

```
In [18]: #Installing the required packages
#!pip install transformers
#!pip install nltk
#!pip install sentencepiece
```

For statement 1

```
In [29]: import spacy
from collections import Counter

nlp = spacy.load("en_core_web_sm")
text = "Definitely share your feedback in the comment section."
```

```
In [19]: from transformers import MarianTokenizer, MarianMTModel

# Load the pre-trained model and tokenizer
model_name = "Helsinki-NLP/opus-mt-en-hi"
tokenizer = MarianTokenizer.from_pretrained(model_name)
model = MarianMTModel.from_pretrained(model_name)

def english_to_hinglish(text):
    # Tokenize the input text
    input_ids = tokenizer.encode(text, return_tensors="pt")

    # Translate to Hinglish
    translation_ids = model.generate(input_ids)

    # Decode the translated text
    translated_text = tokenizer.decode(translation_ids[0], skip_special_tokens=True)

    return translated_text

# Example usage
#english_text = "I had about a 30 minute demo just using this new headset"
hinglish_text = english_to_hinglish(text)
print(hinglish_text)

#pip install sentencepiece

/usr/local/lib/python3.10/dist-packages/transformers/models/arian/tokenization_arian.p
y:194: UserWarning: Recommended: pip install sacremoses.
  warnings.warn("Recommended: pip install sacremoses.")
टिप्पणी खण्ड में अपनी प्रतिक्रिया को नोटित हो साझा करें ।
```

```
In [26]: import spacy
from collections import Counter

nlp = spacy.load("en_core_web_sm")

# Tokenize and preprocess the text
doc = nlp(text)
tokens = [token.text.lower() for token in doc if not token.is_stop and not token.is_punc]

# Calculate word frequencies
word_freq = Counter(tokens)

# Select top keywords based on frequency (you can adjust the number)
top_keywords = [keyword for keyword, _ in word_freq.most_common(5)]

print("Top Keywords:", top_keywords)
```

```

# Assuming you have your list of tokens and top_keywords defined earlier
# and an english_to_hinglish function to perform translation

# Original text
original_text = text

# Tokenize the original text
tokens = original_text.split()

# Initialize an empty list to store the translated tokens
translated_tokens = []

for token in tokens:
    if token in top_keywords:
        # If the token is in the top_keywords list, keep it as is
        translated_tokens.append(token)
    else:
        # Translate the token to Hinglish and add it to the translated_tokens list
        translated_token = english_to_hinglish(token)
        translated_tokens.append(translated_token)

# Join the translated tokens to form the translated text
translated_text = ' '.join(translated_tokens)

print("Text: ",text)
print("Translated text in hindi: ",hinglish_text)
print("Required text: ",translated_text)

```

Top Keywords: ['definitely', 'share', 'feedback', 'comment', 'section']
Text: Definitely share your feedback in the comment section.
Translated text in hindi: टिप्पणी खण्ड में अपनी प्रतिक्रिया को निश्चित ही साझा करें ।
Required text: निश्चित रूप से share आपका feedback में वह comment खंड.

In [26]:

For statement 2

In [27]:

```

#for 2nd statement

text1 = "So even if it's a big video, I will clearly mention all the products."

doc = nlp(text1)
tokens = [token.text.lower() for token in doc if not token.is_stop and not token.is_punc]

# Calculate word frequencies
word_freq = Counter(tokens)

# Select top keywords based on frequency (you can adjust the number)
top_keywords = [keyword for keyword, _ in word_freq.most_common(5)]

print("Top Keywords:", top_keywords)

original_text = text1

# Tokenize the original text
tokens = original_text.split()

# Initialize an empty list to store the translated tokens
translated_tokens = []

for token in tokens:
    if token in top_keywords:
        # If the token is in the top_keywords list, keep it as is
        translated_tokens.append(token)

```

```

    else:
        # Translate the token to Hinglish and add it to the translated_tokens list
        translated_token = english_to_hinglish(token)
        translated_tokens.append(translated_token)

# Join the translated tokens to form the translated text
translated_text = ' '.join(translated_tokens)

print("Text: ",text1)
print("Translated text in hindi: ",english_to_hinglish(text1))
print("Required text: ",translated_text)

```

Top Keywords: ['big', 'video', 'clearly', 'mention', 'products']

Text: So even if it's a big video, I will clearly mention all the products.

Translated text in hindi: तो यह एक बड़ा वीडियो है, तो भी मैं स्पष्ट रूप से सभी उत्पादों का उल्लेख करेंगे।

Required text: तो यहां तक कि यदि यह है एक big वीडियो, आई होगा clearly mention सभी वह उत्पाद.

For statement 3

```

In [28]: #for 3rd statement
text2 = "I was waiting for my bag."

doc = nlp(text2)
tokens = [token.text.lower() for token in doc if not token.is_stop and not token.is_punc]

# Calculate word frequencies
word_freq = Counter(tokens)

# Select top keywords based on frequency (you can adjust the number)
top_keywords = [keyword for keyword, _ in word_freq.most_common(5)]

print("Top Keywords:", top_keywords)

original_text = text2

# Tokenize the original text
tokens = original_text.split()

# Initialize an empty list to store the translated tokens
translated_tokens = []

for token in tokens:
    if token in top_keywords:
        # If the token is in the top keywords list, keep it as is
        translated_tokens.append(token)
    else:
        # Translate the token to Hinglish and add it to the translated_tokens list
        translated_token = english_to_hinglish(token)
        translated_tokens.append(translated_token)

# Join the translated tokens to form the translated text
translated_text = ' '.join(translated_tokens)

print("Text: ",text2)
print("Translated text in hindi: ",english_to_hinglish(text2))
print("Required text: ",translated_text)

```

Top Keywords: ['waiting', 'bag']

Text: I was waiting for my bag.

Translated text in hindi: मैं अपने बैग के लिए इंतजार कर रहा था.

Required text: आई था waiting के लिए मेरा बैग.

In [7]:

In [7]:

In [30]:

In [2]:

In [1]:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

New Section

In []:

In []:

In []:

In []: