Comparative Analysis of Various Algorithms for Fake News Detection

By **Prithwiraj Samanta, Satya Kumari**

Under the guidance of **Dr. Rashmi Panda**

Introduction

- Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article.
- In this work, we have created an analysis report of various algorithms (particularly LSTM + CNN + Attention + Transformer Models) for automated classification of news articles.
- Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various methods and evaluate their performance on real world datasets.

Introduction

- RNN using LSTM units partially solve the vanishing gradient problem, because LSTM units allow gradients to also flow unchanged. LSTM networks can still suffer from the exploding gradient problem. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

- CNNs are regularized versions of multilayer perceptron. Multilayer perceptron usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data.

- Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. CNN LSTMs were developed for visual time series prediction problems and the application of generating textual descriptions from sequences of images or sequence of text document.

- Attention is a mechanism combined in the RNN allowing it to focus on certain parts of the input sequence when predicting a certain part of the output sequence, enabling easier learning and of higher quality. Combination of attention mechanisms enabled improved performance in many tasks making it an integral part of modern RNN networks.

- The development of the Transformer architecture revealed that attention mechanisms were powerful in themselves, and that sequential recurrent processing of data was not necessary to achieve the performance gains of RNNs with attention Transformers use an attention mechanism without an RNN, processing all tokens at the same time and calculating attention weights between them in successive layers.

Literature Survey

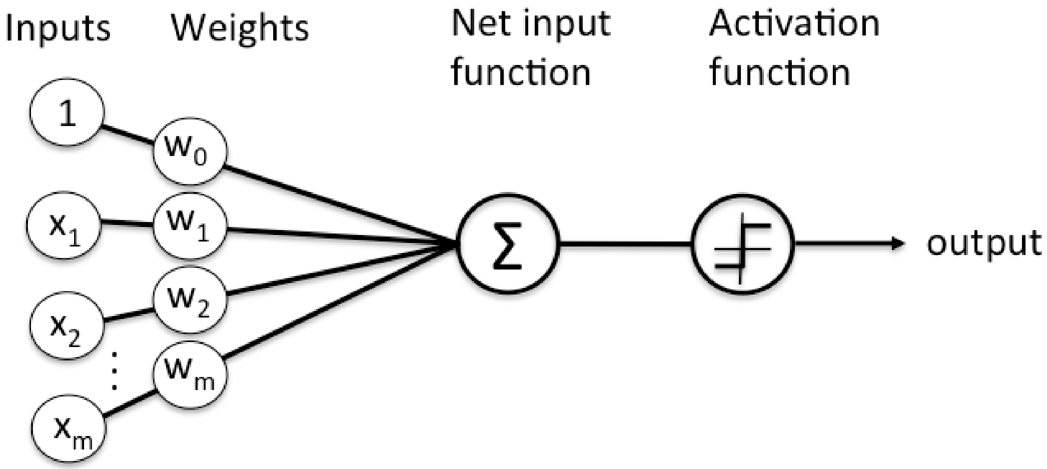
[A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities] This paper is a perfect way to dive into the vast spectrum of Fake News Detection. Starting from the introduction and definition of fake news detection, it takes a very good look to various methods implemented to identify fake news and prevent it from releasing publicly. It talks about different machine learning traditional models used and ongoing deep learning models being used to improve the accuracy of identifying fake news.[Fake News Detection Using Machine Learning approaches: A systematic Review] tells us about accuracy of different models i.e., Naive Bayes, Decision trees, SVM, Neural Networks, Random Forest, XG Boost on different datasets. [Survey on Fake News Detection using Machine learning Algorithms] published at (IJERT) in 2021 shows that Random Forest yielded 65.6 accuracy on liar dataset [liar dataset] preceding naive bayes, svm, logistic regression and decision tree with 63.7, 63, 62.5, 60 accuracies.[Enquiring Minds: Early Detection of Rumours in Social Media from Enquiry Posts] by Xuzhou in 2015 proposed a rumour detector which identifies trending rumours on twitter. The detector, which searches for rare but informative phrases, combined with clustering and a classifier on the clusters, yields surprisingly good performance. According to this detector, on twitter out of 50 candidate statements, about one third of them are real rumours.[Development of Fake News Model Using Machine Learning through Natural Language Processing] published by Ahmed in International Journal of Computer and Information Engineering in 2020 gives us all the relevant information regarding implementation of machine learning on fake news. It gives us an overview of Methodology, pre-processing and implementation tasks of a model. The passive aggressive classifier gives 0.93 accuracy on fake news dataset which is the maximum of all the classifiers used[Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)] published in 2020 at IEEE proposed a hybrid deep learning model (a combination of CNN and LSTM) which experiments with dimension reduction techniques and pre-processing. The dimensionality reduction methods it uses are principal component analysis [2018 on using Principal Component Analysis] and Chi-square [chi square reduction] on fake news challenge dataset. The best accuracy is yielded by CNN-LSTM with PCA, that is 97.8%. [Deep Two-path Semi-supervised Learning for Fake News Detection] this study shows an implementation of deep two-path semi-supervised learning model "dstl" on PHEME dataset. To train and test the model both labelled and unlabelled dataset is used. The model contains three CNNs. The performance of dstl is inspected with different ratios of labelled data. The proposed model surpasses the F-score of bidirectional recurrent neural (BRNN), 35.85%, by yielding F-Score of 57.98% with 30% labelled data.[A Novel Hybrid Deep Learning Model for Sentiment Classification] published in 2020 proposes a tree structure model having two branches with the idea of using CNN and LSTM parallel. As we know CNN is good at extracting spatial features and LSTM is good at finding long-term dependencies in data. Further it suggests to concatenate both output vector and implement SoftMax layer. The input data of the branches differ due to pre-processing methods and corpus representation. CNN yields best accuracy when used with character level embedding where content like URL information, emoji, stop words are also taken into account. RNN variants like LSTM, Bi-LSTM, GRU input data is ready after pre-processing and applying word embedding methods like FastText, which can embed the words successfully which are not present in the corpus. It yields an F1-score of 0.89 on self-mined dataset which consists of 17,289 Turkish tweets.[A Closer Look at Fake News Detection: A Deep Learning Perspective] The paper uses Fake news challenge dataset which has 75000 instances is divided into train and validation data. The baseline models used are CNN models and BERT. The proposed model uses attention layer with CNN and RNN. To improve the accuracy dropout layer and max pooling are used. The best accuracy it achieved is 71.21% on competition test set. [Indonesia's Fake News Detection Using Transformer Network] The dataset used is a combination of three datasets which are WILD dataset, LIAR dataset and a dataset taken from one of the Kaggle competition. Models like fine-tuned BERT, CNN-LSTM, CNN are used with embedding layer. The best accuracy of 90% is achieved by BERT.[Fake News Identification on Twitter with Hybrid CNN and RNN Models] Recurrent Neural Network is proficient at detecting pattern on Sequential data. This paper experiments with various models containing RNN layer. The dataset used consists of 5800 tweets used in the work of Zubiaga et al[]. The model with LSTM layer performs the best with the accuracy of 82.29%.

Motivation

With the advancement of technology, digital news is more widely exposed to users globally and contributes to the increment of spreading hoaxes and disinformation online. Fake news can be found through popular platforms such as social media and the Internet. There have been multiple solutions and efforts in the detection of fake news where it even works with artificial intelligence tools. However, fake news intends to convince the reader to believe false information which deems these articles difficult to perceive. The rate of producing digital news is large and quick, running daily at every second, thus it is challenging for machine learning to effectively detect fake news.

In the discourse of not being able to detect fake news, the world would no longer hold value in truth. Fake news paves the way for deceiving others and promoting ideologies. These people who produce the wrong information benefit by earning money with the number of interactions on their publications. Spreading disinformation holds various intentions, in particular, to gain favour in political elections, for business and products, done out of spite or revenge. Humans can be gullible and fake news is challenging to differentiate from the normal news. Most are easily influenced especially by the sharing of friends and family due to relations and trust. We tend to base our emotions from the news, which makes accepting not difficult when it is relevant and stance from our own beliefs. Therefore, we become satisfied with what we want to hear and fall into these traps.

Implementation



- **Perceptron** is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

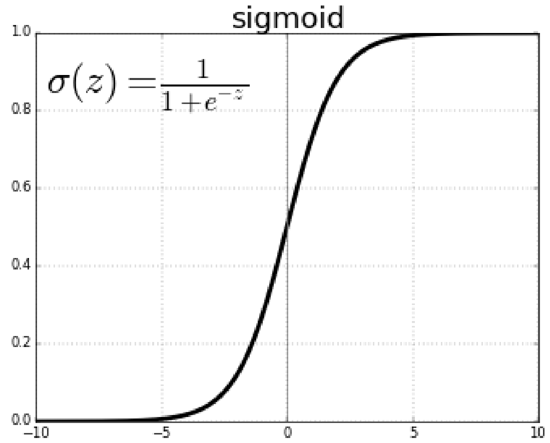
- **Algorithm**

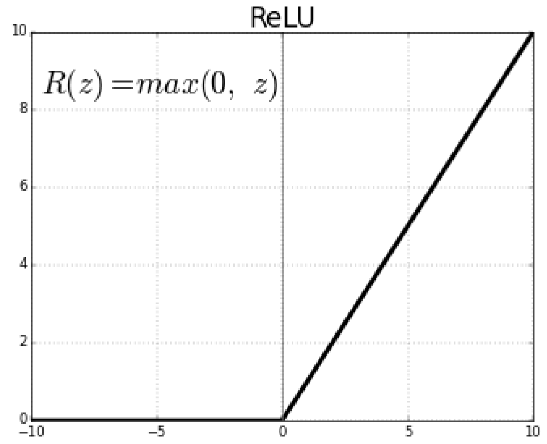
1)For every input, multiply that input by its weight.

2)Sum all of the weighted inputs.

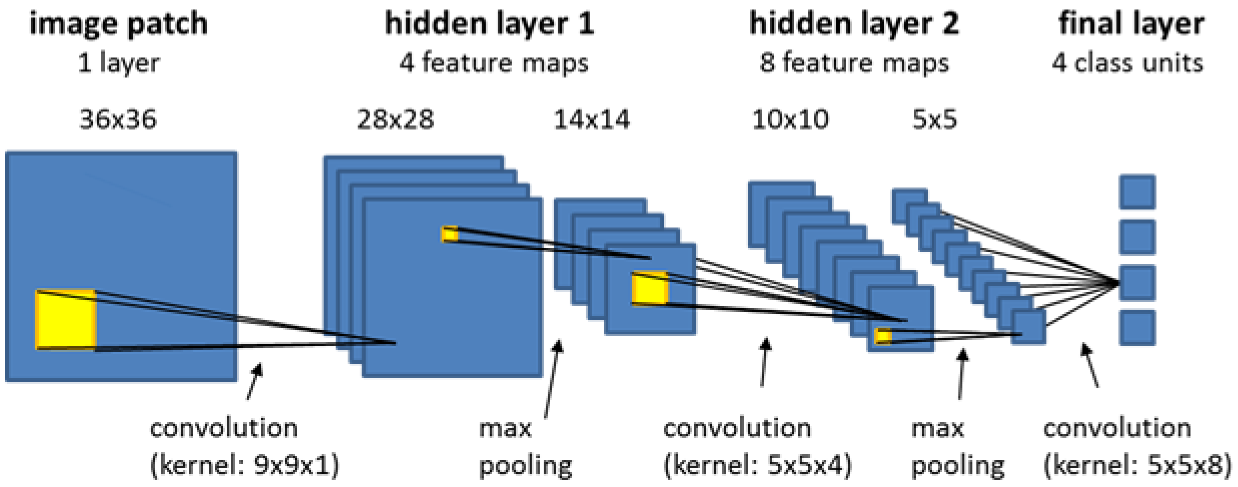
3)Compute the output of the perceptron based on that sum passed through an activation function (the sign of the sum).

- **Activation** function also known as transfer function. of a node defines the output of that node given an input or set of inputs. A standard integrated circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on input.

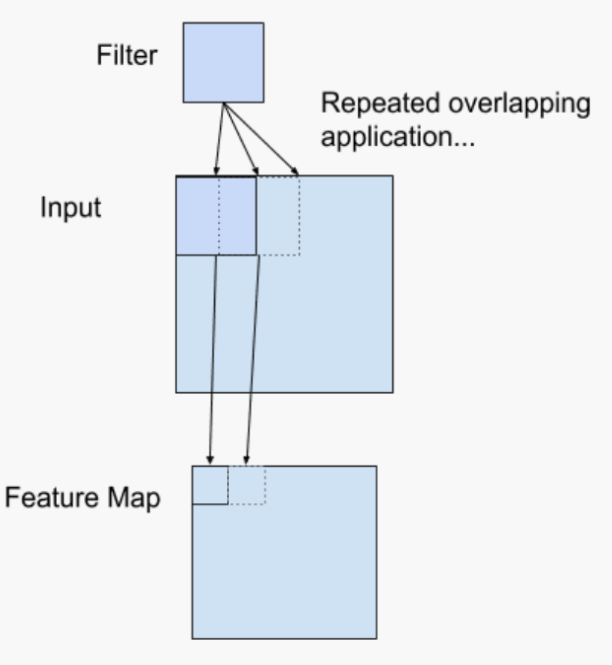




Implementation



- Convolutional Neural Network, CNNs are regularized versions of multilayer perceptron. Multilayer perceptron usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme.
- It mainly consist of two types of layers – Convolution Layer and Max Pooling Layer.
- In Convolution layer, filters are analogous to weights in multi-layer perceptron. The multiplication is performed between an array of input data and a two-dimensional array of weights, called a filter or a kernel. Using a filter smaller than the input is intentional as it allows the same filter (set of weights) to be multiplied by the input array multiple times at different points on the input. Specifically, the filter is applied systematically to each overlapping part or filter-sized patch of the input data, left to right, top to bottom.
- If the filter is designed to detect a specific type of feature in the input, then the application of that filter systematically across the entire input image allows the filter an opportunity to discover that feature anywhere in the image. This capability is commonly referred to as translation invariance, e.g. the general interest in whether the feature is present rather than where it was present.
- As the filter is applied multiple times to the input array, the result is a two-dimensional array of output values that represent a filtering of the input. As such, the two-dimensional output array from this operation is called a “feature map“. Convolutional neural networks do not learn a single filter; they, in fact, learn multiple features in parallel for a given input. For example, it is common for a convolutional layer to learn from 32 to 512 filters in parallel for a given input. This gives the model 32, or even 512, different ways of extracting features from an input, or many different ways of both “learning to see” and after training, many different ways of “seeing” the input data.
- A filter must always have the same number of channels as the input, often referred to as “depth“. Color images have multiple channels, typically one for each color channel, such as red, green, and blue.



Implementation

- A limitation of the feature map output of convolutional layers is that they record the precise position of features in the input. This means that small movements in the position of the feature in the input image will result in a different feature map. This can happen with re-cropping, rotation, shifting, and other minor changes to the input image. A pooling layer is a new layer added after the convolutional layer. Specifically, after a nonlinearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer

- Pooling involves selecting a pooling operation, much like a filter to be applied to feature maps. The size of the pooling operation or filter is smaller than the size of the feature map; specifically, it is almost always 2×2 pixels applied with a stride of 2 pixels.

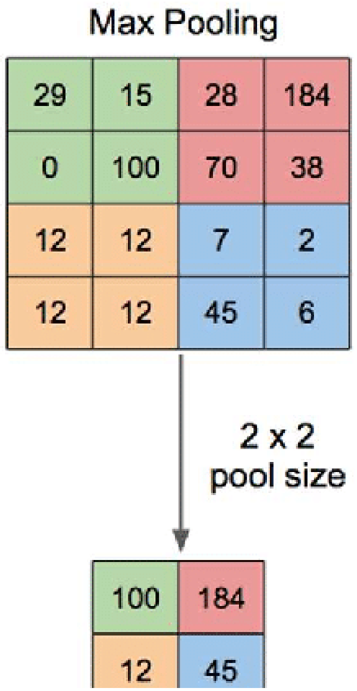
- This means that the pooling layer will always reduce the size of each feature map by a factor of 2, e.g. each dimension is halved, reducing the number of pixels or values in each feature map to one quarter the size. For example, a pooling layer applied to a feature map of 6×6 (36 pixels) will result in an output pooled feature map of 3×3 (9 pixels).

- The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location. This capability added by pooling is called the model's invariance to local translation.

- There are two types of pooling layers – Max Pooling and Average Pooling.

- **Overfitting** happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model.

- **Underfitting** refers to a model that can neither model the training data nor generalize to new data. An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.

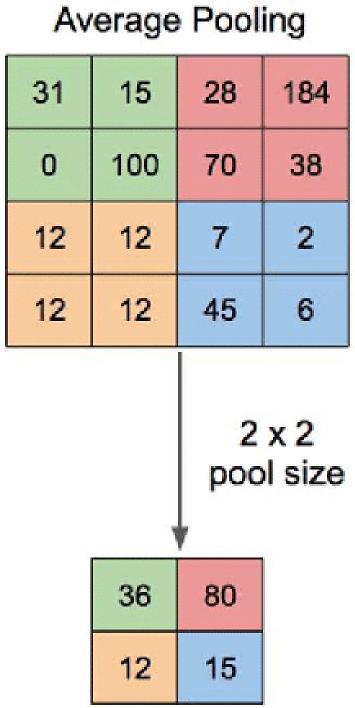


Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2 pool size

100	184
12	45



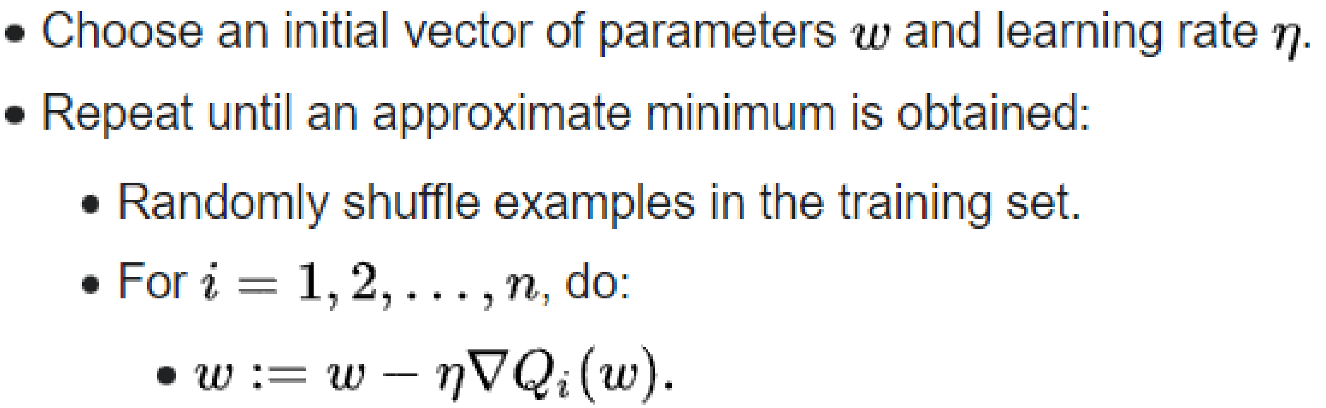
Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2 pool size

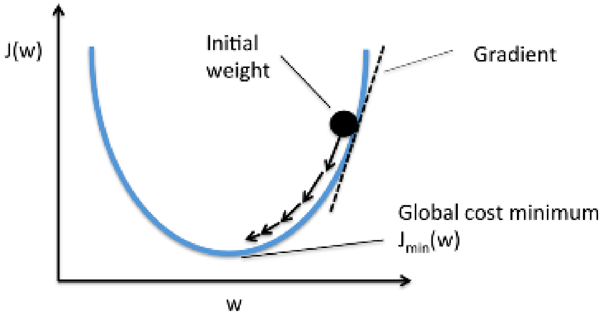
36	80
12	15

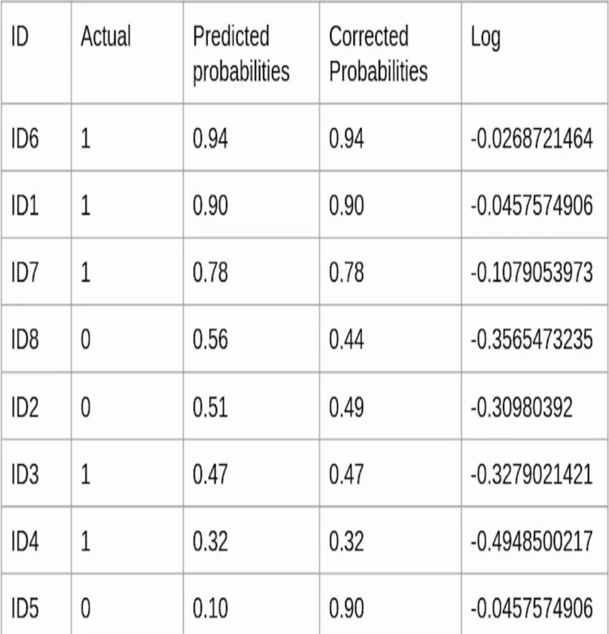
Implementation



- Choose an initial vector of parameters w and learning rate η.
- Repeat until an approximate minimum is obtained:
  - Randomly shuffle examples in the training set.
  - For $i = 1, 2, \ldots, n$, do:
    - $w := w - \eta \nabla Q_i(w).$

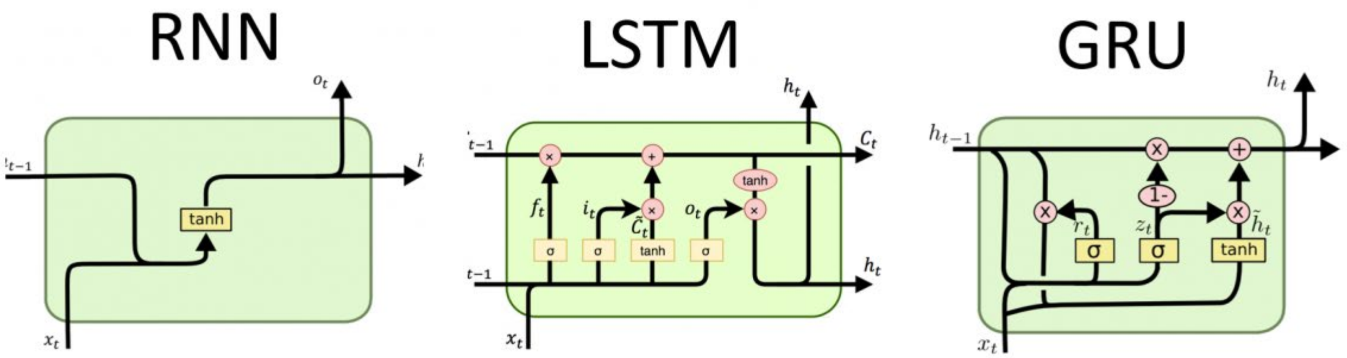
- Optimizers are used to update weights and biases i.e. the internal parameters of a model to reduce the error.
- In most learning networks, error is calculated as the difference between the actual output y and the predicted output y' . The function that is used to compute this error is known as Loss Function also known as Cost function
- Entropy is a measure of the uncertainty associated with a given distribution q(y). Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value.
- **Binary Cross Entropy is the negative average of the log of corrected predicted probabilities**.
- Adam Optimizer, calculates an **exponential moving average of the gradient** and **the squared gradient**, and the parameters beta1 and beta2 control the decay rates of these moving averages.
- These calculated parameters along with learning rate determines the change in weight that will lead towards minima of loss function.
- As the backpropagation algorithm advances downwards(or backward) from the output layer towards the input layer, the gradients often get smaller and smaller and approach zero which eventually leaves the weights of the initial or lower layers nearly unchanged. As a result, the gradient descent never converges to the optimum. This is known as the vanishing gradients problem.
- On the contrary, in some cases, the gradients keep on getting larger and larger as the backpropagation algorithm progresses. This, in turn, causes very large weight updates and causes the gradient descent to diverge. This is known as the exploding gradients problem.
- In a neural network, **Batch Normalization** is achieved through a normalization step that fixes the means and variances of each layer's inputs.



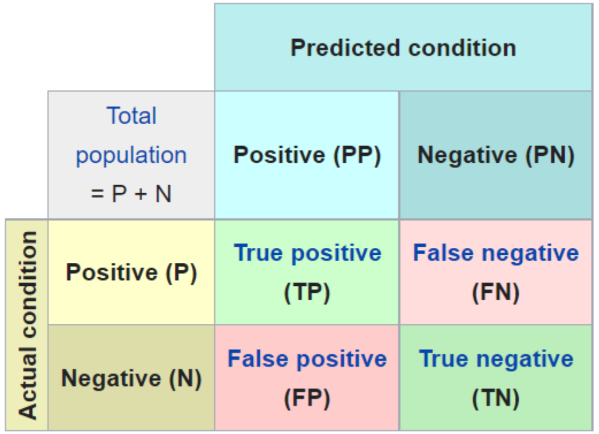


ID	Actual	Predicted probabilities	Corrected Probabilities	Log
ID6	1	0.94	0.94	-0.0268721464
ID1	1	0.90	0.90	-0.0457574906
ID7	1	0.78	0.78	-0.1079053973
ID8	0	0.56	0.44	-0.3565473235
ID2	0	0.51	0.49	-0.30980392
ID3	1	0.47	0.47	-0.3279021421
ID4	1	0.32	0.32	-0.4948500217
ID5	0	0.10	0.90	-0.0457574906

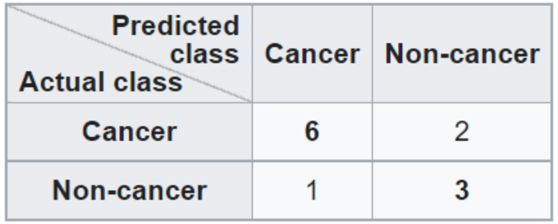
Implementation



- Optimizers are used
- A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed or undirected graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. Recurrent neural networks are theoretically Turing complete and can run arbitrary programs to process arbitrary sequences of inputs.
- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.
- LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

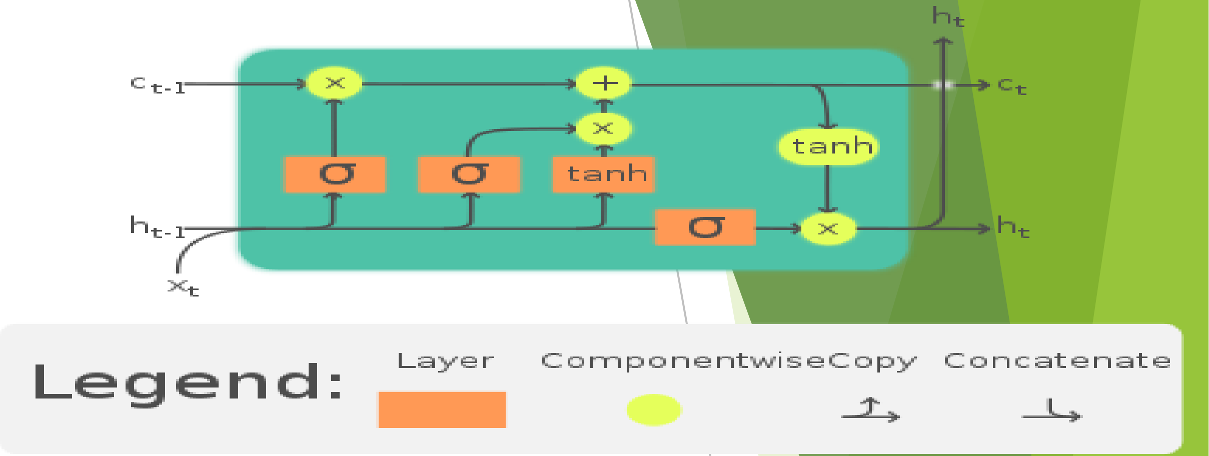


		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)



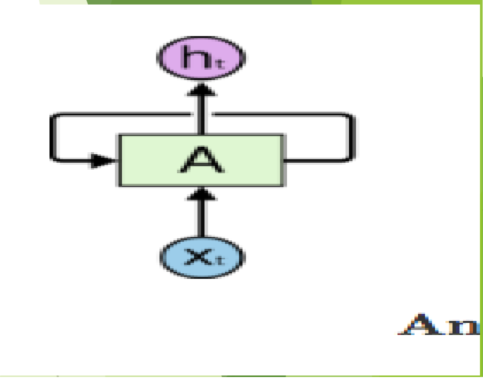
Predicted class Actual class	Cancer	Non-cancer
Cancer	6	2
Non-cancer	1	3

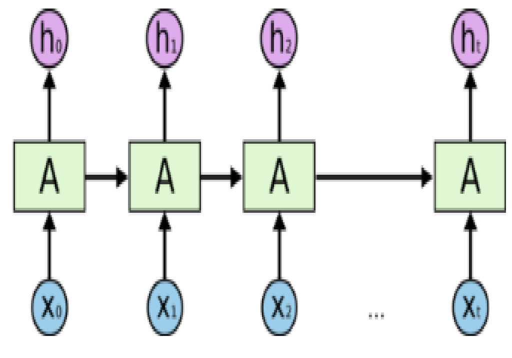
Implementation



- *An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. These blocks can be thought of as a differentiable version of the memory chips in a digital computer. Each one contains one or more recurrently connected memory cells and three multiplicative units – the input, output and forget gates – that provide continuous analogues of write, read and reset operations for the cells. The net can only interact with the cells via the gates.*

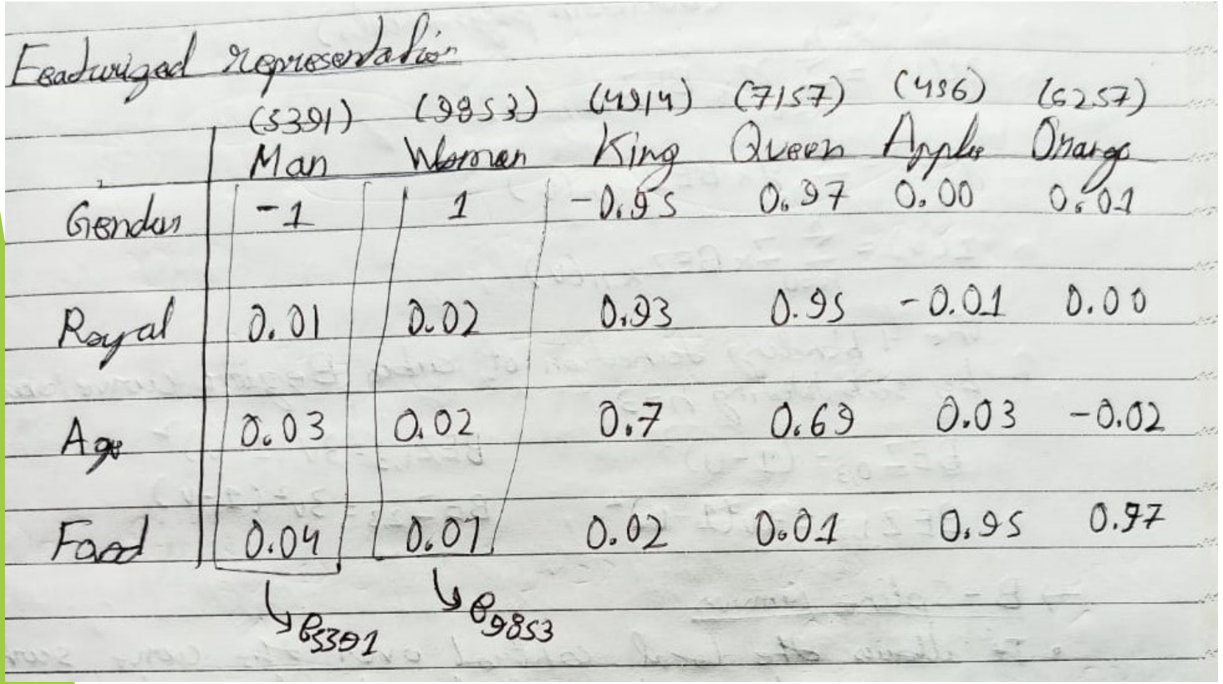
- *Each memory cell's internal architecture guarantees constant error flow within its constant error carrousel CEC… This represents the basis for bridging very long time lags. Two gate units learn to open and close access to error flow within each memory cell's CEC. The multiplicative input gate affords protection of the CEC from perturbation by irrelevant inputs. Likewise, the multiplicative output gate protects other units from perturbation by currently irrelevant memory contents.*





ed recurrent neural network.

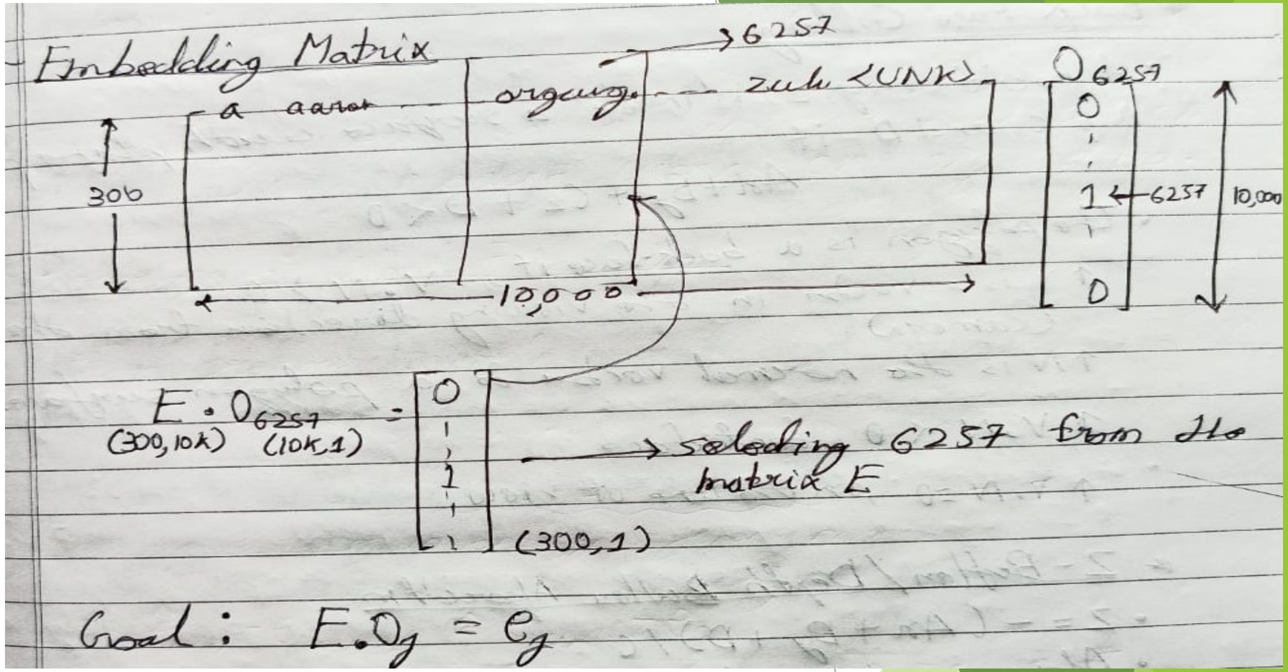
Implementation



Featurized representation

	(5391) Man	(9853) Woman	(4914) King	(7157) Queen	(456) Apple	(6257) Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.07	0.02	0.01	0.95	0.97

↳ e_{5391} ↳ e_{9853}

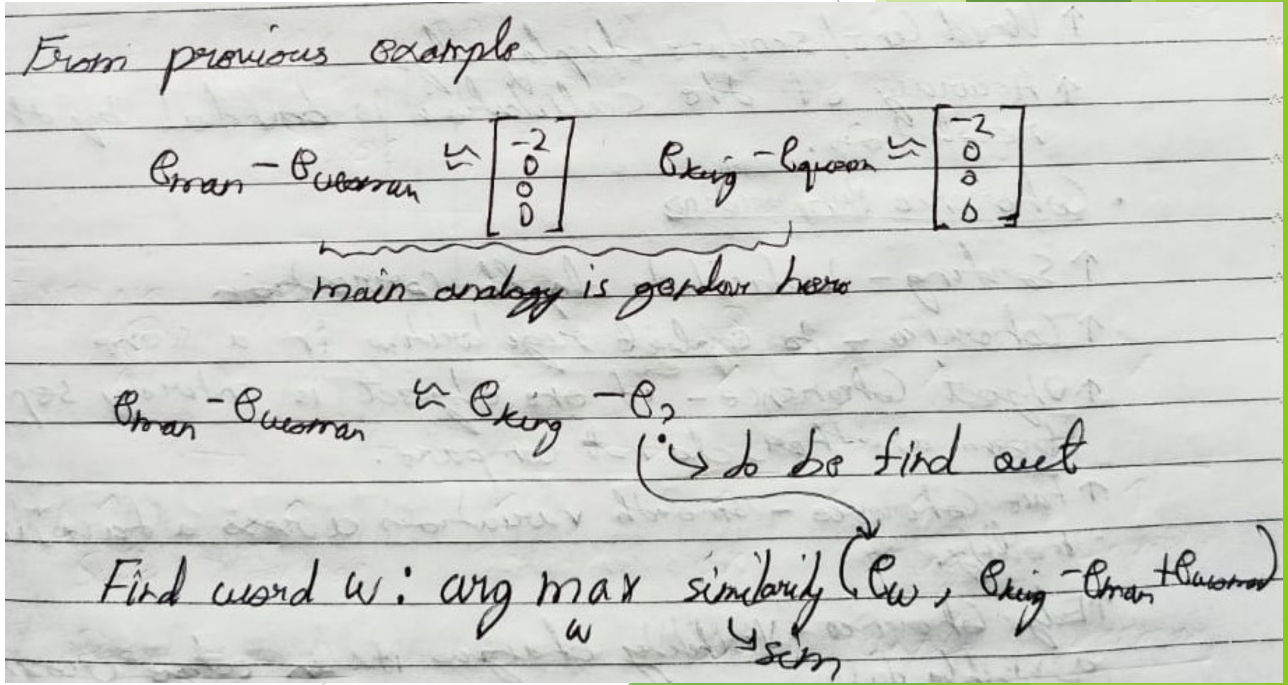


Embedding Matrix

$$\underset{(300,10k)}{E} \cdot \underset{(10k,1)}{O_{6257}} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} \quad (300,1)$$

→ selecting 6257 from the matrix E

Goal : $E \cdot O_j = e_j$



From previous example

$$\underbrace{e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad e_{king} - e_{queen}}_{\text{main analogy is gender here}} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

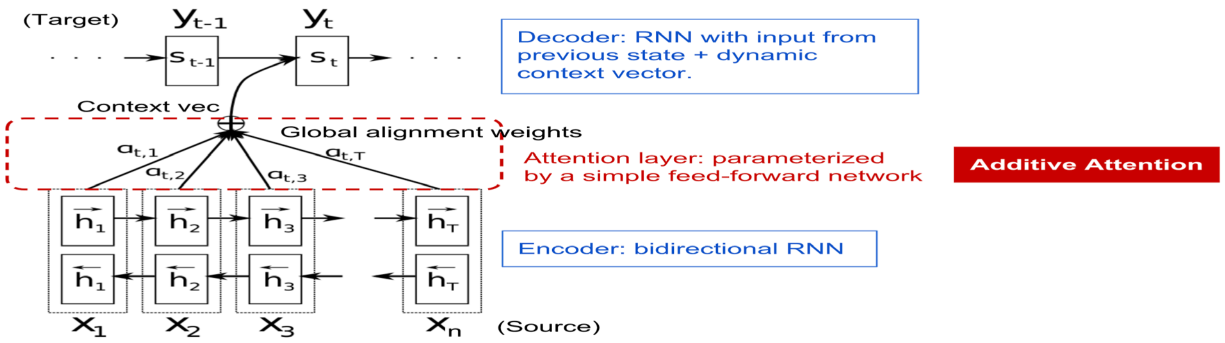
$e_{man} - e_{woman} \approx e_{king} - e_?$

↳ to be find out

Find word w : $\arg\max_{w}$ similarity $(e_w, e_{king} - e_{man} + e_{woman})$

↳ sim

Implementation



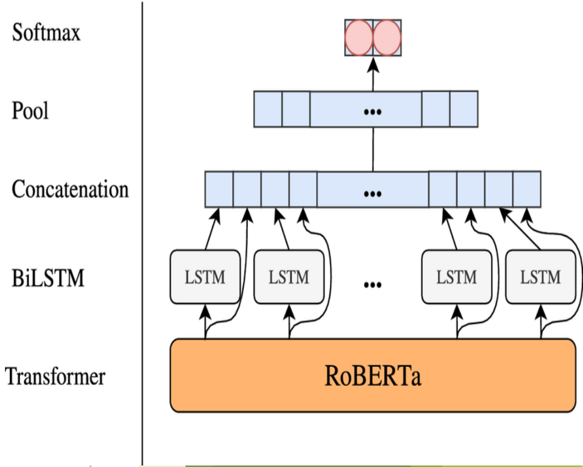
Decoder: RNN with input from previous state + dynamic context vector.

Attention layer: parameterized by a simple feed-forward network

Additive Attention

Encoder: bidirectional RNN

- The Attention mechanism was introduced to improve the performance of the encoder decoder model for Machine Translation. It was introduced by Bahdanau et al to address the bottleneck problem that arises with the use of a fixed-length encoding vector where the decoding would have limited access to the information provided by input. Attention is the ability to dynamically highlight and use the salient parts of the information at hand.
- The general Attention mechanism makes us of three main components namely the queries Q, the keys k, the values V.
- Each query vector is matched against a database of keys to compute a score value.This matching operation is computed as the dot product of the specific query under consideration with each key vector.
- The scores are passed through a Softmax operation to generate the weights. The generalized attention is then computed by a weighted sum of the value vectors where each value vector is paired with a corresponding key.
- Within the context of machine translation, each word in an input sentence would be attributed its own query, key and value vectors. These vectors are generated by multiplying the encoder’s representation of the specific word under consideration, with three different weight matrices that would have been generated during training.



Result and Analysis

- A robustly optimized method for pretraining natural language processing (NLP) systems that improves on Bidirectional Encoder Representations from Transformers, or BERT, the self-supervised method released by Google in 2018. BERT is a revolutionary technique that achieved state-of-the-art results on a range of NLP tasks while relying on unannotated text drawn from the web, as opposed to a language corpus that’s been labeled specifically for a given task.

- Specifically, RoBERTa is trained with dynamic masking, full-sentences without NSP loss, large mini-batches and a larger byte-level. Additionally, the two other important factors that have been under-emphasized in previous work: (1) the data used for pretraining, and (2) the number of training passes through the data. RoBERTa builds on BERT’s language masking strategy, wherein the system learns to predict intentionally hidden sections of text within otherwise unannotated language examples. RoBERTa, which was implemented in PyTorch, modifies key hyperparameters in BERT, including removing BERT’s next-sentence pretraining objective, and training with much larger mini-batches and learning rates. This allows RoBERTa to improve on the masked language modeling objective compared with BERT and leads to better downstream task performance. We also explore training RoBERTa on an order of magnitude more data than BERT, for a longer amount of time. We used existing unannotated NLP datasets as well as CC-News, a novel set drawn from public news articles.The transformer model is pretrained for 100K steps over a comparable BOOKCORPUS plus WIKIPEDIA dataset.

Conclusion

- In our study we started with simple LSTM models and tested their performance which was proportional to the number of features and the size of network. But still it produced maximum accuracy up to 89%.
- We observed that LSTM + CNN models are capable of producing accuracy up to 90% with much smaller network compared to simple LSTM model. But there was a problem of overfitting indicated of accuracy of training set(99%).
- We used RNN + Attention model which has accuracy about 95%. Then while studying attention model further we found that attention model is itself sufficient to give the result.
- Then we switched to transformer models which has accuracy up to 99%. In our study we have able to find out how gradually we progressed from LSTM models to transformer models.
- Our study will help the future researcher to understand the how these models are derived from its predecessor models and what improved its performance from its predecessor.

Future Work

- We are planning to work on detecting fake news shown in form of videos. We will be using our knowledge of text-based fake news detection and speech to text conversion.
- We will convert the speech in video into text, and then try to predict whether the news is fake or not. We may develop algorithm to identify the fake speaker and then warn the users against him/her.
- We have not yet created a data pipeline for our models. Our next work will contain data pipeline to automate the entire process of fetching data and converting it to required form.
- We will need to create the dataset as we have not found a suitable dataset for detecting fake news shown in form of videos.

References

- Xinyi Zhou and Reza Zafarani. 2020. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. ACM Comput. Surv. 53, 5, Article 109 (September 2020). https://doi.org/10.1145/3395046
- Ayat Abedalla, Aisha Al-Sadi, and Malak Abdullah. 2019. A Closer Look at Fake News Detection: A Deep Learning Perspective. ICAAI Proceedings of the 2019 3rd International Conference in Artificial Intelligence (October 2019). https://doi.org/10.1145/3369114.3369149
- Jibran Fawaid, Asiyah Awalina, and Rifky Yunus Krishnabayu . 2021. Indonesia’s Fake News Detection using Transformer Network. SIET. 6th International Conference on Sustainable Information Engineering and Technology (September 2021) https://doi.org/10.1145/3479645.3479666
- Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. 2018. Fake News Identification on Twitter with Hybrid CNN and RNN Models. SMSociety. 9th International Conference on Social Media and Society. (July 2018). https:// doi.org/10.1145/3217804.3217917
- S. I. Manzoor, J. Singla and Nikita, "Fake News Detection Using Machine Learning approaches: A systematic Review," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 230-234, https://doi.org/10.1109/ICOEI.2019.8862770.
- Zhao, Zhe & Resnick, Paul & Mei, Qiaozhu. (2015). Enquiring Minds: Early Detection of Rumours in social media from Enquiry Posts. 1395-1405. https://doi.org/10.1145/2736277.2741637.
- Sa, Ahmed & hinkelmann, knut & Corradini, Flavio. (2020). Development of Fake News Model using Machine Learning through Natural Language Processing.
- M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi and B. -W. On, "Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)," in IEEE Access, vol. 8, pp. 156695-156706, 2020, https://doi.org/10.1109/ACCESS.2020.3019735.
- arXiv:1906.05659 [cs.CL]
- M. U. Salur and I. Aydin, "A Novel Hybrid Deep Learning Model for Sentiment Classification," in IEEE Access, vol. 8, pp. 58080-58093, 2020, https://doi.org/10.1109/ACCESS.2020.2982538.