Fake News Detection Android APP

Presented By:

Himanshu Patel , Akarsh Mathur, Harshit Jain and Kunal Sahu

0126IT181027, 0126IT181003, 0126IT181026, 0126IT181035

INFORMATION TECHNOLOGY 7TH SEM “A”

INTRODUCTION

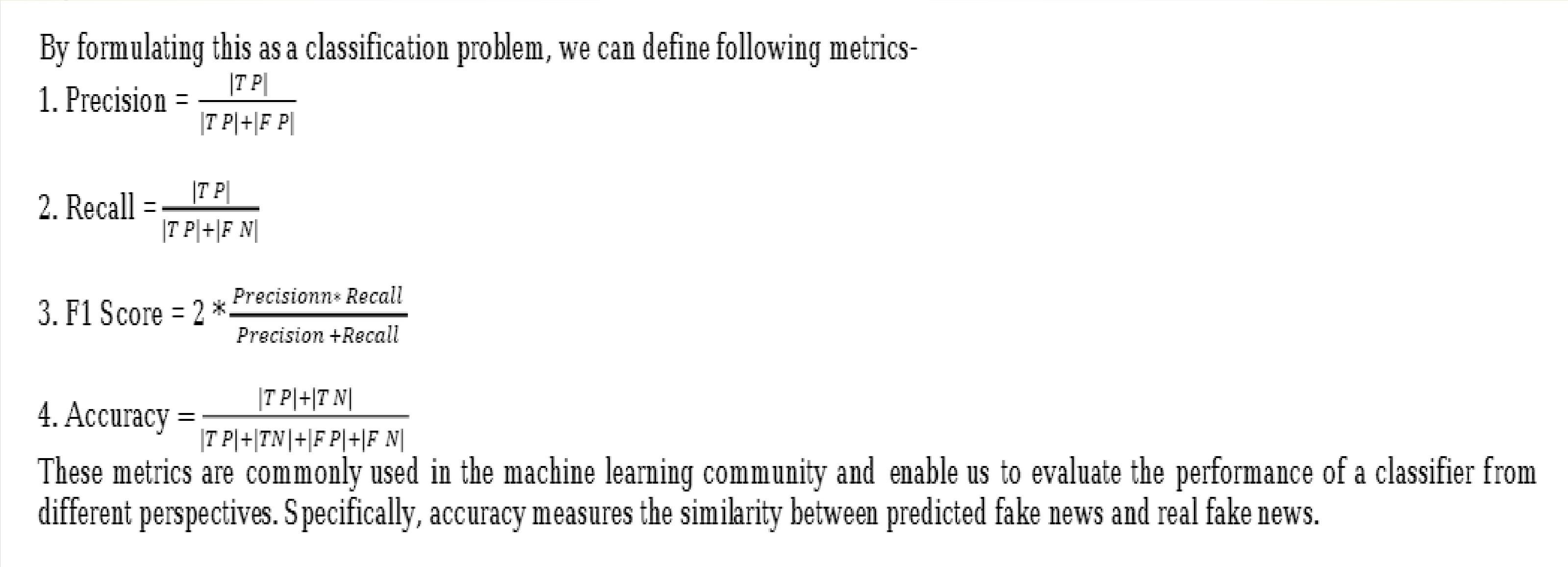
Fake news is false or misleading information presented as news. It often has the aim of damaging the reputation of a person or entity, or making money through advertising revenue.

ABSTRACT

- In our modern era where the internet is ubiquitous, everyone relies on various online resources for news.
- Along with the increase in the use of social media platforms like Facebook, Twitter, etc. news spread rapidly among millions of users within a very short span of time.
- The spread of fake news has far-reaching consequences like the creation of biased opinions to swaying election outcomes for the benefit of certain candidates.
- Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits.
- In this ppt, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning.
- We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news.

Objective

- The project mainly deals analysis of news from various news websites.
- Our Main objective is to get the overall correctness of news for a particular topic based on its correctness probability.



By formulating this as a classification problem, we can define following metrics-

1. Precision = $\frac{|TP|}{|TP|+|FP|}$

2. Recall = $\frac{|TP|}{|TP|+|FN|}$

3. F1 Score = $2 * \frac{Precisionn * Recall}{Precision + Recall}$

4. Accuracy = $\frac{|TP|+|TN|}{|TP|+|TN|+|FP|+|FN|}$

These metrics are commonly used in the machine learning community and enable us to evaluate the performance of a classifier from different perspectives. Specifically, accuracy measures the similarity between predicted fake news and real fake news.

GOALS

In FAKE DETECTOR, the fake news detection problem is formulated as a **credibility score inference problem**, and FAKEDETECTOR aims at learning a prediction model to infer the credibility labels of news articles, creators and subjects simultaneously.

Social media and news outlets publish fake news to increase readership or as part of psychological warfare.

In general, the goal is profiting through clickbaits. Clickbaits lure users and entice curiosity with flashy headlines or designs to click links to increase advertisements revenues.

METHODOLOGY

- This ppt explains the system which is developed in three parts. The first part is static which works on machine learning classifier.
- We studied and trained the model with 4 different classifiers and chose the best classifier for final execution.
- The second part is dynamic which takes the keyword/text from user and searches online for the truth probability of the news.
- The third part provides the authenticity of the text input by user.we have used Python and its tflite libraries.
- Python has a huge set of libraries and extensions, which can be easily used in Machine Learning.
- tflite Learn library is the best source for machine learning algorithms where nearly all types of machine learning algorithms are readily available for Python, thus easy and quick evaluation of ML algorithms is possible.

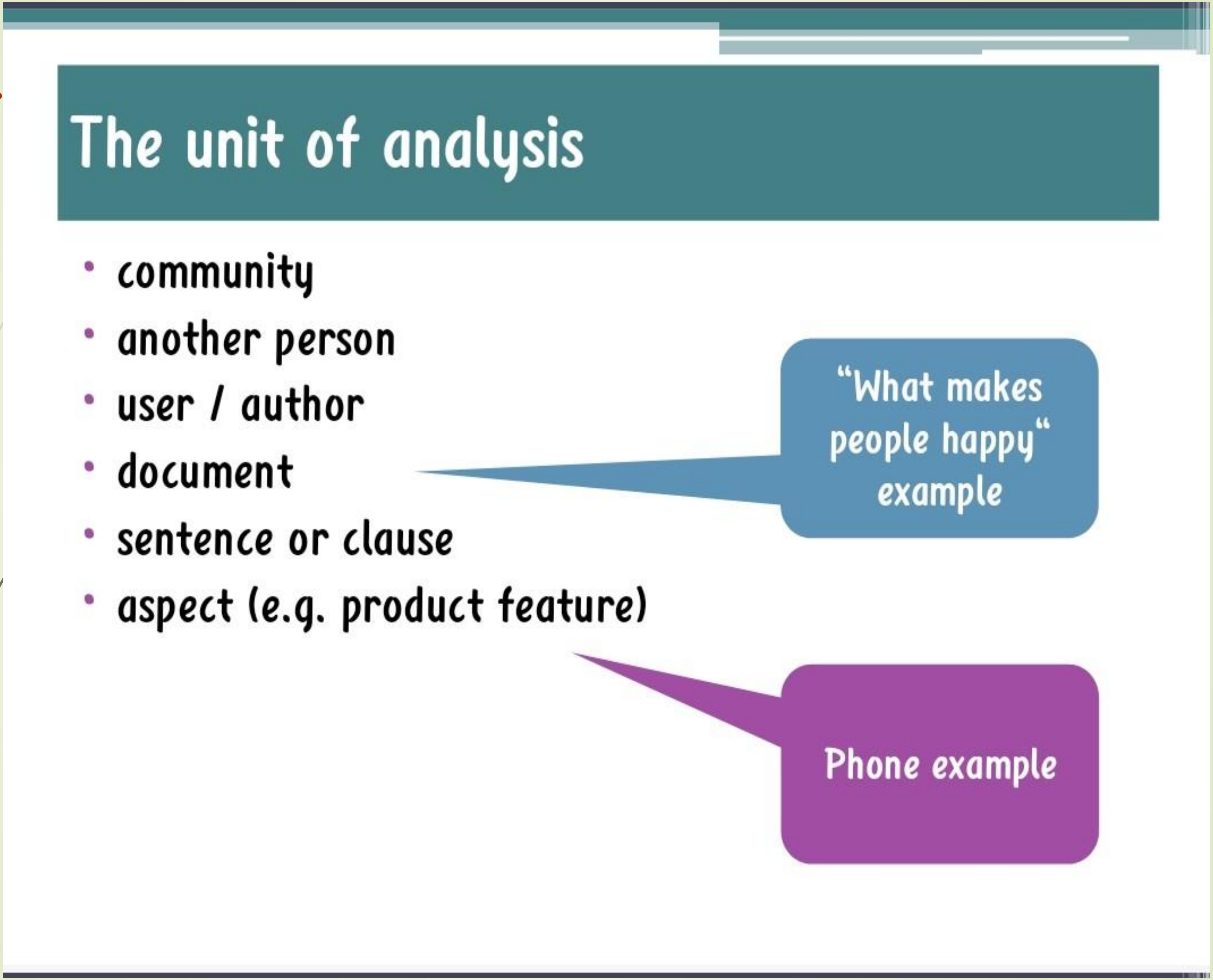
DATA Collection and Analysis

We can get online news from different sources like social media websites, search engine, homepage of news agency websites or the fact-checking websites. On the Internet, there are a few publicly available datasets for Fake news classification like Buzzfeed News, LIAR , BS Detector kaggle etc

We used built-in Kaggle datasets for fake and genuine news.
Kaggle allows **users to find and publish data sets, explore and build models in** a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges

The folder has two files; (1) fakeNews.csv, and (2) trueNews.csv. The data in .csv files contain the news article and the corresponding fake rating collected from the USA, India, and Europe regions.

train.csv: A full training dataset with the following attributes:
id: unique id for a news article
title: the title of a news article
author: author of the news article
text: the text of the article; could be incomplete
label: a label that marks the article as potentially unreliable

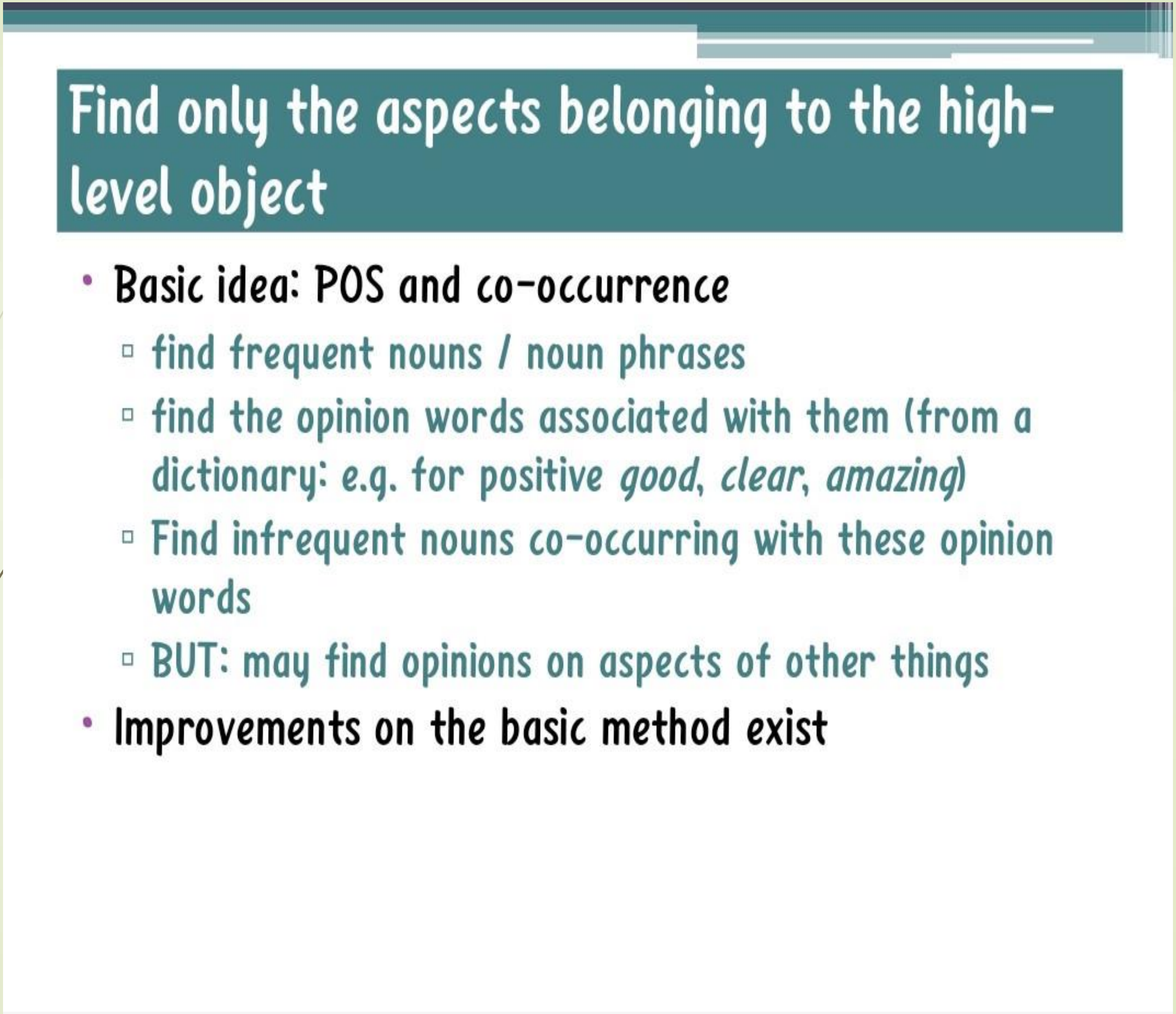


The unit of analysis

- community
- another person
- user / author
- document
- sentence or clause
- aspect (e.g. product feature)

“What makes people happy“ example

Phone example



Find only the aspects belonging to the high-level object

- Basic idea: POS and co-occurrence
  - find frequent nouns / noun phrases
  - find the opinion words associated with them (from a dictionary: e.g. for positive *good*, *clear*, *amazing*)
  - Find infrequent nouns co-occurring with these opinion words
  - BUT: may find opinions on aspects of other things
- Improvements on the basic method exist

Fake News

- Identify the orientation of information in a piece of text

The andaman a
Nicobar island
renamed in
the honour of netaji
Subhash chandra
Bose.

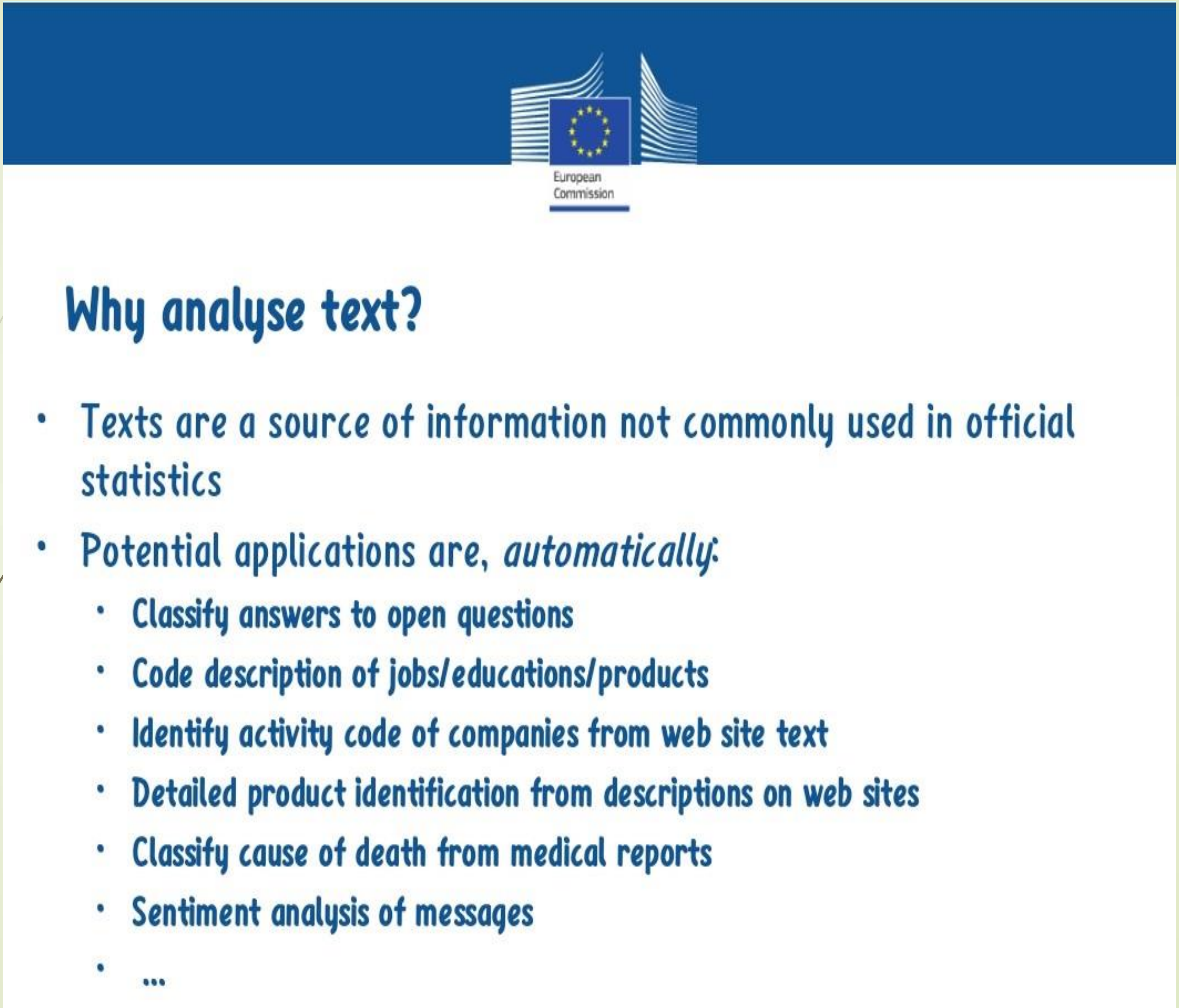
[Genuine]

India will be a
Superpower by
2050.

[Predictory]

Ganga is the longest
River in the world.

[Fake]

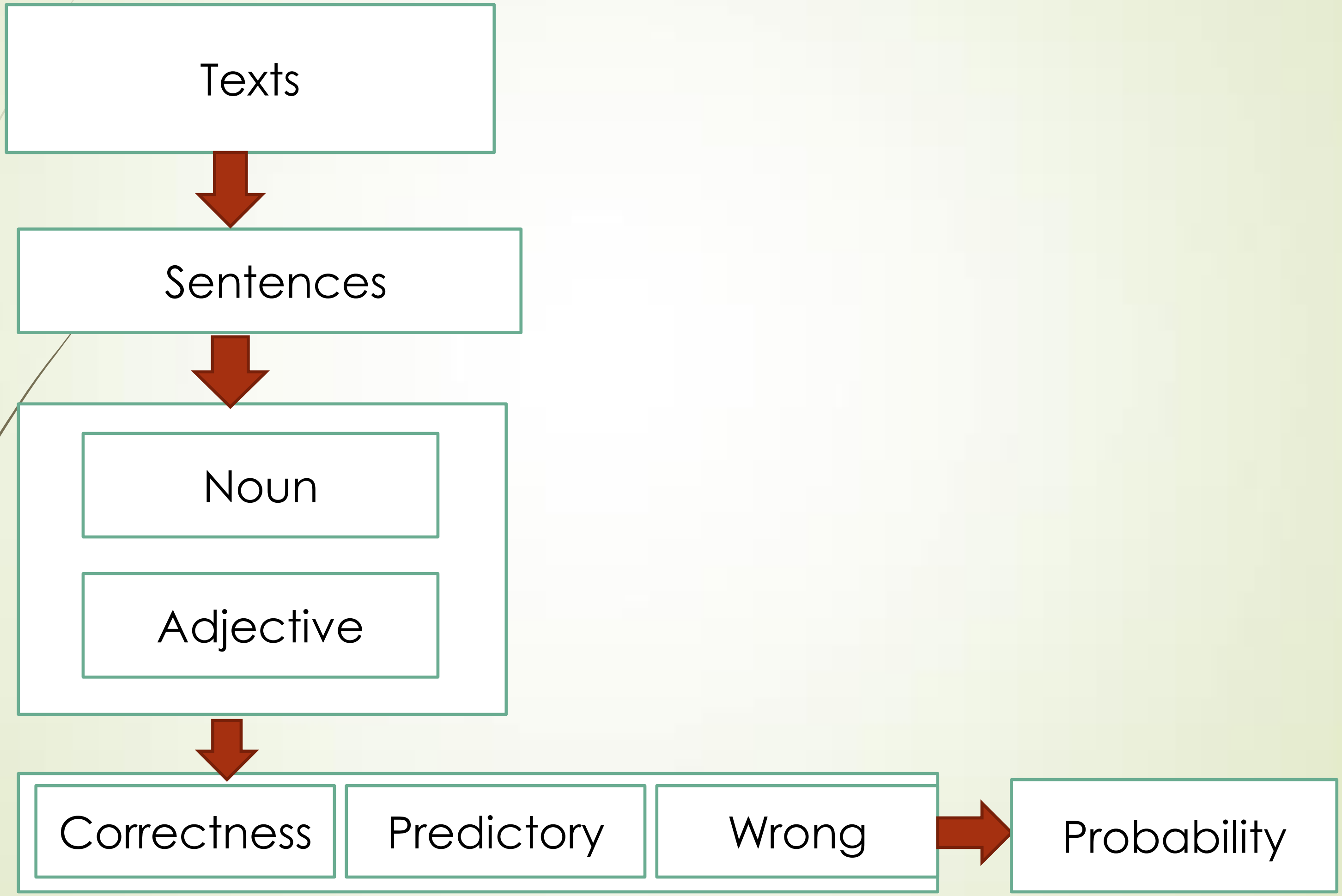


European
Commission

Why analyse text?

- Texts are a source of information not commonly used in official statistics
- Potential applications are, *automatically*:
  - Classify answers to open questions
  - Code description of jobs/educations/products
  - Identify activity code of companies from web site text
  - Detailed product identification from descriptions on web sites
  - Classify cause of death from medical reports
  - Sentiment analysis of messages
  - …

Case Diagram



Texts

Sentences

Noun

Adjective

Correctness

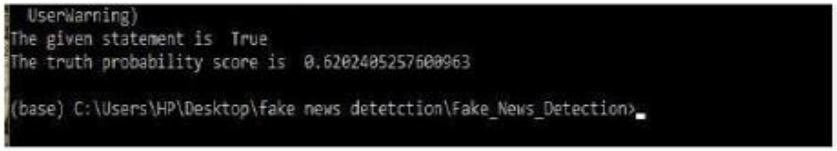
Predictory

Wrong

Probability

Working of the tflite Model

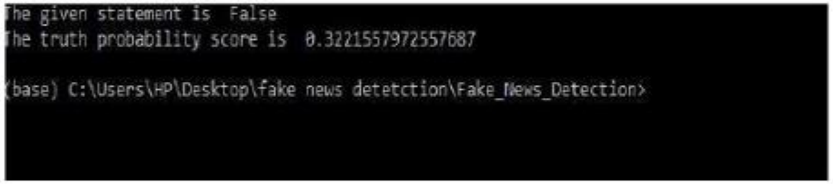
A. Static System-



```
  UserWarning)
The given statement is  True
The truth probability score is  0.6202405257600963

(base) C:\Users\HP\Desktop\fake news detetction\Fake_News_Detection>
```

Figure 3: Static output (True)



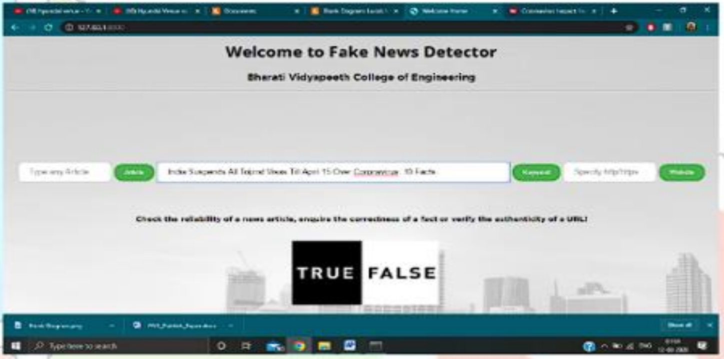
```
The given statement is  False
The truth probability score is  0.3221557972557687

(base) C:\Users\HP\Desktop\fake news detetction\Fake_News_Detection>
```

Figure 4: Static Output (False)

Working of the Android APP

B. Dynamic System-



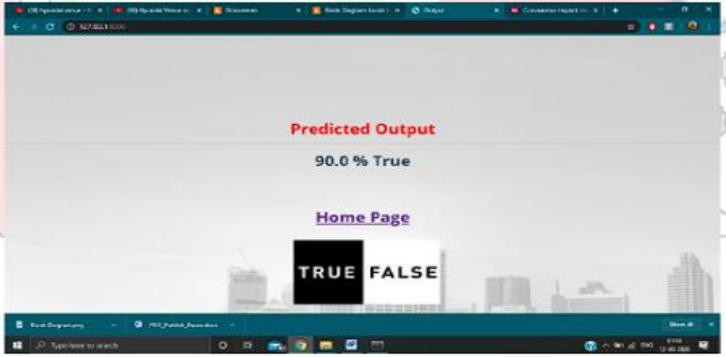
Welcome to Fake News Detector

Bharati Vidyapeeth College of Engineering

India Suspends All Tourist Visas Till April 15 Over Coronavirus: 10 Facts

Check the reliability of a news article, enquire the correctness of a fact or verify the authenticity of a URL!

Figure 5: Fake News Detector (Home Screen)



Predicted Output

90.0 % True

Home Page

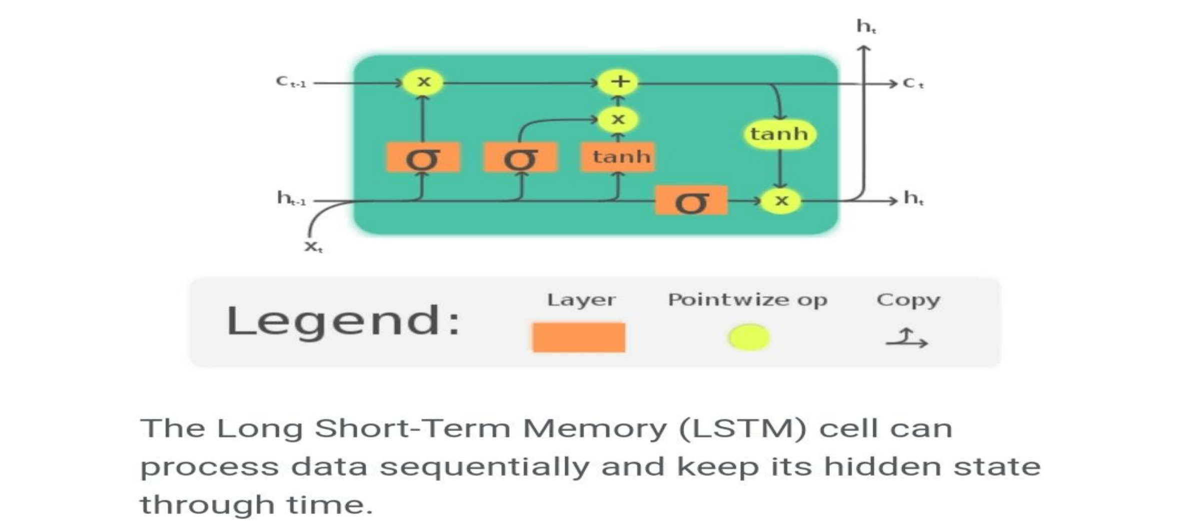
Figure 6: Fake News Detector (Output page)

Areas To Be covered for Fake News Detection

- Long Short-Term Memory (LSTM)
- Recurrent Neural Networks
- Word Embeddings
- Convolutional Neural Networks
- DropOut
- Padding
- TensorFlow.Keras
- Average Pooling and Pooling Layers
- Adam Optimizer

Long Short-Term Memory (LSTM)

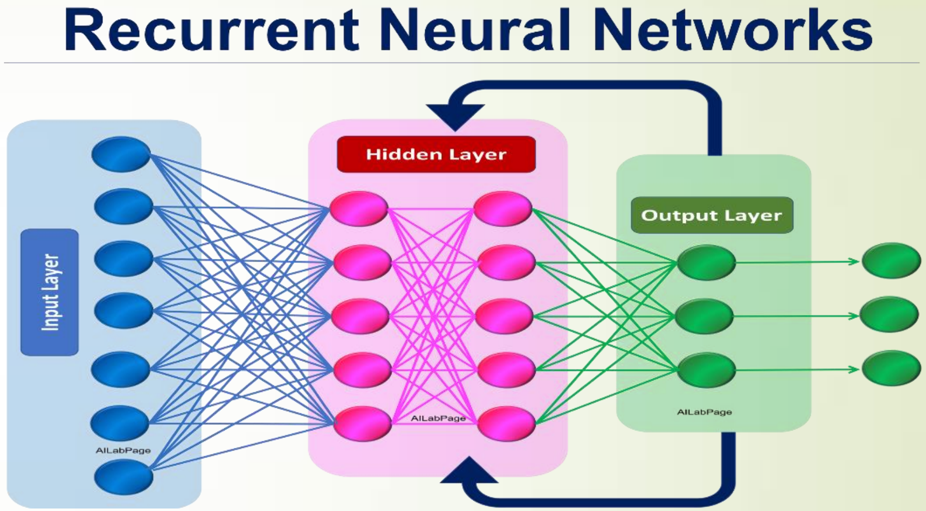
Long short-term memory (**LSTM**) is an artificial **recurrent neural network** (RNN) architecture used in the field of **deep learning**. Unlike standard **feedforward neural networks**, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected **handwriting recognition**, **speech recognition** and anomaly detection in network traffic or IDSs (intrusion detection systems).



The Long Short-Term Memory (LSTM) cell can process data sequentially and keep its hidden state through time.

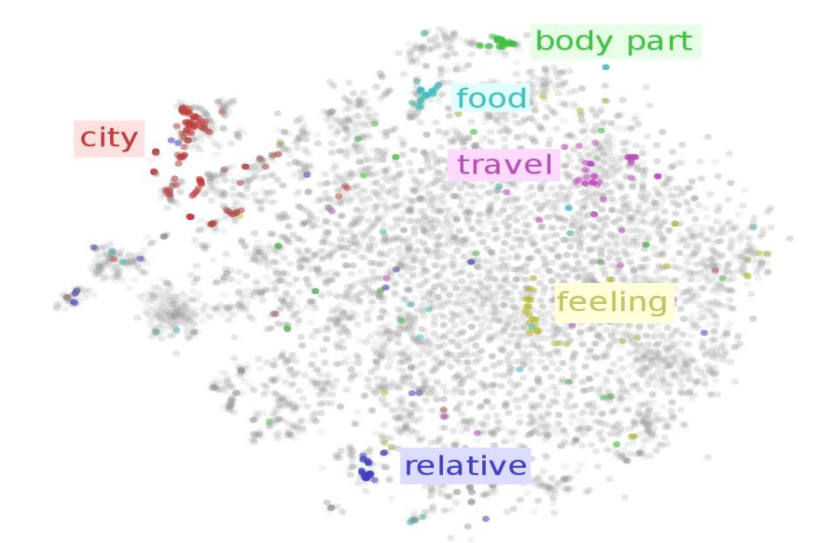
Recurrent Neural Networks

A **recurrent neural network** (**RNN**) is a class of artificial neural networks where connections between nodes form a directed 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.



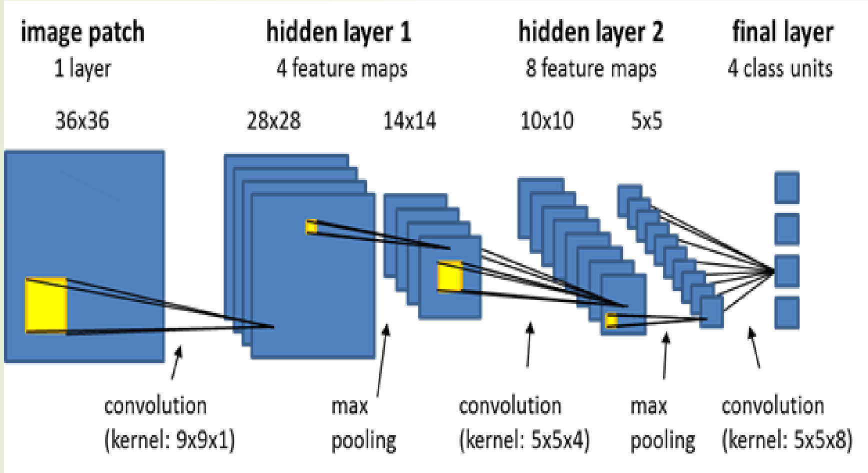
Word Embedding

- In **natural language processing** (NLP), **Word embedding** is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning. Word embeddings can be obtained using a set of **language modeling** and **feature learning** techniques where words or phrases from the vocabulary are mapped to **vectors** of **real numbers**.



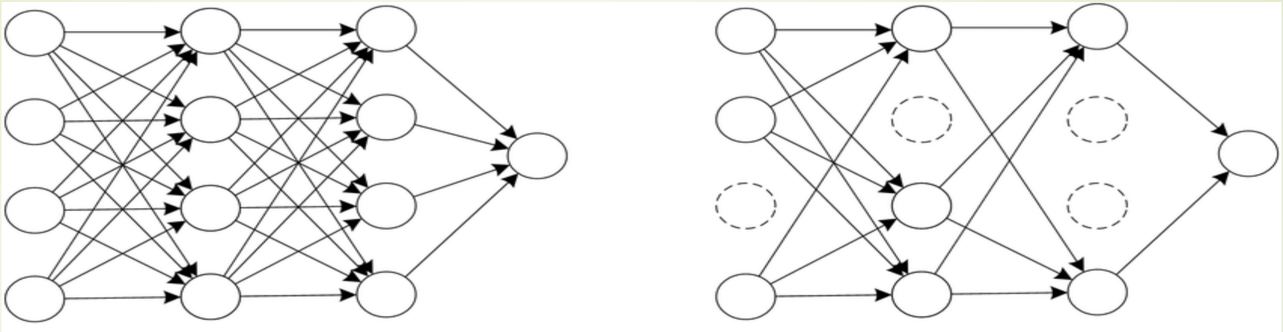
Methods to generate this mapping include **neural networks**, **dimensionality reduction** on the word **co-occurrence matrix**, probabilistic models, explainable knowledge base method, and explicit representation in terms of the context in which words appear.

Convolutional Neural Networks



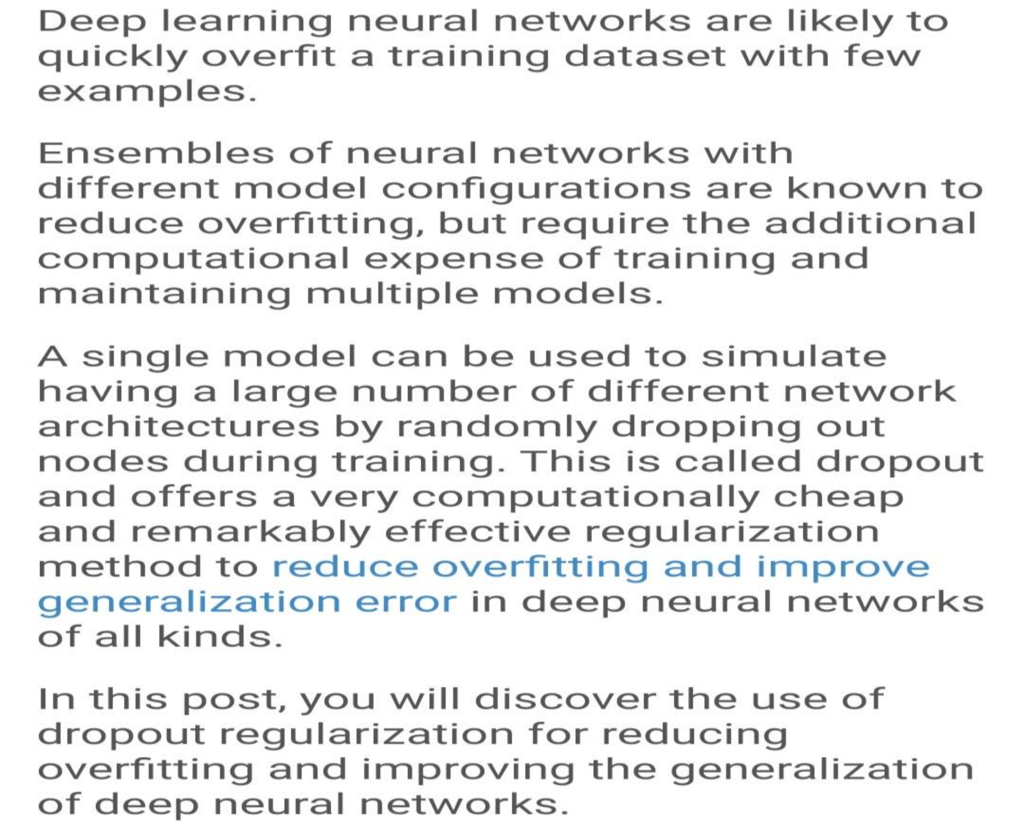
In deep learning, a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep neural network, most commonly applied to analyze visual imagery. They are also known as **shift invariant** or **space invariant artificial neural networks** (**SIANN**), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series.

Drop Out



(a) Standard Neural Network

(b) Network after Dropout



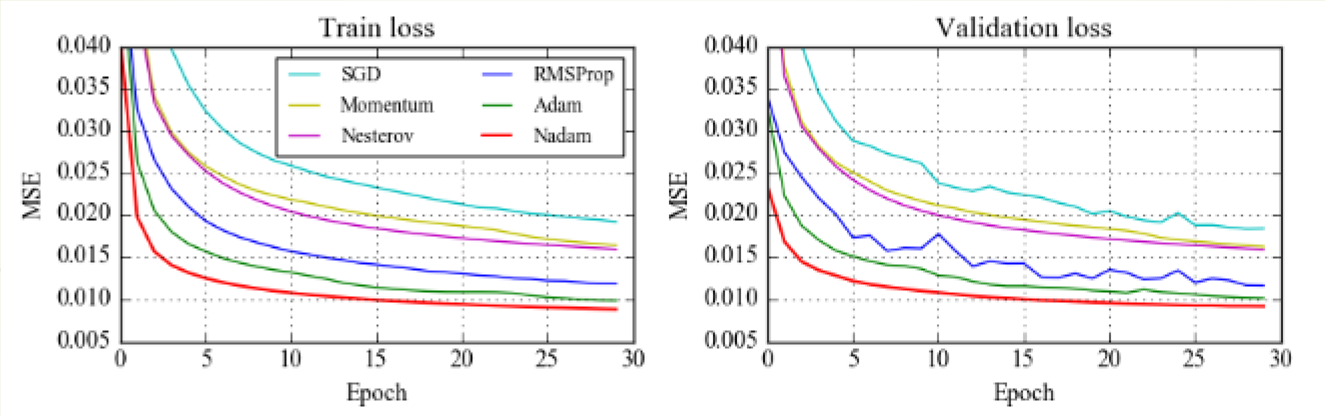
Deep learning neural networks are likely to quickly overfit a training dataset with few examples.

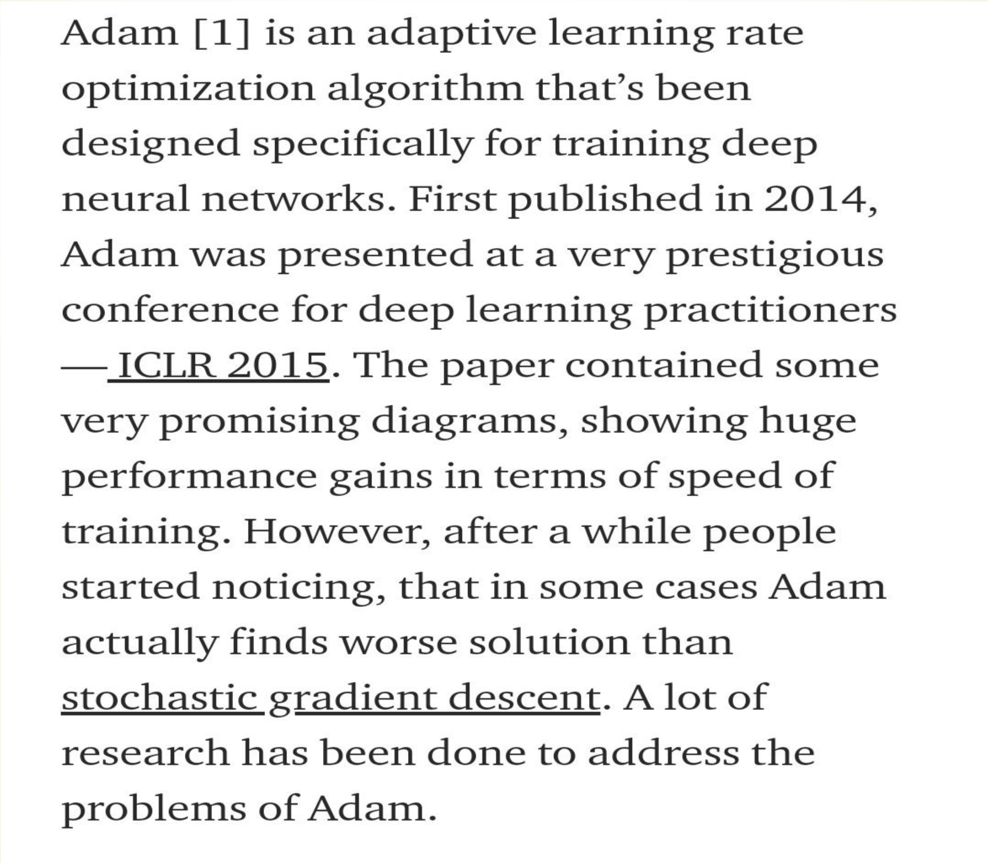
Ensembles of neural networks with different model configurations are known to reduce overfitting, but require the additional computational expense of training and maintaining multiple models.

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is called dropout and offers a very computationally cheap and remarkably effective regularization method to reduce overfitting and improve generalization error in deep neural networks of all kinds.

In this post, you will discover the use of dropout regularization for reducing overfitting and improving the generalization of deep neural networks.

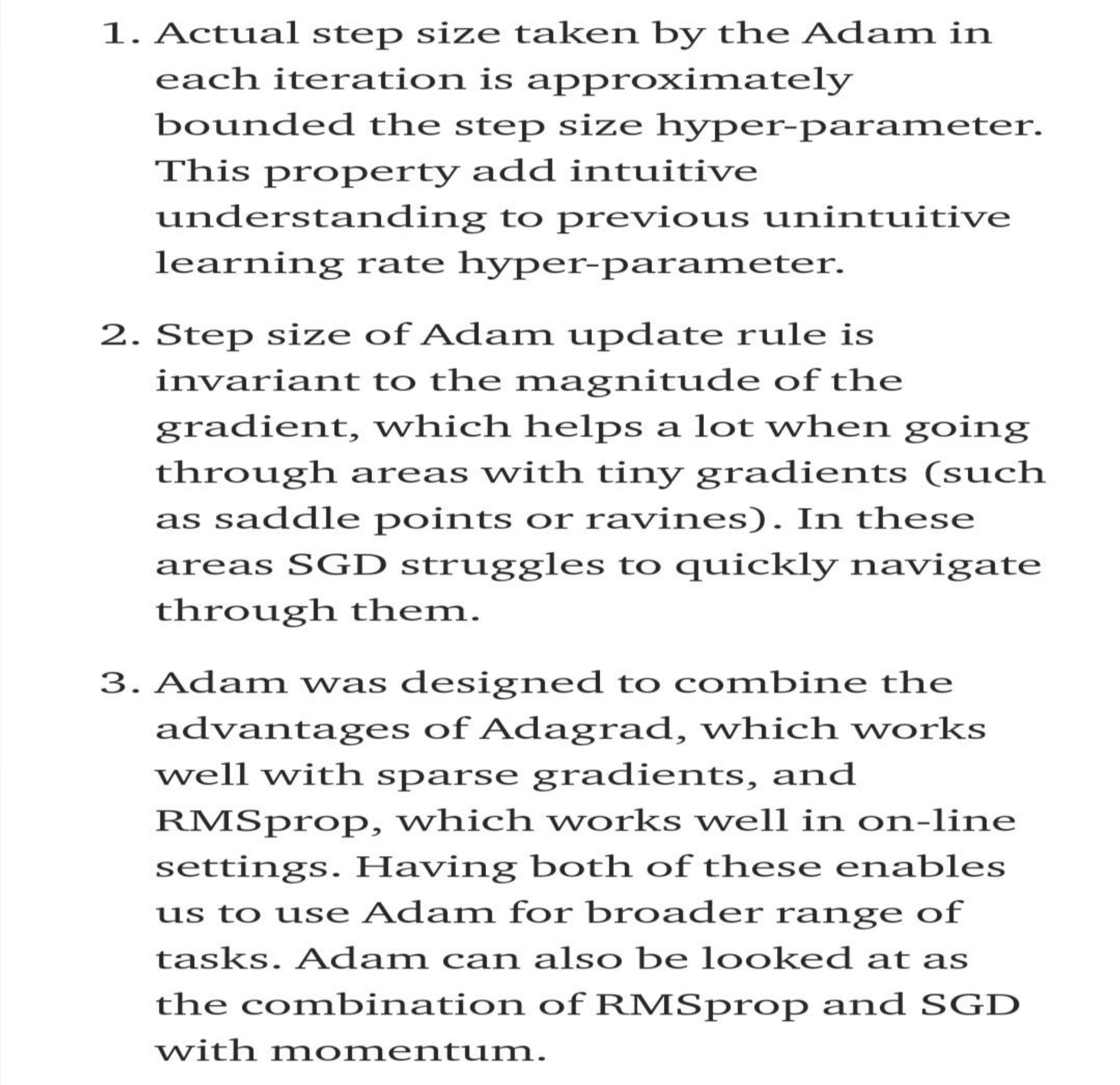
Adam Optimizer





Adam [1] is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. First published in 2014, Adam was presented at a very prestigious conference for deep learning practitioners — ICLR 2015. The paper contained some very promising diagrams, showing huge performance gains in terms of speed of training. However, after a while people started noticing, that in some cases Adam actually finds worse solution than stochastic gradient descent. A lot of research has been done to address the problems of Adam.

Properties Of Adam Optimizer

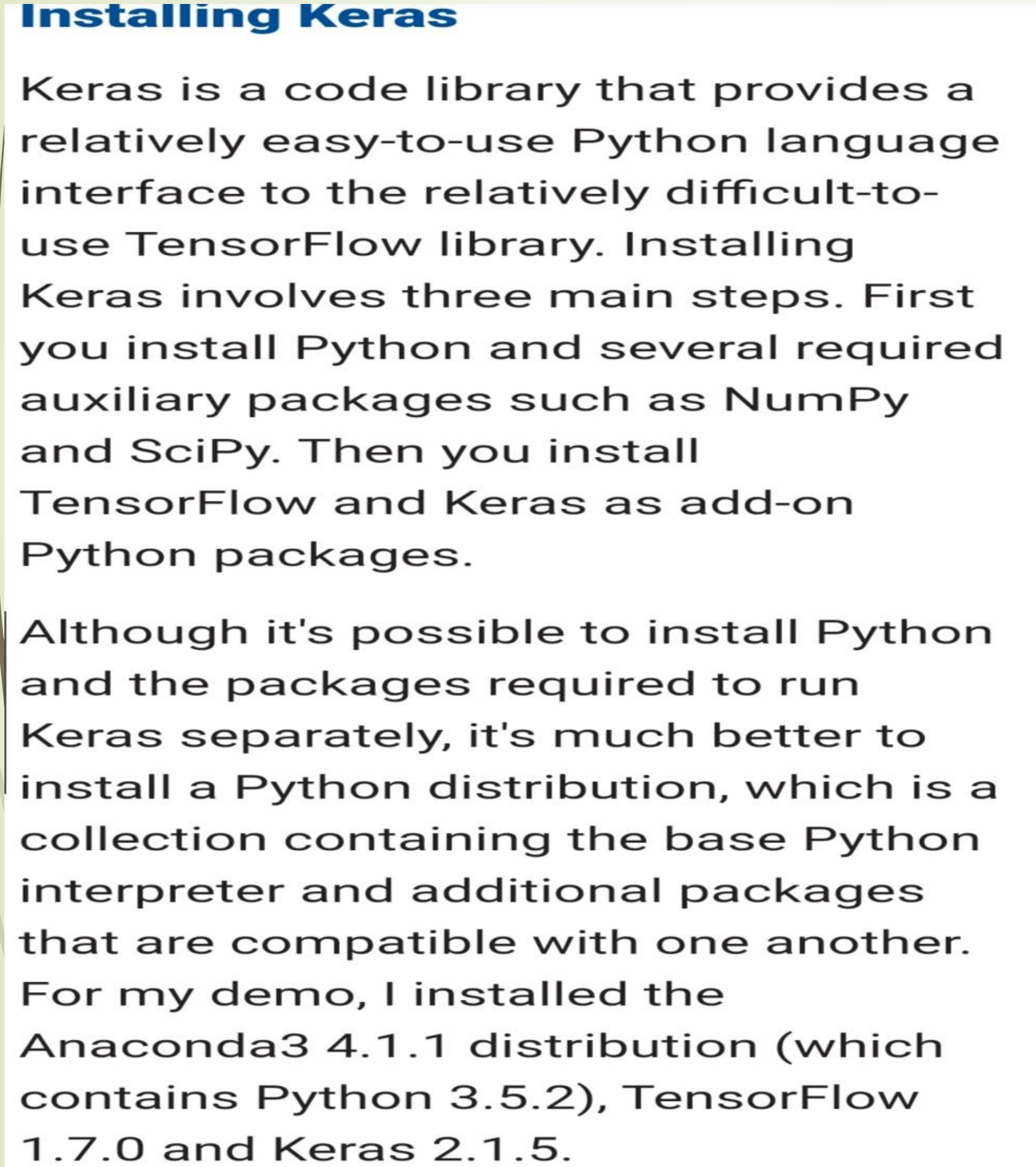


1. Actual step size taken by the Adam in each iteration is approximately bounded the step size hyper-parameter. This property add intuitive understanding to previous unintuitive learning rate hyper-parameter.

2. Step size of Adam update rule is invariant to the magnitude of the gradient, which helps a lot when going through areas with tiny gradients (such as saddle points or ravines). In these areas SGD struggles to quickly navigate through them.

3. Adam was designed to combine the advantages of Adagrad, which works well with sparse gradients, and RMSprop, which works well in on-line settings. Having both of these enables us to use Adam for broader range of tasks. Adam can also be looked at as the combination of RMSprop and SGD with momentum.

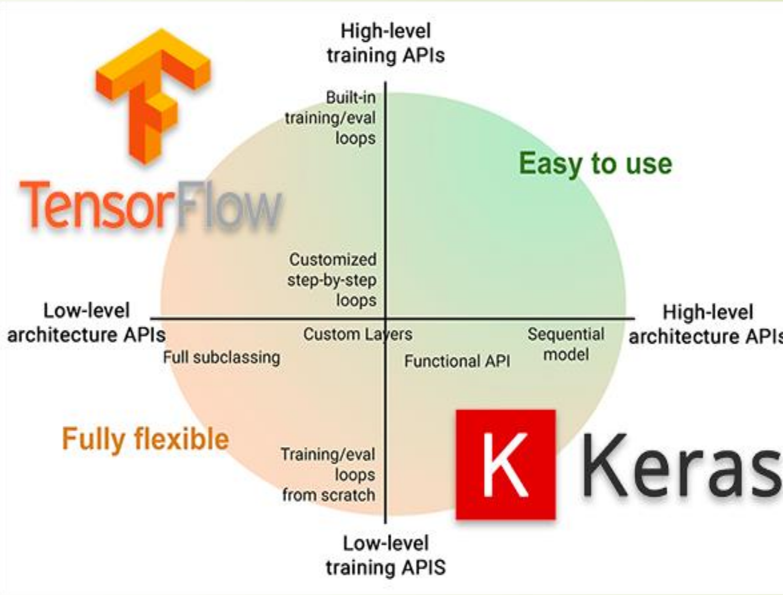
Keras



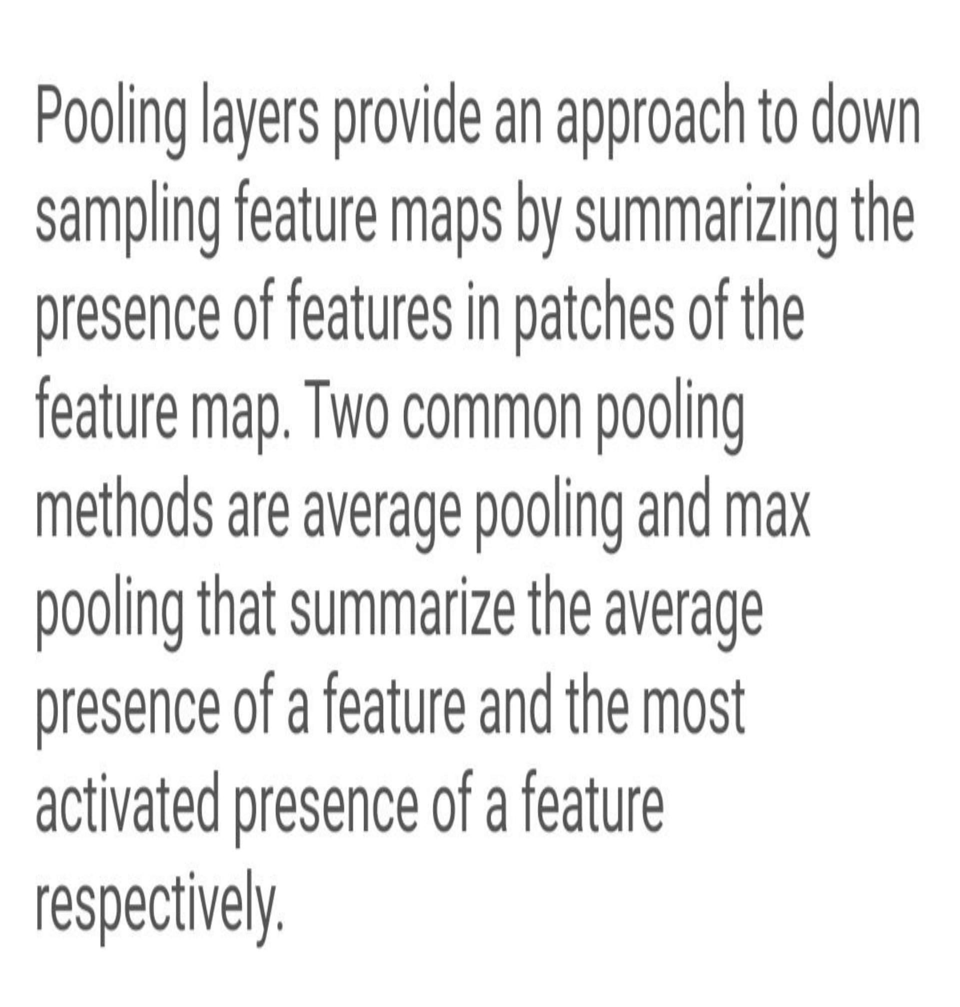
Installing Keras

Keras is a code library that provides a relatively easy-to-use Python language interface to the relatively difficult-to-use TensorFlow library. Installing Keras involves three main steps. First you install Python and several required auxiliary packages such as NumPy and SciPy. Then you install TensorFlow and Keras as add-on Python packages.

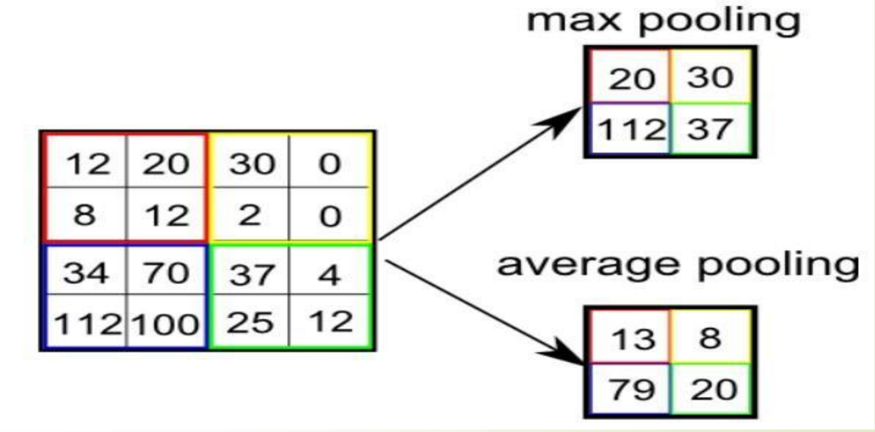
Although it's possible to install Python and the packages required to run Keras separately, it's much better to install a Python distribution, which is a collection containing the base Python interpreter and additional packages that are compatible with one another. For my demo, I installed the Anaconda3 4.1.1 distribution (which contains Python 3.5.2), TensorFlow 1.7.0 and Keras 2.1.5.



Pooling Layer and Average Pooling



Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively.



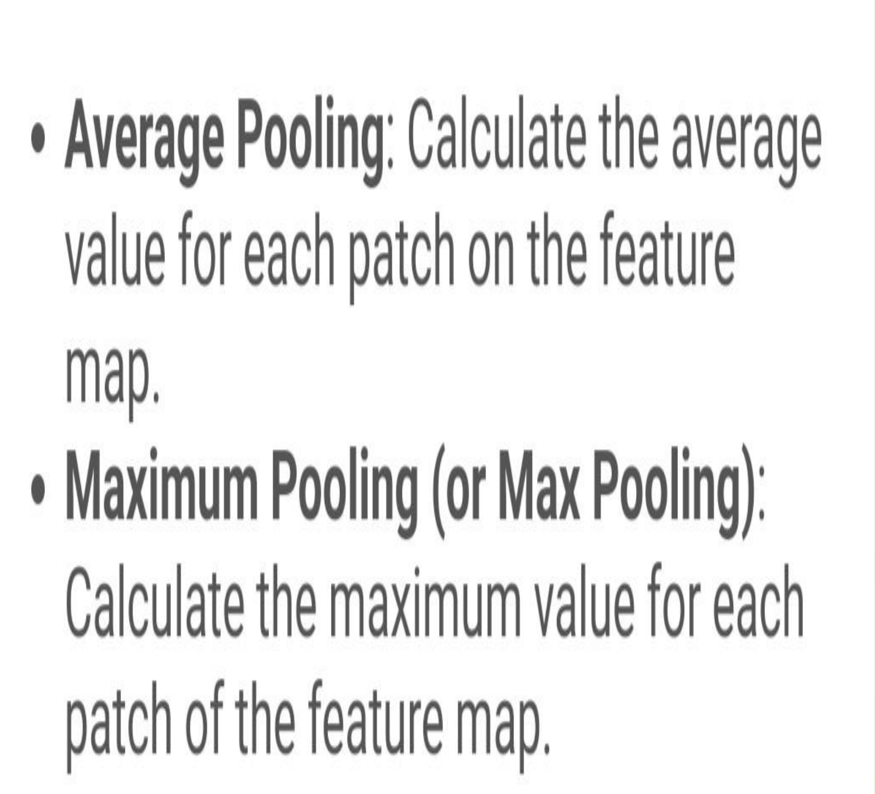
12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

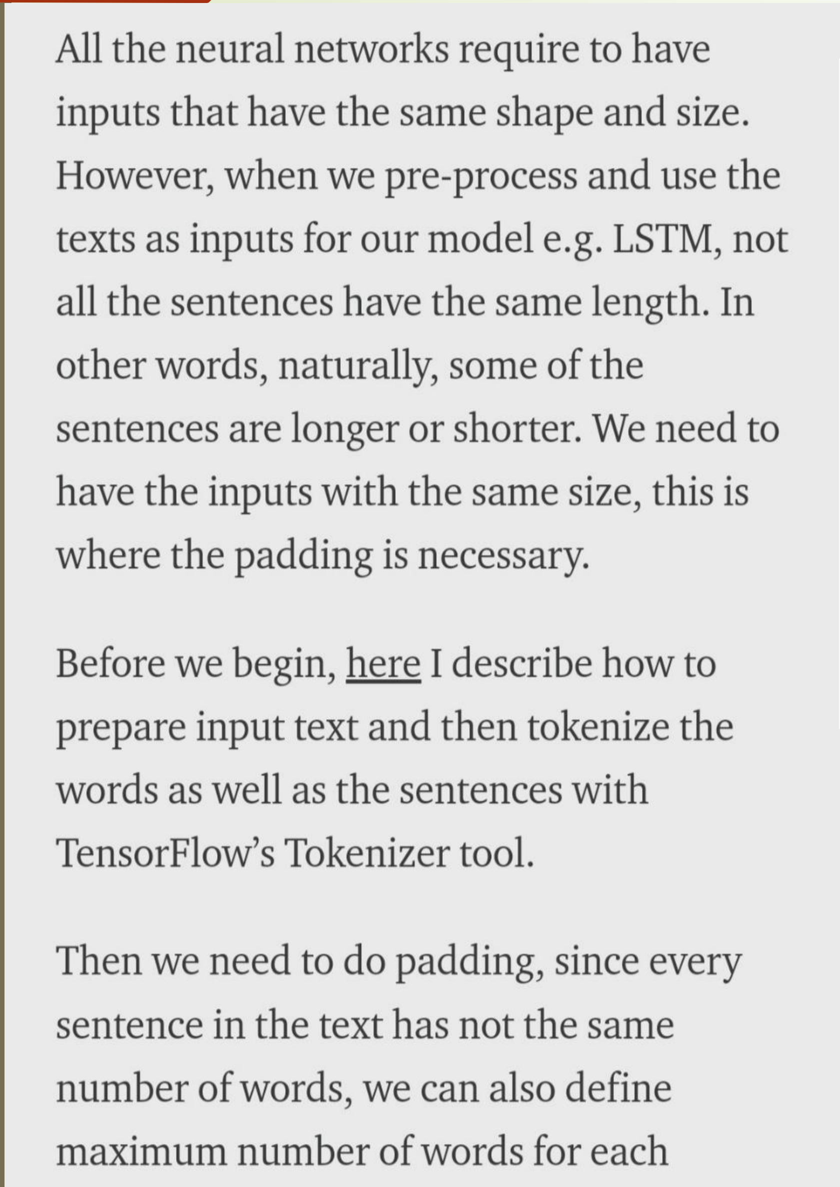
average pooling

13	8
79	20



- **Average Pooling**: Calculate the average value for each patch on the feature map.
- **Maximum Pooling (or Max Pooling)**: Calculate the maximum value for each patch of the feature map.

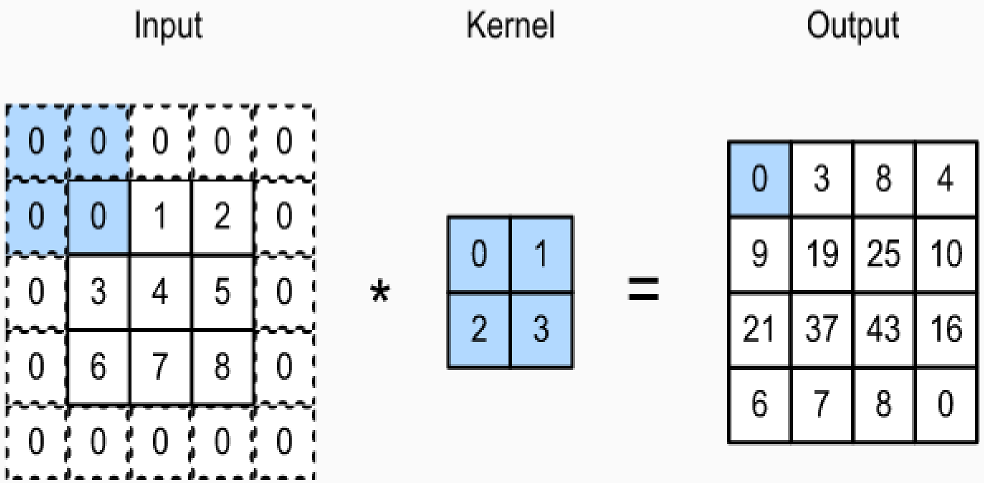
Padding



All the neural networks require to have inputs that have the same shape and size. However, when we pre-process and use the texts as inputs for our model e.g. LSTM, not all the sentences have the same length. In other words, naturally, some of the sentences are longer or shorter. We need to have the inputs with the same size, this is where the padding is necessary.

Before we begin, here I describe how to prepare input text and then tokenize the words as well as the sentences with TensorFlow's Tokenizer tool.

Then we need to do padding, since every sentence in the text has not the same number of words, we can also define maximum number of words for each



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 3 & 4 & 5 & 0 \\ 0 & 6 & 7 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 8 & 4 \\ 9 & 19 & 25 & 10 \\ 21 & 37 & 43 & 16 \\ 6 & 7 & 8 & 0 \end{bmatrix}$$

Two-dimensional cross-correlation with padding.

Software Specifications

FRONT END	: Android Studio
BACK END	: tflite, Python, Kotlin, Pandas
DataSets	: Kaggle
PLATFORM	: Windows

RESULT

Implementation was done using the above algorithms with Vector features- Count Vectors and Tf-Idf vectors at Word level and N gram level.

Accuracy was noted for all models.

We used K-fold cross validation technique to improve the effectiveness of the models.

A. Dataset split using K-fold cross validation This cross-validation technique was used for splitting the dataset randomly into k-folds. (k-1) folds were used for building the model while kth fold was used to check the effectiveness of the model. This was repeated until each of the k-folds served as the test set. I used 3- fold cross validation for this experiment where 67% of the data is used for training the model and remaining 33% for testing.

B. Confusion Matrices for Static System After applying various extracted features (Bag-of-words, Tf-Idf. N-grams) on three different classifiers (Naïve bayes, Logistic Regression and Random Forest), their confusion matrix showing actual set and predicted sets

CONCLUSION AND FUTURE SCOPE

In the 21st century, the majority of the tasks are done online.

Newspapers that were earlier preferred as hard-copies are now being substituted by applications like Facebook, Twitter, and news articles to be read online.

Whatsapp's forwards are also a major source.

The growing problem of fake news only makes things more complicated and tries to change or hamper the opinion and attitude of people towards use of digital technology.

When a person is deceived by the real news two possible things happen- People start believing that their perceptions about a particular topic are true as assumed.

Thus, in order to curb the phenomenon, we have developed our Fake news Detection system that takes input from the user and classify it to be true or fake.

To implement this, various NLP and Machine Learning Techniques have to be used.

The model is trained using an appropriate dataset and performance evaluation is also done using various performance measures.

The best model, i.e. the model with highest accuracy is used to classify the news headlines or articles.

As evident above for static search, our best model came out to be Logistic Regression with an accuracy of 65%.

Hence we then used grid search parameter optimization to increase the performance of logistic regression which then gave us the accuracy of 75%.

Hence we can say that if a user feed a particular news article or its headline in our model, there are 75% chances that it will be classified to its true nature.

The user can check the news article or keywords online; he can also check the authenticity of the website.

The accuracy for dynamic system is 93% and it increases with every iteration.

We intend to build our own dataset which will be kept up to date according to the latest news.

All the live news and latest data will be kept in a database using Web Crawler and online database.

References

- https://www.GeeksforGeeks.org/
- https://www.python.org/
- https://www.kaggle.com/
- https://docs.python.org/3/tutorial/
- https://www.google.com/
- https://www.Wikipedia.org/

Thank You