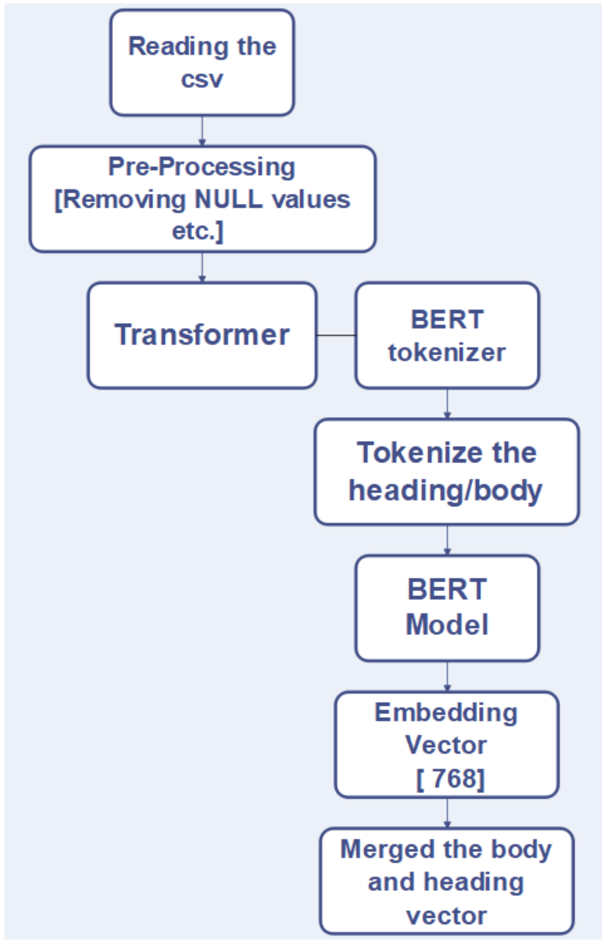

Fig. 2.1 Fake News detection using BERT

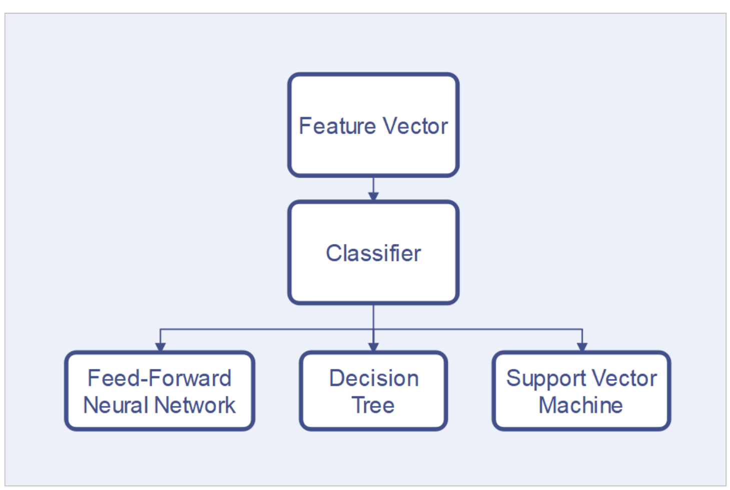

Fig. 2.2 Classifier Used

2.3 Fake news detection using LSTM

We used the iNLTK model to generate the word embedding vector, and after passing it into LSTM, it further fed into Multilayer perceptron and then trained the model.

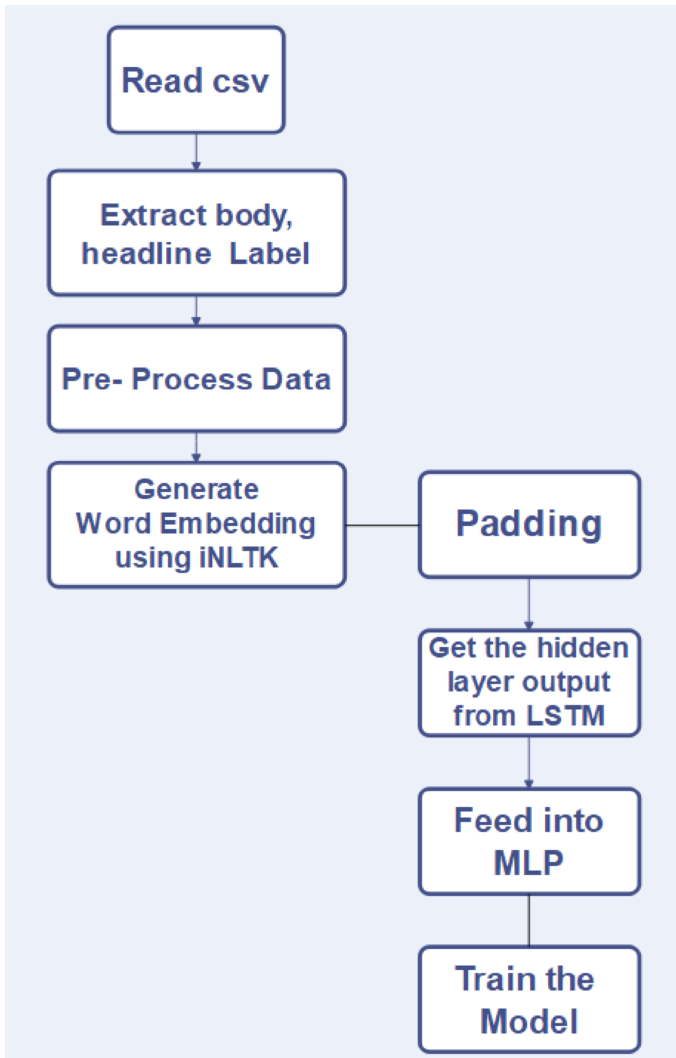

Fig. 2.3 Fake News detection using LSTM

Chapter 3

Results and Discussion

3.1 Experimental Scenario-1 (BBC-ner)

In experimental scenario-1, we used the BBC-ner data set. After extracting features from Fig.2.1, we performed classification on the Feed-Forward neural network, Decision Tree, and Support Vector Machine and calculated Accuracy, F1-score, and Macro Average.

BBC-ner				
Classifier	Multi-Lingual BERT			Indic BERT
Feed-Forward Neural Network	Accuracy		0.5	0.5
	F1-score	Class-0	0.00	0.00
		Class-1	0.67	0.67
	Macro Average		0.33	0.33
Decision Tree	Accuracy		0.37	0.40
	F1-score	Class-0	0.38	0.41
		Class-1	0.38	0.39
	Macro Average		0.38	0.40
Support Vector Machine	Accuracy		0.41	0.41
	F1-score	Class-0	0.31	0.42
		Class-1	0.50	0.41
	Macro Average		0.40	0.41

Table 3.1 BBC-ner

3.2 Experimental Scenario-2 (BBC-POS)

BBC-POS				
Classifier	Multi-Lingual BERT			Indic BERT
Feed-Forward Neural Network	Accuracy		0.5	0.5
	F1-score	Class-0	0.00	0.00
		Class-1	0.67	0.67
	Macro Average		0.34	0.34
Decision Tree	Accuracy		0.49	0.51
	F1-score	Class-0	0.49	0.51
		Class-1	0.5	0.51
	Macro Average		0.38	0.40
Support Vector Machine	Accuracy		0.57	0.56
	F1-score	Class-0	0.6	0.59
		Class-1	0.53	0.52
	Macro Average		0.57	0.56

Table 3.2 BBC-POS

3.3 Experimental Scenario-3 (BBC-Random Merge)

BBC-Random Merge				
Classifier	Multi-Lingual BERT			Indic BERT
Feed-Forward Neural Network	Accuracy		0.485	0.48
	F1-score	Class-0	0.00	0.00
		Class-1	0.65	0.65
	Macro Average		0.33	0.33
Decision Tree	Accuracy		0.60	0.61
	F1-score	Class-0	0.61	0.62
		Class-1	0.60	0.61
	Macro Average		0.60	0.61
Support Vector Machine	Accuracy		0.65	0.67
	F1-score	Class-0	0.64	0.74
		Class-1	0.67	0.58
	Macro Average		0.66	0.66

Table 3.3 BBC-Random Merge

Chapter 4

Conclusion and Future Work

Our project used different machine learning models, such as BERT, LSTM, SVM, Decision Tree, etc. Our project gives 65% accuracy to the Hindi fake news dataset. Our study determines that some of the fundamental algorithms of machine learning give decent results on the critical issue of spreading fake news in the Hindi language. These pre-trained models from iNLTK and BERT can be used as-is for various NLP tasks or fine-tuned on domain-specific datasets.

Further, these models can run on Navbharat ner, navbharat pos, and navbharat ramdommerge.Additionally, fine-tuning can be done on the bert model.