FAKE JOB OFFER TWEETS ANALYSIS AND SUMMARIZATION

OBJECTIVE

The primary goal of this script is to analyze Twitter data related to "fake job offers" and "job scams," focusing on identifying tweets that mention individuals as victims. After gathering these tweets, the script processes and summarizes the content using Natural Language Processing (NLP) techniques. The summarized tweets are evaluated using ROUGE scores to assess the quality of the generated summaries. The script performs multiple steps: data collection, cleaning, victim identification, summarization, and evaluation.

DATA COLLECTION (TWEET FETCHING)

The script begins by collecting tweets from Twitter using the tweepy API. It searches for tweets related to specific keywords such as:

- "fake job offer"
- "job scam"

The script ensures that only tweets written in English are fetched, and retweets are excluded. Depending on availability, pagination ensures that up to 100 tweets can be retrieved per request, allowing for 100 tweets or more.

The collected tweets are stored in a data frame and then saved to a CSV file for further processing.

DATA CLEANING

Data cleaning is an essential preprocessing step that involves removing unnecessary or noisy data. In this case, the text of the tweets is cleaned to remove:

- URLs
- Mentions (e.g., @username)
- Hashtags (e.g., #jobfraud)
- Punctuation marks and special characters

The script uses regular expressions to standardize the raw tweet text. The text is also converted to lowercase to ensure consistency. This step helps focus on the meaningful content of the tweets without distraction from irrelevant components.

HANDLING MISSING VALUES

- Any rows with missing cleaned text are dropped to ensure the remaining data is valid.

HANDLING DUPLICATES

- The script ensures that duplicate tweets are removed based on the cleaned text to avoid redundancy in the dataset.

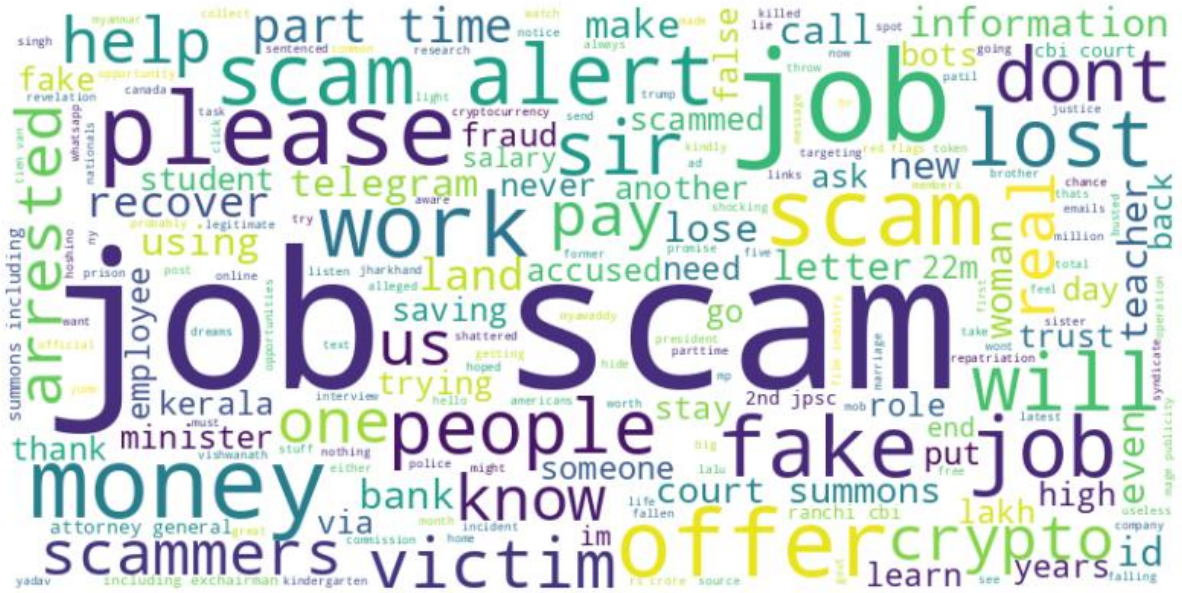
VICTIM IDENTIFICATION

The next step involves identifying tweets that mention victims of job scams. The script uses a set of predefined keywords such as:

- "scammed"
- "fraud"
- "lost money"
- "victim"
- "scam"

Each tweet is checked for the presence of these keywords, and if any of them are found, the tweet is flagged as potentially related to a victim. The filtered tweets are stored in a separate DataFrame (victims_df), making it easy to focus on the relevant tweets for summarization.

WORD CLOUD GENERATION



A word cloud is generated from the victim-related tweets to visually represent the most frequently occurring words. This visualization helps identify recurring themes or keywords, such as scam tactics or common phrases used by victims. The WordCloud class from the wordcloud library is used to create the word cloud, and matplotlib is employed to display the image.

KEY SENTENCE EXTRACTION (Using spaCy)

Before summarizing the tweets, key sentences are extracted using spaCy, a powerful NLP library. The goal is to focus on the most important parts of each tweet, specifically sentences that mention "PERSON" entities (i.e., individuals or victims). This step enhances the relevance of the summaries by ensuring that the extracted sentences are meaningful and provide useful context.

SUMMARIZATION (Using BART Transformer)

Summarization involves generating shorter versions of the original text while preserving its core meaning. The script uses the BART transformer model from Hugging Face to perform text summarization. BART is a pre-trained model capable of both text generation and understanding.

The script dynamically adjusts the length of the generated summary by setting max_length based on the length of the input tweet. If the tweet is too short, a warning message is returned stating that the text is too short for summarization.

CHALLENGES IN SUMMARIZATION

- Half Summaries: Some summaries may be cut off or truncated if the model's output length exceeds the specified max_length. This may result in incomplete summaries, leaving out key details from the original tweet.
- Inconsistent Summarization: Some summaries are only partial or inaccurate, potentially because the summarization model does not always capture the most essential information from longer or more complex tweets.

BATCH PROCESSING OF SUMMARIZATION

The summarization process is performed in batches to enhance efficiency, processing groups of tweets together (e.g., 50 tweets per batch). This reduces the overall time to summarize a large number of tweets and helps maintain a manageable flow of operations.

EVALUATION USING ROUGE SCORE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used to evaluate the quality of generated summaries. It compares the overlap between the generated summary and a reference (or ground truth) text, the original tweet content. The script computes three types of ROUGE scores:

- ROUGE-1: Measures the overlap of unigrams (individual words) between the reference and generated summary.
- ROUGE-2: Measures the overlap of bigrams (pairs of consecutive words).
- ROUGE-L: Measures the longest common subsequence (LCS), a sequence that appears in both texts in the same order.

INTERPRETATION OF ROUGE SCORES

The ROUGE scores provide a quantitative assessment of how well the summarization model has performed. A higher score indicates that the generated summary is closer to the reference text, meaning the model has better preserved the core information.

For example, the ROUGE scores in this case were:

- ROUGE-1: 0.7806
- ROUGE-2: 0.7441
- ROUGE-L: 0.7491

These scores suggest that the summarizer is fairly effective in capturing key information from the original tweet. Specifically:

- ROUGE-1 (0.7806) indicates that the unigrams (individual words) in the summary have a high overlap with the original text, suggesting good overall coverage of the content.
- ROUGE-2 (0.7441) shows that the summarizer has a relatively high degree of overlap in terms of bigrams (pairs of words), indicating that it captures more nuanced patterns in the text.
- ROUGE-L (0.7491) indicates a moderate degree of overlap in terms of the longest common subsequence, which suggests that the summary retains much of the important sequence of words in the original tweet.

The higher these scores, the more accurate and relevant the generated summaries are in comparison to the original text.

OUTPUT

The final output consists of:

1. Summarized Tweets: A CSV file (summarized.csv) containing the victim-related tweets along with their corresponding summaries.
2. ROUGE Evaluation: The average ROUGE scores for evaluating the summarization quality.

While the ROUGE scores provide a solid indication of summary quality, it is essential to note that even high ROUGE scores may not always correlate with perfect summarization. Human judgment and contextual analysis are crucial for a more comprehensive evaluation.

CONCLUSION

This script provides an end-to-end pipeline for processing, analyzing, and summarizing tweets related to fake job offers and job scams. The steps involved include:

1. Data Collection: Gathering tweets using the tweepy API.
2. Data Cleaning: Removing irrelevant elements and handling missing or duplicate values.
3. Victim Identification: Filtering tweets mentioning job scam victims.
4. Text Summarization: Using the BART model to generate concise summaries of the tweets.
5. Evaluation: Using ROUGE scores to assess the summarization quality.

AREAS FOR IMPROVEMENT

- Handling Missing Values: Some missing cleaned text values still have summaries associated with them, which should be addressed to ensure consistency.
- Partial Summaries: Some generated summaries are incomplete due to truncation, particularly when the input tweet is too long. Reducing max_length could help resolve this issue but might also lead to shorter summaries than desired.