CSPC54 AI-ML
Project Proposal

Group 14

Roll Numbers
106119112
106119100

Group Members
SATYARTH PANDEY
RAJNEESH PANDEY

Section
CSE-B

A Deep Learning Approach for Automatic Detection of Fake News

Introduction

In the emergence of social and news media, data is constantly being created day by day. The data so generated are enormous in amount, and often contain misinformation. Hence it is necessary to check it’s truthfulness. Nowadays Many people mostly rely on social media and many other online news feeds as their only platforms for news consumption.

Therefore, in order to deliver genuine news to such consumers, checking the truthfulness of such online news content is of utmost priority to News Industries. The task is very difficult for a machine as even a human being can not easily distinguish between a genuine and fake article.

This paper proposes two effective models based on Deep learning for solving Fake News detection problems in Online News contents of multiple domains. The techniques are then evaluated on the two renowned Datasets of this field, namely “FakeNews AMT” and “Celebrity” for fake news detection.

Literature Survey

A sufficient number of works could be found in the literature in fake news detection. We could detect fake news at two levels, namely the *conceptual level* and *operational level* :

★ Thorne et al. (2018) (https://aclanthology.org/N18-1074/) introduced a novel dataset for fact-checking and verification where evidence is large Wikipedia corpus.

★ The work of Conroy et al. (2015) (https://dl.acm.org/doi/10.5555/2857070.2857152) fostered linguistics and fact checking based approaches to distinguish between real and fake news, which could be considered as the work at conceptual level.

★ The work of Rubin et al. (2015) (https://dl.acm.org/doi/10.5555/2857070.2857153) defined that conceptually there are three types of fake news:

- i) Serious Fabrications
- ii) Hoaxes and
- iii) Satire.

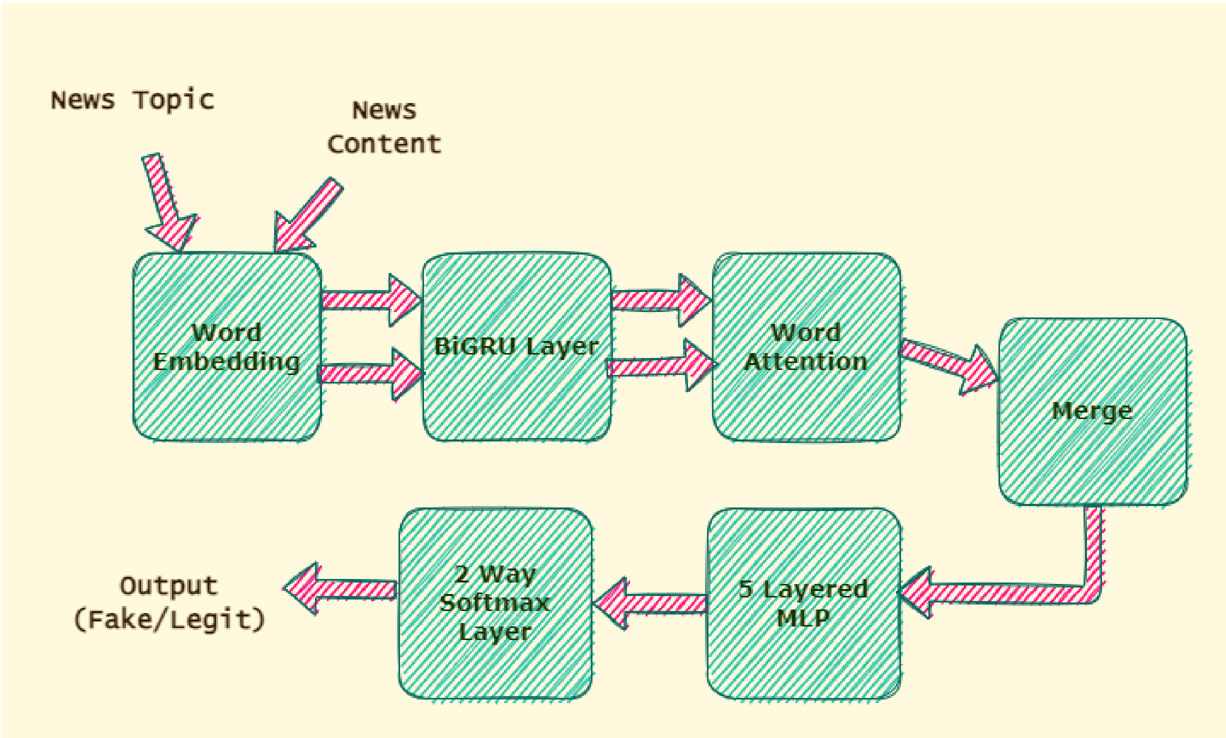
★ The Fake News Challenge 2 organized a competition to explore how Artificial Intelligence technologies could be fostered to combat fake news. Almost 50 participants participated and submitted their systems. Hanselowski et al. (2018) (https://aclanthology.org/C18-1158/) performed retrospective analysis of the three best participating systems of the Fake News Challenge.

★ The work of Saikh et al. (2019) (https://link.springer.com/chapter/10.1007%2F978-3-030-23281-8_30) detected fake news through stance detection and also correlated this stance classification problem with Textual Entailment (TE). They tackled this problem using statistical machine learning and deep learning approaches separately and with a combination of both of these. This system achieved state-of-the-art *results*.

Design and Implementation

Model 1:

Architectural Diagram for the First model



I. Pass title and content for **Word Embedding**. After Word Embedding of inputs the resultant vectors passed to BiGRU Encoding Layer.

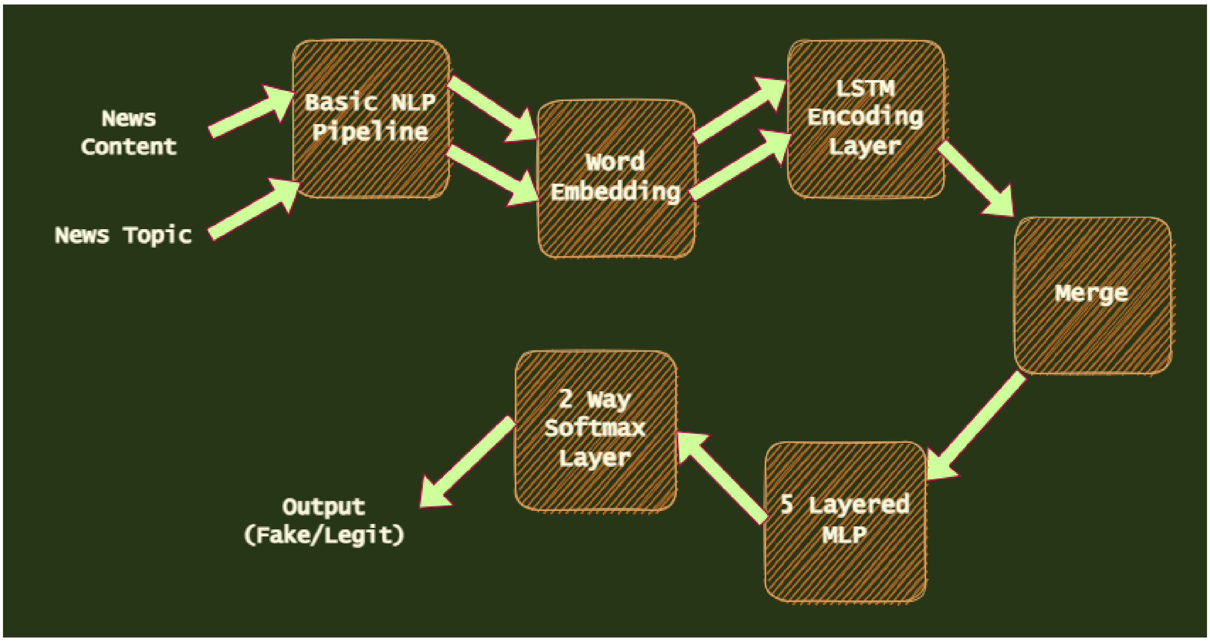
II. Bidirectional GRU's are a type of bidirectional recurrent neural networks with only the input and forget gates which allows the use of information from both previous time steps and later time steps to make predictions about the current state.

Modification

In the Modified Algorithm, We have used LSTM (Long short-term memory) as the Encoding layer instead of a BiGRU Model . LSTM is a type of recurrent neural network capable of learning order dependence in sequence prediction problems, which makes it a powerful tool in our scenario.

```
# Simplistic LSTM Architecture
model=Sequential()
model.add(Embedding(voc_size,embedding_vector_features,input_length=sent_length))
model.add(LSTM(100))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Architectural Diagram for the Modified model



III. Attention is one component of a network's architecture, and is in charge of managing and quantifying the interdependence

IV. Merge Input Content and Input Title from above layers into Nodes of MLP (Multilayer Layer Perceptron).

V. We use 512, 256, 128, 50 and 10 neurons, respectively, for five such layers of Perceptron with ReLU activation in each layer. Between each such layer, we employ 20% dropout as a measurement of regularization.

VI. Finally, the output from the last fully connected layer is fed into a final classification layer with softmax activation function having 2 neurons from which We obtain Classification Output-label as Fake or Legit.

Insights of Implementation

```
input_text = Input(shape=(SEQUENCE_LENGTH,vector_size,), name='input_text')

# Add Encoding Layer: BiGRU
x = Bidirectional(GRU(100, return_sequences=True))(input_text)

# Apply Attention Model at word level
x_attention = Hierarchical_Attention(100)(x)

# Build a 5 Layered MLP
z = Dense(units = 512, activation = 'relu')(x_attention)
z = Dropout(0.2)(z)
z = Dense(units = 256, activation = 'relu')(z)
z = Dropout(0.2)(z)
z = Dense(units = 128, activation = 'relu')(z)
z = Dropout(0.2)(z)
z = Dense(units = 50, activation = 'relu')(z)
z = Dropout(0.2)(z)
z = Dense(units = 10, activation = 'relu')(z)
z = Dropout(0.2)(z)

output = Dense(units = 2, activation = 'softmax')(z)

model = Model(inputs = [input_text], outputs = output)
model.compile(optimizer= 'adam', loss= 'categorical_crossentropy', metrics=['acc'])
history = model.fit(x_train,y_train, batch_size=16, epochs=10)

# Model has been trained!
```

Model 2:

Another Approach proposed by the Research paper was a model whose Embedding Layer is based on ELMo (Embedding for Language Model) and the MLP Network, which is the same as the one applied in Model 1.

ELMo Embedding:

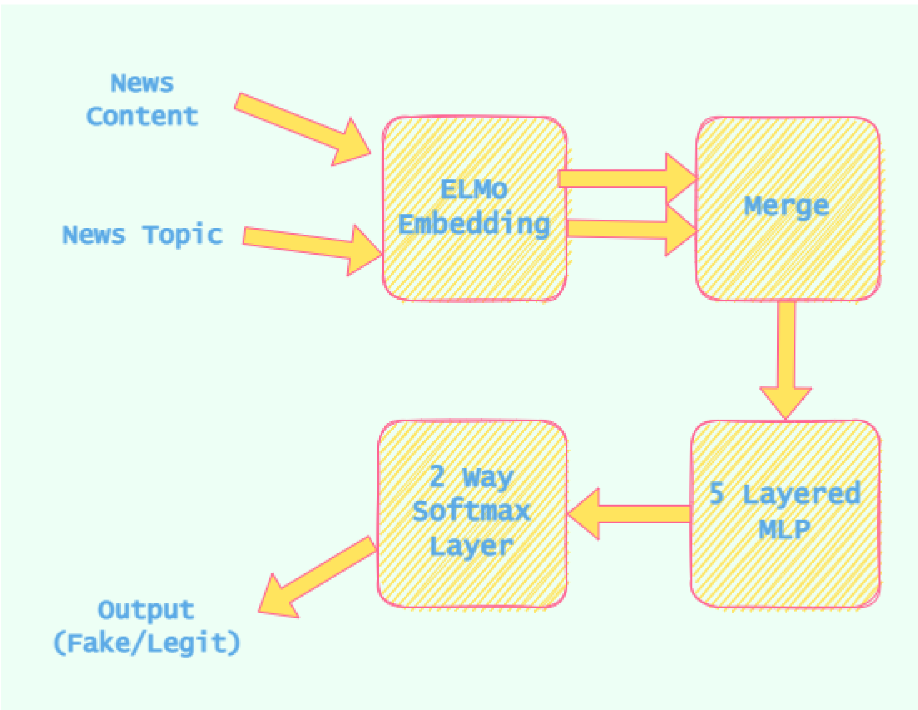
Embedding from Language Model (ELMo) has several advantages over other word vector methods (like Word Embedding in model 1), and found to be a good performer in many challenging NLP problems. It has key features like :

i) Contextual ii) Deep iii) Character based

Insights of Implementation

```python
embed = hub.Module("https://tfhub.dev/google/elmo/2", trainable=True)

# ELMo Embedding
def ELMoEmbedding(x):
    return embed(tf.squeeze(tf.cast(x, tf.string)), signature="default", as_dict=True)["default"]

input_text = Input(shape=(1,), dtype=tf.string)

# Add Embedding Layer (based on ELMo)
embedding = Lambda(ELMoEmbedding, output_shape=(1024, ))(input_text)

# Build a 5 Layered MLP
z = embedding
z = Dense(units = 512, activation = 'relu')(z)
z = Dropout(0.2)(z)
z = Dense(units = 256, activation = 'relu')(z)
z = Dropout(0.2)(z)
z = Dense(units = 128, activation = 'relu')(z)
z = Dropout(0.2)(z)
z = Dense(units = 50, activation = 'relu')(z)
z = Dropout(0.2)(z)
z = Dense(units = 10, activation = 'relu')(z)
z = Dropout(0.2)(z)

output = Dense(2, activation='softmax')(z)

model = Model(inputs=[input_text], outputs = output)
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit(x_train, y_train,batch_size=16, epochs=10)

# Model has been trained!
```

Architectural Diagram for the Second Model



Results and discussions

- Following is the analysis of Datasets being used for this Project and their composition:

Class Distribution and Word Statistics for Datasets.

Dataset	# of Examples	Average Words/sentence	Words	Label
FakeNewsAMT	240	132/5	31,990	Fake
	240	139/5	33,378	Legit
Celebrity	250	399/17	39,440	Fake
	250	700/33	70,975	Legit

- These are the results of Cross-Domain analysis (i.e. model trained on FakeNews AMT and tested on Celebrity and vice versa) to explore the applicability of our systems across the domains.

Cross-Domain Analysis on the Best Performing System

Training	Testing	Accuracy(%)
Celebrity	FakeNewsAMT	69.5
FakeNewsAMT	Celebrity	56.1

- Comparison between Test Accuracies of Proposed Models and Modified Algorithm Model

Classification Results for the Fake News AMT and Celebrity News Dataset with Two Proposed Methods and Comparison with Modified Model

Data Set	System	Model	Test Accuracy (%)
FakeNews AMT	Proposed	Model 1	77.08
		Model 2	83.3
	Modified Model	Bi LSTM Model	80.04
Celebrity	Proposed	Model 1	76..53
		Model 2	79
	Modified Model	Bi LSTM Model	82.67

- Testing and Comparing the different Models by Training on Multi-Domain data or Domain-wise data and testing on Domain-wise data

Result of Exp. A(Trained on Multi-domain Data and Tested on Domain wise Data) and Exp. B (Trained on Domain wise Data and Tested on Domain wise Data)

Domain	Exp. A			Exp. B		
	Model 1	Model 2	Mod. Model	Model 1	Model 2	Mod. Model
Business	74.75	78.75	78.08	63.56	68.56	62
Education	77.25	91.25	79.42	65.65	70.65	71.16
Technology	76.22	88.75	74.12	64.3	65.35	66.48
Politics	73.75	88.75	81.92	64.27	69.22	70.34
Entertainment	68.25	76.25	77.2	65.89	71.2	69.45
Sports	70.75	73.75	75	67.86	71.45	72.01

Conclusion and Future work

In this Paper, Two Deep Learning Algorithms were proposed to solve the problem of Fake News detection . The Research Paper compares these algorithms with the Existing Algorithms, where they give better results. We modified the proposed algorithm which gives higher accuracy on some of the Multidomain-News datasets when compared to the Proposed Models.

The ideas of Base Paper and Modified Algorithm can be considered and used for developing a fake content detection algorithm for Social Media, which is a huge requirement nowadays.

Use of transfer learning and injection external knowledge can be made for better understanding of Fake News Detection.

Handling of Named Entities efficiently and incorporating their embedding with the normal phrases can be targeted to improve the model's performance .

References.

Paper Link :

https://arxiv.org/abs/2005.04938

Official Dataset :

https://lit.eecs.umich.edu/downloads.html#Fake%20News

Note : If the Dataset Download fails on Chrome Browser, use Firefox Browser or below link.

Above Dataset uploaded on G-Drive :
https://drive.google.com/drive/folders/1v1LDcvGZhBV-Ffq1uPfeyAT2NVrtORFf?usp=sharing

Dataset for Multi-Domain Analysis :
https://drive.google.com/drive/folders/1bCq3cV5woVcvSBcOzlMUEXlRW4AqpAp9

Reference Links explored:

- https://tfhub.dev/google/elmo/3
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU