GRT INSTITUTE OF
ENGINEERING AND
TECHNOLOGY, TIRUTTANI - 631209
Approved by AICTE, New Delhi Affiliated to Anna University, Chennai

NAAN MUDHALVAN PROJECT(IBM)

IBM AI 101 ARTIFICIAL INTELLIGENCE-GROUP 1

DONE BY
GOKUL NATH S
(Email: 12345.gokulnath.s@gmail.com)
(NM ID: au110321106013)
ECE 3Rd Year
From the Department of
ELECTRONICS AND COMMUNICATION ENGINEERING

NAAN MUDHALVAN PROJECT(IBM)

IBM AI 101 ARTIFICIAL INTELLIGENCE-GROUP 1

PROJECT:

TEAM-6 FAKE NEWS DETECTION USING NLP

PHASE III : DEVELOPMENT PART-1

What is mean by fake news detection?

Fake news detection refers to the process of identifying and verifying the accuracy of news, information, or stories that are circulated, published, or shared, primarily through digital or social media platforms, but which are intentionally false or misleading. The term "fake news" can encompass various types of misleading or deceptive content, including:

1. **Misinformation:** Inaccurate information that is shared without malicious intent. It may result from errors, misunderstandings, or misinterpretations.

2. **Disinformation:** Deliberately false information spread with the intent to deceive, manipulate public opinion, or achieve some other agenda, often for political, financial, or ideological purposes.

3. **Malinformation:** True information shared with the intent to harm, discredit, or harass individuals or entities. This can include sharing private information or facts taken out of context.

Detecting fake news typically involves a combination of manual and automated processes, often using technology and fact-checking methods:

1. **Fact-Checking:** Organizations and individuals fact-check claims made in news stories, scrutinizing their accuracy. Fact-checkers research and verify claims with credible sources.

2. **Source Analysis:** Examining the source of the information is crucial. Authentic news usually comes from established, reputable news outlets or recognized experts. Fake news may originate from less credible sources or obscure websites.

3. **Content Analysis:** Analyzing the content for inconsistencies, sensationalism, or bias can help identify potential fake news. Suspicious stories often lack corroborating evidence or cite unreliable sources.

4. **Reverse Image Search:** Fake news may include manipulated or misleading images. Reverse image searches can help identify the true origin of a picture and reveal if it has been altered.

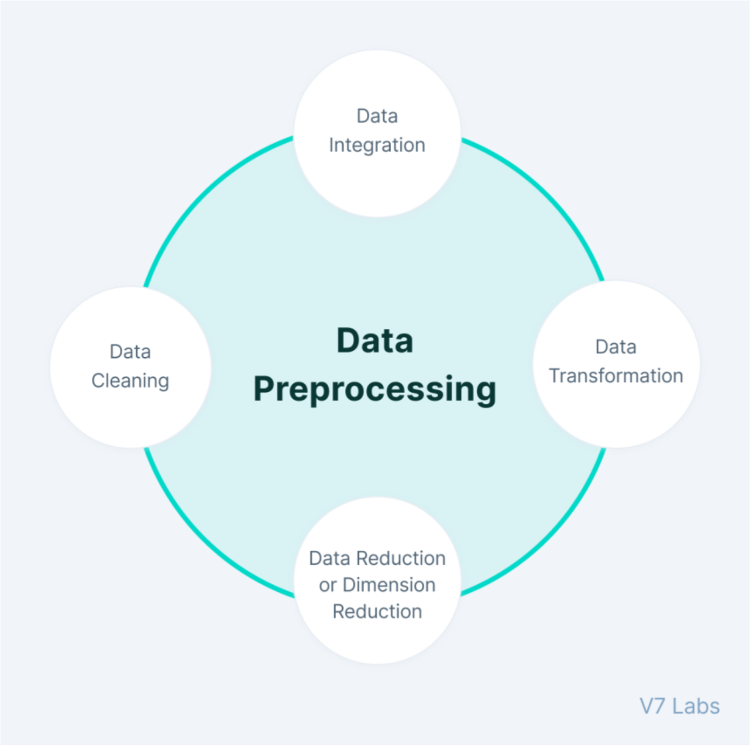
5. **Text Analysis:** Analyzing the language, writing style, and grammar of a news article or social media post can sometimes uncover signs of fake news.

6. **Social Media Analysis:** Monitoring the spread of news on social media platforms and identifying patterns of rapid sharing or bot-driven dissemination can be useful in spotting fake news.

7. **Machine Learning and AI:** Automated algorithms and machine learning models can be trained to analyze patterns in text and content to detect potential fake news. These models can assist in flagging suspicious content for human review.

Detecting fake news is an ongoing challenge due to the evolving nature of misinformation and the proliferation of digital platforms. It requires vigilance, critical thinking, and the cooperation of individuals, fact-checkers, social media platforms, and technology to combat the spread of false information. It's important to verify information from multiple reliable sources before accepting it as true.

PRE-PROCESSING THE DATASET



With the explosion of online fake news and disinformation, it is increasingly difficult to discern fact from fiction. And as natural language processing become more popular, Fake News detection serves as a great introduction to NLP.

Google Cloud Natural Language API is a great platform to use for this project. Simply upload a dataset, train the model, and use it to predict new articles.

But before we download a Kaggle dataset and get cracking on Google Cloud, it’s in our best interest to pre-process the dataset.

What is pre-processing?

To preprocess your text simply means to bring your text into a form that is predictable and analyzable for your task. The goal of pre-processing is to remove noise. By removing unnecessary features from our text, we can reduce complexity and increase predictability (i.e. our model is faster and better). Removing punctuation, special characters, and ‘filler’ words (the, a, etc.) does not drastically change the meaning of a text.

Approach:

There are many types of text pre-processing and their approaches varied. We will cover the following:

1. Lowercase Text
2. URL Removal
3. Contraction Splitting
4. Tokenization
5. Stemming
6. Lemmatization
7. Stop Word Removal

We’ll be using python due to the availability and power of its data analysis and NLP libraries.

Lowercase & URL Removal

Before we start any of the pre-processing heavy lifting, we want to convert our text to lowercase and remove any URLs in our text. A simple regex expression can handle this for us.

CODE:

```
import re

text = "http://www.google.com hello world"

text = re.sub(r'http\S+', '', text.lower())

print(text)

# hello world
```

Split Contractions

Similar to URLs, contractions can produce unintended results if left alone. The aptly named contractions python library to the rescue! It looks for contractions and splits them into root words.

CODE:

```
import contractions

def remove_contractions(text):

    return ' '.join([contractions.fix(word) for word in 
text.split()])

text = """can't won't shouldn't there's mustn't"""

print(remove_contractions(text))

# can not will not should not there is must not
```

Tokenization

At it’s simplest, tokenization is splitting text into pieces.

Tokenization is breaking the raw text into small chunks. Tokenization breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words.

There are a multitude of ways to implement tokenization and their approaches varied. For our project we utilized RegexpTokenizer within the NLTK library. Using regular expressions, RegexpTokenizer will match either tokens or separators (i.e. include or exclude regex matches).

CODE:

```
import nltk

from nltk.tokenize import RegexpTokenizer

# Create tokens out of alphanumeric characters

tokenizer = RegexpTokenizer(r'\w+')

tokens = tokenizer.tokenize("I think pineapple pizza is
gross and not worth $15!")

print(tokens)

# ['I', 'think', 'pineapple', 'pizza', 'is', 'gross', 'and',
'not', 'worth', '15']
```

Our string input is split by grouping alphanumeric characters. Notice the “$” and “!” characters do not appear in our tokenized list.

Stemming

Text normalization is the process of simplifying multiple variations or tenses of the same word. Stemming and lemmatization are two methods of text normalization, the former being the simpler of the two. To stem a word, we simply remove the suffix of a word to reduce it to its root.

CODE:

```
# Using Porter Stemmer implementation in nltk
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
def stem(tokens):
    return [stemmer.stem(token) for token in tokens]
tokens = ['jumped', 'jumps', 'jumped']
print(stem(tokens))
# ['jump', 'jump', 'jump']
```

As an example, “jumping”, “jumps”, and “jumped” all are stemmed to “jump.”

Stemming is not without its faults, however. We can run into the issue of overstemming. Overstemming is when words with different meanings are stemmed to the same root — a false positive.

CODE:

```
tokens = ['universal', 'university', 'universe']
print(stem(tokens))
# ['univers', 'univers', 'univers']
```

Understemming is also a concern. See how words that should stem to the same root do not — a false negative.

CODE:

```
tokens = ['alumnus', 'alumni', 'alumnae']
print(stem(tokens))
# ['alumnu', 'alumni', 'alumna']
```

Let’s take a look at a more nuanced approach to text normalization, lemmatization.

Lemmatization

Lemmatization is the process of converting a word to its base form. The difference between stemming and lemmatization is, lemmatization considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errors.

Lemmatization differs from stemming in that it determines a words part of speech by looking at surrounding words for context. For this example we use nltk.pos_tag to assign parts of speech to tokens. We then pass the token and its assigned tag into WordNetLemmatizer, which decides how to lemmatize the token.

CODE:

```
import nltk

from nltk.stem import WordNetLemmatizer

from nltk.corpus import wordnet

lemmatizer = WordNetLemmatizer()

# Convert the nltk pos tags to tags that wordnet can
recognize

def nltkToWordnet(nltk_tag):

  if nltk_tag.startswith('J'):

    return wordnet.ADJ

  elif nltk_tag.startswith('V'):

    return wordnet.VERB

  elif nltk_tag.startswith('N'):

    return wordnet.NOUN

  elif nltk_tag.startswith('R'):

    return wordnet.ADV

  else:

    return None

 # Lemmatize a list of words/tokens

def lemmatize(tokens):

  pos_tags = nltk.pos_tag(tokens)

  res_words = []

 for word, tag in pos_tags:

    tag = nltkToWordnet(tag)

    if tag is None:
      res_words.append(word)
    else:
      res_words.append(lemmatizer.lemmatize(word, tag))
  return res_words
```

Using the following text we can compare the results of our approaches to stemming and lemmatization.

CODE:

```
text = "it takes a great deal of bravery to stand up to our
enemies, but just as much to stand up to our friends"
tokens = tokenizer.tokenize(text)
# STEMMING RESULTS
print(stem(tokens))
#['it', 'take', 'a', 'great', 'deal', 'of', 'braveri', 'to',
'stand', 'up', 'to', 'our', 'enemi', 'but', 'just', 'as',
'much', 'to', 'stand', 'up', 'to', 'our', 'friend']
# LEMMATIZING RESULTS
print(lemmatize(tokens))
#['it', 'take', 'a', 'great', 'deal', 'of', 'bravery', 'to',
'stand', 'up', 'to', 'our', 'enemy', 'but', 'just', 'as',
'much', 'to', 'stand', 'up', 'to', 'our', 'friend']
```

Notice that ‘enemies’ was stemmed to ‘enemi’ but lemmatized to ‘enemy’. Interestingly, ‘bravery’ was stemmed to ‘braveri’ but the lemmatizer did not change the original word. In general, lemmatization is more precise, but at the expense of complexity.

Stop Word Removal

Let’s start this one off with an example. What comes to your mind when you read the following?

“quick brown fox lazy dog”

Hopefully you read that and thought of the common English pangram:

“the quick brown fox jumped over the lazy dog”

If you got it, you didn’t need the missing words to know what I was referencing. “The” doesn’t add any meaning to the sentence, and you had enough context that “jumped” and “over” weren’t necessary. In essence, I removed the stop words.

Stop words are words in the text which do not add any meaning to the sentence and their removal will not affect the processing of text for the defined purpose. They are removed from the vocabulary to reduce noise and to reduce the dimension of the feature set.

CODE:

```
import nltk
nltk.download('words') #download list of english words
nltk.download('stopwords') #download list of stopwords
from nltk.corpus import stopwords
stopWords = stopwords.words('english')
englishWords = set(nltk.corpus.words.words())

def remove_stopWords(tokens):
    return [w for w in tokens if (w in englishWords and w
not in stopWords)]
```

```
tokens =
['the','quick','brown','fox','jumped','over','the','lazy','d
og']
print(remove_stopWords(tokens))
# ['quick', 'brown', 'fox', 'lazy', 'dog']
```

Again the NLTK library comes in clutch here. We can download a set of English words and stop words and compare that against our input tokens (see tokenization). For the purpose of our fake news detector, we return tokens that are English but aren't stop words.

All Together Now

Combining all our steps together (minus stemming), let's compare the before and after. We'll use the following onion article as our article to pre-process. If you want to try it for yourself, check out this google collab.

LAKEWOOD, OH — Following a custom born out of cooperation and respect, local drivers reportedly pulled over to the side of the road Friday to let a pizza delivery guy through. "Gee, I hope it's nothing serious like a big, hungry party," said 48-year-old Rosanna Tuttle, who was just one of the dozens of drivers who quickly moved to the shoulder of the road after catching sight of the speeding pizza-delivery vehicle swerving through traffic in the rearview mirror. "It's honestly just a reflex. Sure, it slows everyone down, but wouldn't you want others to pull over for you if that was your pizza in there? I don't care if I'm late; I just hope that pizza is okay. Let's pray they get there safe." At press time, drivers at the scene had stopped their cars again to rubberneck as the delivery guy rushed into an apartment building carrying a large stack of pizzas and mozzarella sticks.

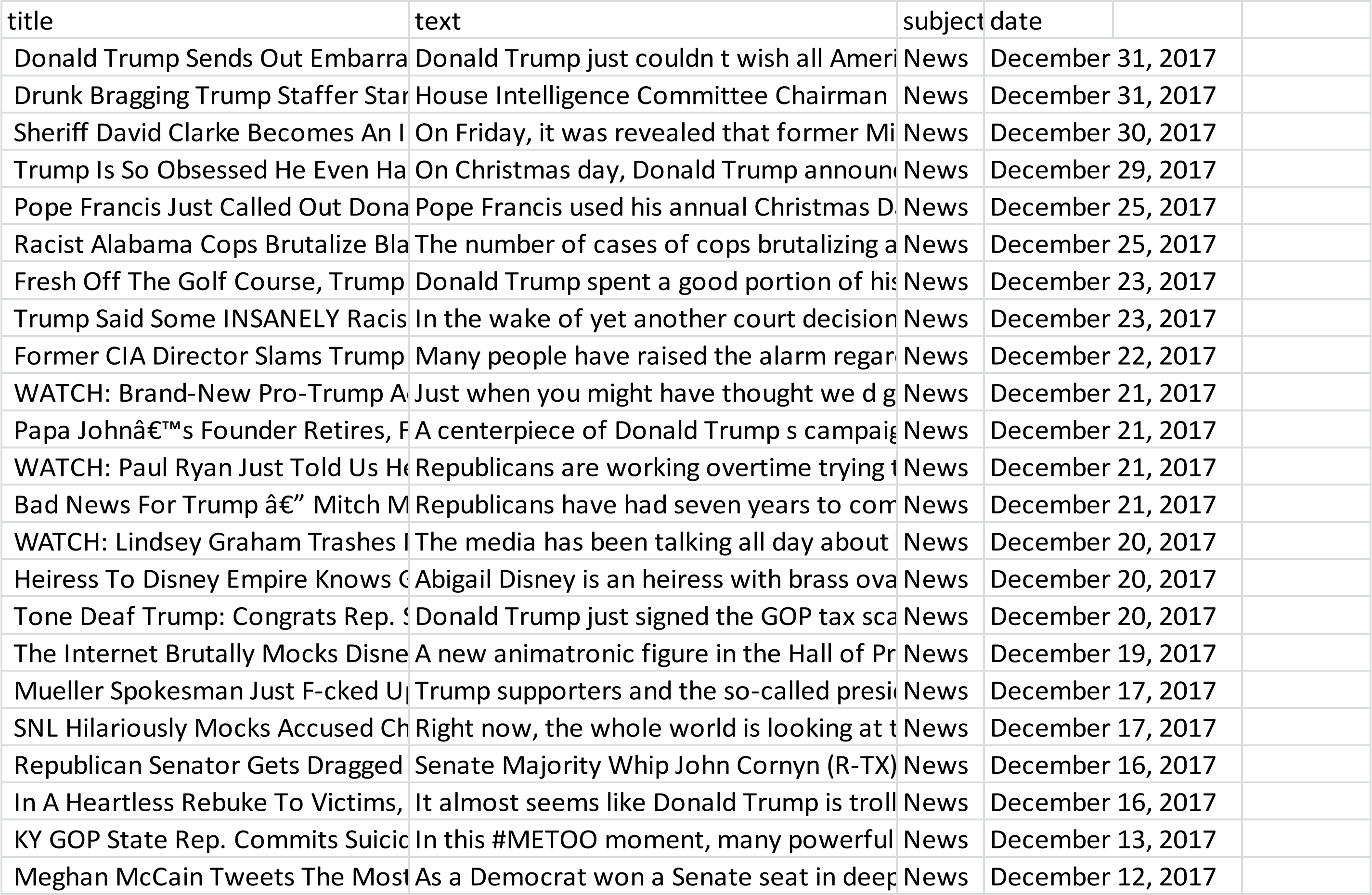
Returns the following.

```
oh follow custom bear respect local driver reportedly pull side road let pizza
delivery guy gee hope nothing serious like big hungry party say year old one dozen
driver quickly move shoulder road catch sight speed pizza delivery vehicle swerve
traffic mirror honestly reflex sure slow everyone would want pull pizza care I late
hope pizza let us pray get safe press time driver scene stop car rubberneck
delivery guy rush apartment building carry large stack pizza stick
```

That’s A Wrap:

Natural Language Processing is a powerful tool to tackle some really interesting questions. Text pre-processing is an integral step in the process, helping us find the signal through the noise. “Garbage in garbage out,” as they say. I hope this article answered some questions and raised a few more — this is just the tip of the iceberg!

DATASET



title	text	subject	date		
Donald Trump Sends Out Embarra	Donald Trump just couldn t wish all Ameri	News	December 31, 2017		
Drunk Bragging Trump Staffer Star	House Intelligence Committee Chairman	News	December 31, 2017		
Sheriff David Clarke Becomes An I	On Friday, it was revealed that former Mi	News	December 30, 2017		
Trump Is So Obsessed He Even Ha	On Christmas day, Donald Trump announ	News	December 29, 2017		
Pope Francis Just Called Out Dona	Pope Francis used his annual Christmas D	News	December 25, 2017		
Racist Alabama Cops Brutalize Bla	The number of cases of cops brutalizing a	News	December 25, 2017		
Fresh Off The Golf Course, Trump	Donald Trump spent a good portion of his	News	December 23, 2017		
Trump Said Some INSANELY Racis	In the wake of yet another court decision	News	December 23, 2017		
Former CIA Director Slams Trump	Many people have raised the alarm regar	News	December 22, 2017		
WATCH: Brand-New Pro-Trump A	Just when you might have thought we d g	News	December 21, 2017		
Papa Johnâ€™s Founder Retires, F	A centerpiece of Donald Trump s campaig	News	December 21, 2017		
WATCH: Paul Ryan Just Told Us He	Republicans are working overtime trying t	News	December 21, 2017		
Bad News For Trump â€” Mitch M	Republicans have had seven years to com	News	December 21, 2017		
WATCH: Lindsey Graham Trashes I	The media has been talking all day about	News	December 20, 2017		
Heiress To Disney Empire Knows G	Abigail Disney is an heiress with brass ova	News	December 20, 2017		
Tone Deaf Trump: Congrats Rep. S	Donald Trump just signed the GOP tax sca	News	December 20, 2017		
The Internet Brutally Mocks Disne	A new animatronic figure in the Hall of Pr	News	December 19, 2017		
Mueller Spokesman Just F-cked Up	Trump supporters and the so-called presi	News	December 17, 2017		
SNL Hilariously Mocks Accused Ch	Right now, the whole world is looking at t	News	December 17, 2017		
Republican Senator Gets Dragged	Senate Majority Whip John Cornyn (R-TX)	News	December 16, 2017		
In A Heartless Rebuke To Victims,	It almost seems like Donald Trump is troll	News	December 16, 2017		
KY GOP State Rep. Commits Suicid	In this #METOO moment, many powerful	News	December 13, 2017		
Meghan McCain Tweets The Most	As a Democrat won a Senate seat in deep	News	December 12, 2017		