Long Text Summary Generation by ML Techniques

Project Created By

Subhransu Mishra | 2023AC05489

AIML CS567 | Assignment 1

Problem Statement

Build a summary of long pieces of text, keeping key information content and overall meaning. The summary must represent the most important or relevant information within the text.

Approach 1: Using IDF TF-IDF Along with POS (Part-of-Speech) Tagging

Import of Tools & Libraries

In [298…

```
import requests
from bs4 import BeautifulSoup
import re

import nltk
# NLTK tokenizer models
nltk.download('punkt')
nltk.download('punkt_tab')

from collections import Counter
from collections import defaultdict

from nltk.corpus import stopwords

nltk.download('stopwords')
# Load NLTK's list of stop words
stop_words = set(stopwords.words('english'))

import math
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /Users/I531265/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     /Users/I531265/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/I531265/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/I531265/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[298… True

Obtain Source Data Using Web-Scrapping Technique followed by Sanitization of Data

In [299…

```
# URL to scrape --> We will be getting Text Directly From WikiPedia Site
url = "https://en.wikipedia.org/wiki/Computer_security"
response = requests.get(url)

if response.status_code == 200:
    # Parse the HTML content of the page
    soup = BeautifulSoup(response.content, 'html.parser')

    # Find the main content of the Wikipedia page | This is to ignore additional references and sidebars --> Data M
    content_div = soup.find('div', {'class': 'mw-parser-output'})
```

```
    # Extract all text within the main content
    paragraphs = content_div.find_all('p')
    text_content = []

    for para in paragraphs:
        text = para.get_text()
        # Add a space to separate paragraphs
        if text.strip():
            text_content.append(text.strip())

    # Join the paragraphs
    full_text = "\n\n".join(text_content)

    # Remove reference numberings like [1], [2], etc. --> Data Sanitization
    full_text = re.sub(r'\[\d+\]', '', full_text)

    # Display only the first 100 lines
    lines = full_text.split('\n')  # Split text into lines
    first_30_lines = lines[:30]  # Get the first 100 lines

    # Combine the lines and add ellipsis for truncation
    preview_text = '\n'.join(first_30_lines) + "\n\n... (Content Truncated for Display) ..."

    # Print the preview text for first 30 lines
    print("Preview of Scraped Text first 30 lines:\n")
    print(preview_text)
else:
    print(f"Failed to fetch the webpage. Status code: {response.status_code}")
```

Preview of Scraped Text first 30 lines:

Computer security (also cybersecurity, digital security, or information technology (IT) security) is the protection of computer software, systems and networks from threats that can lead to unauthorized information disclosure, theft or damage to hardware, software, or data, as well as from the disruption or misdirection of the services they provide.

The significance of the field stems from the expanded reliance on computer systems, the Internet, and wireless network standards. Its importance is further amplified by the growth of smart devices, including smartphones, televisions, and the various devices that constitute the Internet of things (IoT). Cybersecurity has emerged as one of the most significant new challenges facing the contemporary world, due to both the complexity of information systems and the societies they support. Security is particularly crucial for systems that govern large-scale systems with far-reaching physical effects, such as power distribution, elections, and finance.

Although many aspects of computer security involve digital security, such as electronic passwords and encryption, physical security measures such as metal locks are still used to prevent unauthorized tampering. IT security is not a perfect subset of information security, therefore does not completely align into the security convergence schema.

A vulnerability refers to a flaw in the structure, execution, functioning, or internal oversight of a computer or system that compromises its security. Most of the vulnerabilities that have been discovered are documented in the Common Vulnerabilities and Exposures (CVE) database. An exploitable vulnerability is one for which at least one working attack or exploit exists. Actors maliciously seeking vulnerabilities are known as threats. Vulnerabilities can be researched, reverse-engineered, hunted, or exploited using automated tools or customized scripts.

Various people or parties are vulnerable to cyber attacks; however, different groups are likely to experience different types of attacks more than others.

In April 2023, the United Kingdom Department for Science, Innovation & Technology released a report on cyber attacks over the previous 12 months. They surveyed 2,263 UK businesses, 1,174 UK registered charities, and 554 education institutions. The research found that "32% of businesses and 24% of charities overall recall any breaches or attacks from the last 12 months." These figures were much higher for "medium businesses (59%), large businesses (69%), and high-income charities with £500,000 or more in annual income (56%)." Yet, although medium or large businesses are more often the victims, since larger companies have generally improved their security over the last decade, small and mid size businesses (SMBs) have also become increasingly vulnerable as they often "do not have advanced tools to defend the business." SMBs are most likely to be affected by malware, ransomware, phishing, man-in-the-middle attacks, and Denial-of Service (DoS) Attacks.

Normal internet users are most likely to be affected by untargeted cyberattacks. These are where attackers indiscriminately target as many devices, services, or users as possible. They do this using techniques that take advantage of the openness of the Internet. These strategies mostly include phishing, ransomware, water holing and scanning.

To secure a computer system, it is important to understand the attacks that can be made against it, and these threats can typically be classified into one of the following categories:

A backdoor in a computer system, a cryptosystem, or an algorithm is any secret method of bypassing normal authentication or security controls. These weaknesses may exist for many reasons, including original design or poor configuration. Due to the nature of backdoors, they are of greater concern to companies and databases as opposed to individuals.

Backdoors may be added by an authorized party to allow some legitimate access or by an attacker for malicious reasons. Criminals often use malware to install backdoors, giving them remote administrative access to a system. Once they have access, cybercriminals can "modify files, steal personal information, install unwanted software, and even take control of the entire computer."

Backdoors can be very hard to detect and are usually discovered by someone who has access to the application source code or intimate knowledge of the operating system of the computer.

Denial-of-service attacks (DoS) are designed to make a machine or network resource unavailable to its intended users. Attackers can deny service to individual victims, such as by deliberately entering a wrong password enough consecutive times to cause the victim's account to be locked, or they may overload the capabilities of a machine or network and block all users at once. While a network attack from a single IP address can be blocked by adding a new firewall rule, many forms of distributed denial-of-service (DDoS) attacks are possible, where the attack comes from a large number of points. In this case, defending against these attacks is much more difficult. Such attacks can originate from the zombie computers of a botnet or from a range of other possible techniques, including distributed reflective denial-of-service (DRDoS), where innocent systems are fooled into sending traffic to the victim. With such attacks, the amplification factor makes the attack easier for the attacker because they have to use little bandwidth themselves. To understand why attackers may carry out these attacks, see the 'attacker motivation' section.

A direct-access attack is when an unauthorized user (an attacker) gains physical access to a computer, most likely to directly copy data from it or steal information. Attackers may also compromise security by making operating system modifications, installing software worms, keyloggers, covert listening devices or using wireless microphones. Even when the system is protected by standard security measures, these may be bypassed by booting another operating system or tool from a CD-ROM or other bootable media. Disk encryption and the Trusted Platform Module standard are designed to prevent these attacks.

Direct service attackers are related in concept to direct memory attacks which allow an attacker to gain direct access to a computer's memory. The attacks "take advantage of a feature of modern computers that allows certain devices, such as external hard drives, graphics cards, or network cards, to access the computer's memory directly."

Eavesdropping is the act of surreptitiously listening to a private computer conversation (communication), usually between hosts on a network. It typically occurs when a user connects to a network where traffic is not secured or encrypted and sends sensitive business data to a colleague, which, when listened to by an attacker, could be exploited. Data transmitted across an open network allows an attacker to exploit a vulnerability and intercept it via various methods.

... (Content Truncated for Display) ...

Tokenize sentences

In [300…
```
# Tokenize the text into sentences
sentences = nltk.sent_tokenize(full_text)

# Display the first 10 tokenized sentences
print("First 10 Tokenized Sentences:\n")
for sentence in sentences[:10]:
    print(sentence)
```

```
First 10 Tokenized Sentences:

Computer security (also cybersecurity, digital security, or information technology (IT) security) is the protection of computer software, systems and networks from threats that can lead to unauthorized information disclosure, theft or damage to hardware, software, or data, as well as from the disruption or misdirection of the services they provide.
The significance of the field stems from the expanded reliance on computer systems, the Internet, and wireless network standards.
Its importance is further amplified by the growth of smart devices, including smartphones, televisions, and the various devices that constitute the Internet of things (IoT).
Cybersecurity has emerged as one of the most significant new challenges facing the contemporary world, due to both the complexity of information systems and the societies they support.
Security is particularly crucial for systems that govern large-scale systems with far-reaching physical effects, such as power distribution, elections, and finance.
Although many aspects of computer security involve digital security, such as electronic passwords and encryption, physical security measures such as metal locks are still used to prevent unauthorized tampering.
IT security is not a perfect subset of information security, therefore does not  completely align into the security convergence schema.
A vulnerability refers to a flaw in the structure, execution, functioning, or internal oversight of a computer or system that compromises its security.
Most of the vulnerabilities that have been discovered are documented in the Common Vulnerabilities and Exposures (CVE) database.
An exploitable vulnerability is one for which at least one working attack or exploit exists.
```

Create frequency matrix of words in each sentence

In [301…
```
# Initialize the frequency matrix
frequency_matrix = {}

for idx, sentence in enumerate(sentences):
    # Tokenize the sentence into words
    words = nltk.word_tokenize(sentence)

    # Convert words to lowercase and filter out non-alphanumeric tokens
    words = [word.lower() for word in words if word.isalnum()]

    # Create a frequency distribution for the sentence
    frequency_matrix[f"Sentence_{idx+1}"] = Counter(words)

print("Preview of Frequency Matrix:\n")
preview_sentences = list(frequency_matrix.items())[:5]  # Limit display to the first 5 sentences
for sentence_id, freq_dist in preview_sentences:
    print(f"{sentence_id}: {dict(freq_dist)}")

# Add truncation message
print("\n... (Content Truncated for Display) ...")
```

```
Preview of Frequency Matrix:

Sentence_1: {'computer': 2, 'security': 3, 'also': 1, 'cybersecurity': 1, 'digital': 1, 'or': 4, 'information': 2, 'technology': 1, 'it': 1, 'is': 1, 'the': 3, 'protection': 1, 'of': 2, 'software': 2, 'systems': 1, 'and': 1, 'networks': 1, 'from': 2, 'threats': 1, 'that': 1, 'can': 1, 'lead': 1, 'to': 2, 'unauthorized': 1, 'disclosure': 1, 'theft': 1, 'damage': 1, 'hardware': 1, 'data': 1, 'as': 2, 'well': 1, 'disruption': 1, 'misdirection': 1, 'services': 1, 'they': 1, 'provide': 1}
Sentence_2: {'the': 4, 'significance': 1, 'of': 1, 'field': 1, 'stems': 1, 'from': 1, 'expanded': 1, 'reliance': 1, 'on': 1, 'computer': 1, 'systems': 1, 'internet': 1, 'and': 1, 'wireless': 1, 'network': 1, 'standards': 1}
Sentence_3: {'its': 1, 'importance': 1, 'is': 1, 'further': 1, 'amplified': 1, 'by': 1, 'the': 3, 'growth': 1, 'of': 2, 'smart': 1, 'devices': 2, 'including': 1, 'smartphones': 1, 'televisions': 1, 'and': 1, 'various': 1, 'that': 1, 'constitute': 1, 'internet': 1, 'things': 1, 'iot': 1}
Sentence_4: {'cybersecurity': 1, 'has': 1, 'emerged': 1, 'as': 1, 'one': 1, 'of': 2, 'the': 4, 'most': 1, 'significant': 1, 'new': 1, 'challenges': 1, 'facing': 1, 'contemporary': 1, 'world': 1, 'due': 1, 'to': 1, 'both': 1, 'complexity': 1, 'information': 1, 'systems': 1, 'and': 1, 'societies': 1, 'they': 1, 'support': 1}
Sentence_5: {'security': 1, 'is': 1, 'particularly': 1, 'crucial': 1, 'for': 1, 'systems': 2, 'that': 1, 'govern': 1, 'with': 1, 'physical': 1, 'effects': 1, 'such': 1, 'as': 1, 'power': 1, 'distribution': 1, 'elections': 1, 'and': 1, 'finance': 1}

... (Content Truncated for Display) ...
```

Calculate Term Frequency and Generate matrix

In [302…
```
# Initialize the TF matrix
tf_matrix = {}

for idx, sentence in enumerate(sentences):
    # Tokenize the sentence into words
```

```
    words = nltk.word_tokenize(sentence)

    # Convert words to lowercase and filter out non-alphanumeric tokens
    words = [word.lower() for word in words if word.isalnum()]

    # Calculate word frequencies in the sentence
    word_freq = Counter(words)
    total_words = len(words)  # Total words in the sentence

    # Calculate term frequency (TF) for each word
    tf_matrix[f"Sentence_{idx+1}"] = {word: freq / total_words for word, freq in word_freq.items()}

print("Preview of Term Frequency (TF) Matrix:\n")
preview_tf = list(tf_matrix.items())[:5]  # Limit to the first 5 sentences
for sentence_id, tf_dict in preview_tf:
    print(f"{sentence_id}: {tf_dict}")

print("\n... (Content Truncated for Display) ...")
```

```
Preview of Term Frequency (TF) Matrix:

Sentence_1: {'computer': 0.04, 'security': 0.06, 'also': 0.02, 'cybersecurity': 0.02, 'digital': 0.02, 'or': 0.08,
'information': 0.04, 'technology': 0.02, 'it': 0.02, 'is': 0.02, 'the': 0.06, 'protection': 0.02, 'of': 0.04, 'softw
are': 0.04, 'systems': 0.02, 'and': 0.02, 'networks': 0.02, 'from': 0.04, 'threats': 0.02, 'that': 0.02, 'can': 0.0
2, 'lead': 0.02, 'to': 0.04, 'unauthorized': 0.02, 'disclosure': 0.02, 'theft': 0.02, 'damage': 0.02, 'hardware': 0.
02, 'data': 0.02, 'as': 0.04, 'well': 0.02, 'disruption': 0.02, 'misdirection': 0.02, 'services': 0.02, 'they': 0.0
2, 'provide': 0.02}
Sentence_2: {'the': 0.21052631578947367, 'significance': 0.05263157894736842, 'of': 0.05263157894736842, 'field': 0.
05263157894736842, 'stems': 0.05263157894736842, 'from': 0.05263157894736842, 'expanded': 0.05263157894736842, 'reli
ance': 0.05263157894736842, 'on': 0.05263157894736842, 'computer': 0.05263157894736842, 'systems': 0.052631578947368
42, 'internet': 0.05263157894736842, 'and': 0.05263157894736842, 'wireless': 0.05263157894736842, 'network': 0.05263
157894736842, 'standards': 0.05263157894736842}
Sentence_3: {'its': 0.04, 'importance': 0.04, 'is': 0.04, 'further': 0.04, 'amplified': 0.04, 'by': 0.04, 'the': 0.1
2, 'growth': 0.04, 'of': 0.08, 'smart': 0.04, 'devices': 0.08, 'including': 0.04, 'smartphones': 0.04, 'television
s': 0.04, 'and': 0.04, 'various': 0.04, 'that': 0.04, 'constitute': 0.04, 'internet': 0.04, 'things': 0.04, 'iot':
0.04}
Sentence_4: {'cybersecurity': 0.03571428571428571, 'has': 0.03571428571428571, 'emerged': 0.03571428571428571, 'as':
0.03571428571428571, 'one': 0.03571428571428571, 'of': 0.07142857142857142, 'the': 0.14285714285714285, 'most': 0.03
571428571428571, 'significant': 0.03571428571428571, 'new': 0.03571428571428571, 'challenges': 0.03571428571428571,
'facing': 0.03571428571428571, 'contemporary': 0.03571428571428571, 'world': 0.03571428571428571, 'due': 0.035714285
71428571, 'to': 0.03571428571428571, 'both': 0.03571428571428571, 'complexity': 0.03571428571428571, 'information':
0.03571428571428571, 'systems': 0.03571428571428571, 'and': 0.03571428571428571, 'societies': 0.03571428571428571,
'they': 0.03571428571428571, 'support': 0.03571428571428571}
Sentence_5: {'security': 0.05263157894736842, 'is': 0.05263157894736842, 'particularly': 0.05263157894736842, 'cruci
al': 0.05263157894736842, 'for': 0.05263157894736842, 'systems': 0.10526315789473684, 'that': 0.05263157894736842,
'govern': 0.05263157894736842, 'with': 0.05263157894736842, 'physical': 0.05263157894736842, 'effects': 0.0526315789
4736842, 'such': 0.05263157894736842, 'as': 0.05263157894736842, 'power': 0.05263157894736842, 'distribution': 0.052
63157894736842, 'elections': 0.05263157894736842, 'and': 0.05263157894736842, 'finance': 0.05263157894736842}

... (Content Truncated for Display) ...
```

Create a table for documents per words POS Technique

In [303…

```
# Initialize a dictionary to count documents per word
word_document_count = defaultdict(int)

# Use the existing frequency matrix
for sentence_id, freq_dist in frequency_matrix.items():
    # Get unique words in the sentence
    unique_words = set(freq_dist.keys())

    # Perform PoS tagging
    pos_tags = nltk.pos_tag(unique_words)

    # Update document count for nouns and proper nouns, excluding stop words
    for word, tag in pos_tags:
        if word not in stop_words and tag in ('NN', 'NNP'):
            word_document_count[word] += 1

# Sort the words by document count in descending order
sorted_words = sorted(word_document_count.items(), key=lambda x: x[1], reverse=True)

# Print the top 30 words in a table format
print(f"{'Word':<20}{'Documents Appeared In':<20}")
print("-" * 40)
for word, count in sorted_words[:30]:  # Display only the top 30
    print(f"{word:<20}{count:<20}")
```

```
Word                Documents Appeared In
----------------------------------------
security            104
computer            61
cybersecurity       41
information         40
access              36
system              35
network             29
cyber               24
internet            22
government          22
technology          20
attack              20
software            18
use                 17
attacker            14
protection          13
secure              13
card                13
risk                13
organization        12
response            12
vulnerability       11
malware             11
credit              11
order               11
world               10
program             10
management          10
infrastructure      10
act                 9
```

Calculate IDF and generate matrix

In [304…

```python
# Total number of sentences (documents)
N = len(frequency_matrix)

# Calculate IDF for nouns and proper nouns only
idf_matrix = {word: math.log(N / count) for word, count in word_document_count.items() if count > 0}

# Sort the IDF matrix by values in descending order
sorted_idf = sorted(idf_matrix.items(), key=lambda x: x[1], reverse=True)

# Print the top 30 IDF words
print(f"{'Word':<20}{'IDF':<10}")
print("-" * 30)
for word, idf in sorted_idf[:30]:  # Display only the top 30
    print(f"{word:<20}{idf:<10.4f}")
```

```
Word                IDF
------------------------------
lead                6.1527
disruption          6.1527
misdirection        6.1527
wireless            6.1527
reliance            6.1527
significance        6.1527
importance          6.1527
complexity          6.1527
metal               6.1527
perfect             6.1527
therefore           6.1527
schema              6.1527
subset              6.1527
flaw                6.1527
execution           6.1527
cve                 6.1527
exploitable         6.1527
income              6.1527
midsize             6.1527
decade              6.1527
openness            6.1527
method              6.1527
algorithm           6.1527
cryptosystem        6.1527
configuration       6.1527
install             6.1527
modify              6.1527
steal               6.1527
intimate            6.1527
resource            6.1527
```

Calculate TF-IDF and generate matrix

In [305…
```python
# Initialize the TF-IDF matrix
tf_idf_matrix = {}

# Calculate TF-IDF for each word in each sentence
for sentence_id, word_freq in frequency_matrix.items():
    tf_idf_matrix[sentence_id] = {}
    for word, tf in word_freq.items():
        if word in idf_matrix:  # Only calculate for words in IDF matrix
            tf_idf_matrix[sentence_id][word] = tf * idf_matrix[word]

# Combine all TF-IDF scores into a single list for sorting and ranking
tf_idf_scores = []
for sentence_id, word_scores in tf_idf_matrix.items():
    for word, tf_idf in word_scores.items():
        tf_idf_scores.append((word, tf_idf, sentence_id))

# Sort the scores by TF-IDF value in descending order
sorted_tf_idf = sorted(tf_idf_scores, key=lambda x: x[1], reverse=True)

# Print the top 30 TF-IDF scores
print(f"{'Word':<20}{'TF-IDF':<10}{'Sentence':<10}")
print("-" * 40)
for word, tf_idf, sentence_id in sorted_tf_idf[:30]:  # Display only the top 30
    print(f"{word:<20}{tf_idf:<10.4f}{sentence_id:<10}")
```

```
Word                TF-IDF    Sentence
----------------------------------------
dod                 36.9164   Sentence_410
loss                18.1732   Sentence_218
care                16.3788   Sentence_229
hygiene             15.1624   Sentence_177
team                13.6299   Sentence_89
energy              13.0829   Sentence_270
data                12.3055   Sentence_49
architecture        12.3055   Sentence_117
data                12.3055   Sentence_124
something           12.3055   Sentence_146
inoculation         12.3055   Sentence_152
node                12.3055   Sentence_178
layer               12.3055   Sentence_193
data                12.3055   Sentence_266
data                12.3055   Sentence_331
data                12.3055   Sentence_354
disclosure          12.3055   Sentence_402
ware                12.3055   Sentence_443
design              12.2199   Sentence_111
control             12.2199   Sentence_166
health              12.2199   Sentence_229
cyber               11.8987   Sentence_175
memory              10.9192   Sentence_42
effect              10.9192   Sentence_81
spoofing            10.9192   Sentence_92
html                10.9192   Sentence_96
list                10.9192   Sentence_164
hospital            10.9192   Sentence_266
minister            10.9192   Sentence_372
policy              10.9192   Sentence_374
```

Score the sentences

In [306…
```python
sentence_scores = {}

# Calculate the score for each sentence
for sentence_id, word_scores in tf_idf_matrix.items():
    # Sum all TF-IDF scores of words in the sentence
    sentence_scores[sentence_id] = sum(word_scores.values())

# Sort the sentences by their scores in descending order
sorted_sentences = sorted(sentence_scores.items(), key=lambda x: x[1], reverse=True)

# Print the top 5 scored sentences
print(f"{'Sentence':<15}{'Score':<10}")
print("-" * 30)
for sentence_id, score in sorted_sentences[:5]:  # Display top 5 sentences
    print(f"{sentence_id:<15}{score:<10.4f}")
```

```
Sentence       Score
------------------------------
Sentence_410   106.8150
Sentence_382   102.0011
Sentence_402   96.2920
Sentence_313   90.6078
Sentence_391   90.3172
```

Set the threshold - Currently At 70 Percent

In [307…
```python
# Calculate thresholds
mean_threshold = sum(sentence_scores.values()) / len(sentence_scores)
max_threshold = max(sentence_scores.values()) * 0.7

print(f"Mean Threshold: {mean_threshold:.4f}")
print(f"70% of Max Threshold: {max_threshold:.4f}")
```

```
Mean Threshold: 29.0885
70% of Max Threshold: 74.7705
```

Generate the summary

In [308…
```python
# Select sentences above the thresholds
selected_sentences = []
selected_words = set()

for sentence_id, score in sorted(sentence_scores.items(), key=lambda x: x[1], reverse=True):
    sentence_index = int(sentence_id.split('_')[1]) - 1
    sentence_text = sentences[sentence_index]
    words_in_sentence = set(sentence_text.lower().split())

    # Check if the sentence score meets either threshold
    if score > max_threshold:
        # Avoid redundancy by checking word overlap
        if len(selected_words.intersection(words_in_sentence)) / len(words_in_sentence) < 0.5:
            selected_sentences.append((sentence_index, sentence_text))
            selected_words.update(words_in_sentence)

# Sort selected sentences by their original order
selected_sentences.sort(key=lambda x: x[0])

# Combine selected sentences into a summary
summary = " ".join([s[1] for s in selected_sentences])

print("Generated Refined Summary:\n")
print(summary)
```

```
Generated Refined Summary:

Computer security (also cybersecurity, digital security, or information technology (IT) security) is the protection of computer software, systems and networks from threats that can lead to unauthorized information disclosure, theft or damage to hardware, software, or data, as well as from the disruption or misdirection of the services they provide. As the human component of cyber risk is particularly relevant in determining the global cyber risk an organization is facing, security awareness training, at all levels, not only provides formal compliance with regulatory and industry mandates but is considered essential in reducing cyber risk and protecting individuals and companies from the great majority of cyber threats. Websites and apps that accept or store credit card numbers, brokerage accounts, and bank account information are also prominent hacking targets, because of the potential for immediate financial gain from transferring money, making purchases, or selling the information on the black market. The most common web technologies for improving security between browsers and websites are named SSL (Secure Sockets Layer), and its successor TLS (Transport Layer Security), identity management and authentication services, and domain name services allow companies and consumers to engage in secure communications and commerce. However, as these devices serve as potential access points to the hospital network, security threats increase, and hospitals have to introduce adequate security measures which, for example, comply with the Health Insurance Portability and Accountability Act (HIPAA). The intruders were able to obtain classified files, such as air tasking order systems data and furthermore able to penetrate connected networks of National Aeronautics and Space Administration's Goddard Space Flight Center, Wright-Patterson Air Force Base, some Defense contractors, and other private sector organizations, by posing as a trusted Rome center user. In this policy, the US says it will: Protect the country by keeping networks, systems, functions, and data safe; Promote American wealth by building a strong digital economy and encouraging strong domestic innovation; Peace and safety should be kept by making it easier for the US to stop people from using computer tools for bad things, working with friends and partners to do this; and increase the United States' impact around the world to support the main ideas behind an open, safe, reliable, and compatible Internet. In response to the Colonial Pipeline ransomware attack President Joe Biden signed Executive Order 14028 on May 12, 2021, to increase software security standards for sales to the government, tighten detection and security on existing systems, improve information sharing and training, establish a Cyber Safety Review Board, and improve incident response. In 2017, CCIPS published A Framework for a Vulnerability Disclosure Program for Online Systems to help organizations "clearly describe authorized vulnerability disclosure and discovery conduct, thereby substantially reducing the likelihood that such described activities will result in a civil or criminal violation of law under the Computer Fraud and Abuse Act (18 U.S.C. The US Department of Defense (DoD) issued DoD Directive 8570 in 2004, supplemented by DoD Directive 8140, requiring all DoD employees and all DoD contract personnel involved in information assurance roles and activities to earn and maintain various industry Information Technology (IT) certifications in an effort to ensure that all DoD personnel involved in network infrastructure defense have minimum levels of IT industry recognized knowledge, skills and abilities (KSA).
```

Approach 2: Using TF-IDF Vectorization Along with PageRanking Algorithm and Cosine Similarity

In [309…
```python
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import networkx as nx

# Step 1: Tokenize the text into sentences
sentences = nltk.sent_tokenize(full_text)

# Step 2: Generate the TF-IDF matrix for sentence similarity
vectorizer = TfidfVectorizer().fit_transform(sentences)
```

```
similarity_matrix = cosine_similarity(vectorizer)

# Step 3: Build the sentence graph
sentence_graph = nx.from_numpy_array(similarity_matrix)

# Step 4: Rank sentences using the PageRank algorithm
scores = nx.pagerank(sentence_graph)

# Step 5: Sort sentences by score
ranked_sentences = sorted(((scores[i], s) for i, s in enumerate(sentences)), reverse=True)

# Step 6: Select the top N sentences for the summary
N = 10  # Number of sentences in the summary
summary = " ".join([s for _, s in ranked_sentences[:N]])

# Print the improved summary
print("Generated Summary:\n")
print(summary)
```

```
Generated Summary:

Computer security (also cybersecurity, digital security, or information technology (IT) security) is the protection
of computer software, systems and networks from threats that can lead to unauthorized information disclosure, theft
or damage to hardware, software, or data, as well as from the disruption or misdirection of the services they provid
e. In practice, the role of a security architect would be to ensure the structure of a system reinforces the securit
y of the system, and that new changes are safe and meet the security requirements of the organization. To secure a c
omputer system, it is important to understand the attacks that can be made against it, and these threats can typical
ly be classified into one of the following categories:
A backdoor in a computer system, a cryptosystem, or an algorithm is any secret method of bypassing normal authentica
tion or security controls. The role of the government is to make regulations to force companies and organizations to
protect their systems, infrastructure and information from any cyberattacks, but also to protect its own national in
frastructure such as the national power-grid. Spoofing is an act of pretending to be a valid entity through the fals
ification of data (such as an IP address or username), in order to gain access to information or resources that one
is otherwise unauthorized to obtain. The National Cyber Security Policy 2013 is a policy framework by the Ministry o
f Electronics and Information Technology (MeitY) which aims to protect the public and private infrastructure from cy
berattacks, and safeguard "information, such as personal information (of web users), financial and banking informati
on and sovereign data". In Side-channel attack scenarios, the attacker would gather such information about a system
or network to guess its internal state and as a result access the information which is assumed by the victim to be s
ecure. In this policy, the US says it will: Protect the country by keeping networks, systems, functions, and data sa
fe; Promote American wealth by building a strong digital economy and encouraging strong domestic innovation; Peace a
nd safety should be kept by making it easier for the US to stop people from using computer tools for bad things, wor
king with friends and partners to do this; and increase the United States' impact around the world to support the ma
in ideas behind an open, safe, reliable, and compatible Internet. In the United Kingdom, a nationwide set of cyberse
curity forums, known as the U.K Cyber Security Forum, were established supported by the Government's cybersecurity s
trategy in order to encourage start-ups and innovation and to address the skills gap identified by the U.K Governmen
t. Backdoors can be very hard to detect and are usually discovered by someone who has access to the application sour
ce code or intimate knowledge of the operating system of the computer.
```

In [ ]: