

## ✓ Install all the Libraries

```
# @title Install all the Libraries
!pip install googletrans==3.1.0a0
!pip install pymupdf
!pip install -q transformers==4.43.1 einops accelerate langchain bitsandbytes sentencepiece langchain_community langchain_huggingfac
!pip install spacy
!pip install Together
!pip install deep-translator
!pip install -q transformers==4.43.1 einops accelerate langchain bitsandbytes sentencepiece langchain_community langchain_huggingfac
!pip install spacy
```

 [Show hidden output](#)

## ✓ API Implementation

### ✓ Load the API Key and Model

```
# @title Load the API Key and Model
import os
from together import Together
from deep_translator import GoogleTranslator
from google.colab import userdata
import torch
import spacy
import fitz
import time
import ipywidgets as widgets
from IPython.display import Markdown

Lkey = userdata.get('Llama_Key')
client = Together(api_key=Lkey)
model = "meta-llama/Meta-Llama-3.1-8B-Instruct-Turbo"
```

### ✓ All the functions

```
# @title All the functions
def printmd(string):
    display(Markdown(string))

def generate_study_plan(days, syllabus):
    input_prompt = f"Create a detailed roadmap to complete the following syllabus in {days} days:\n{syllabus}, not before the days i
    prompt = """
    You are an expert in making study plans. You are an academic assistant who helps users plan their studies.
    You have to make a detailed study plan. Share the study resources for each topic, the number of focused hours to study.
    Dont give me any outline for the study plan, I need the detailed brief study plan with covering the important topics first and t
    Cover the Important topics from the syllabus first and then cover rest of the things.
    Dont respond with anything else other than the study plan. When you tell the day along with the day also tell what should be dor
    Share some important resources for each topic like youtube videos with Youtube video title, books, etc. Dont share if the resour
    """

    messages = [
        {"role": "system", "content": prompt},
        {"role": "user", "content": input_prompt},
    ]
    response = client.chat.completions.create(
        model=model,
        messages=messages,
        max_tokens=4096,
        temperature=0.6,
        top_p=0.7,
        top_k=50,
        repetition_penalty=1,
        stop=["<|eot_id|>", "<|eom_id|>"],
        stream=False
    )
    return (response.choices[0].message.content)

# Translate the plan to user's preferred language
def translate_study_plan(plan_text, target_lang):
    translator = GoogleTranslator(source='auto', target=target_lang)
    translated_chunks = []
    # if target_lang == 'en':
    #     chunks = split_text(plan_text, max_length=5000)
```

```

#         # Translate each chunk and store the result
#         for chunk in chunks:
#             translated_chunks.append(translator.translate(chunk))
#         # Join the translated chunks back into a single string
#         translated_plan = " ".join(translated_chunks)
#         return translated_plan

chunks = split_text(plan_text, max_length=1000) # Try smaller chunks
for chunk in chunks:
    attempts = 3
    for attempt in range(attempts):
        try:
            translated_chunks.append(translator.translate(chunk))
            time.sleep(1) # Short sleep to avoid rate limit
            break # Exit loop on success
        except RequestError as e:
            print(f"Attempt {attempt + 1} failed: {e}")
            time.sleep(2 ** attempt) # Exponential backoff
    else:
        print("All attempts failed for chunk:", chunk) # Log failure

# Join the translated chunks back into a single string
translated_plan = " ".join(translated_chunks)
return translated_plan

# Read PDF content using PyMuPDF
def read_pdf_input(pdf_path):
    doc = fitz.open(pdf_path)
    text = ""
    for page_num in range(len(doc)):
        page = doc.load_page(page_num) # Load each page
        text += page.get_text("text") # Extract text from page
    return text

def read_text_input(text_path):
    if os.path.exists(text_path):
        with open(text_path, 'r') as file:
            return file.read()
    return text_path

def extract_topics(text):
    nlp = spacy.load("en_core_web_sm")
    doc = nlp(text)
    topics = set()
    for ent in doc.ents:
        if ent.label_ in {"ORG", "GPE", "PERSON", "WORK_OF_ART", "EVENT"}:
            topics.add(ent.text)
    return list(topics)

def generate_summary(syllabus):
    input_prompt = f"Generate a simple summary for this syllabus: {syllabus}"
    prompt = """
    You are an expert in generating summaries and getting the important topics for the user syllabus.
    Be friendly with the user, suggest the summary with proper points.
    Dont respond with anything else than the summary.
    """
    messages = [
        {"role": "system", "content": prompt},
        {"role": "user", "content": input_prompt},
    ]
    response = client.chat.completions.create(
        model=model,
        messages=messages,
        max_tokens=512,
        temperature=0.6,
        top_p=0.7,
        top_k=50,
        repetition_penalty=1,
        stop=["<|eot_id|>", "<|eom_id|>"],
        stream=False
    )
    return (response.choices[0].message.content)

def split_text(text, max_length):
    chunks = []
    # print("slipping")
    while len(text) > max_length:

```

```
        # Find the last space within the max_length to avoid splitting words
        split_point = text[:max_length].rfind(' ')
        chunks.append(text[:split_point])
        text = text[split_point:]
    chunks.append(text)
    return chunks
```

```
import ipywidgets as widgets
```

```
def get_languages():
    languages = {
        'English': 'en',
        'Hindi': 'hi',
        'Marathi': 'mr',
        'Bengali': 'bn',
        'Gujarati': 'gu',
        'Tamil': 'ta',
        'Telugu': 'te',
        'Kannada': 'kn',
        'Malayalam': 'ml',
        'Punjabi': 'pa',
        'Odia': 'or',
        'Assamese': 'as',
        'Nepali': 'ne',
        'Urdu': 'ur',
        'Sindhi': 'sd',
        'Bhojpuri': 'bho',
        'Manipuri': 'mni',
        'Spanish': 'es',
        'French': 'fr',
        'German': 'de',
        'Italian': 'it',
        'Japanese': 'ja',
        'Korean': 'ko',
        'Russian': 'ru',
    }

    # Create a dropdown for input language selection
    input_langauge = widgets.Dropdown(
        options=languages.keys(),
        description='Select Input Language:',
    )


    # Create a dropdown for input language selection
    output_langauge = widgets.Dropdown(
        options=languages.keys(),
        description='Select Output Language:',
    )

    display(input_langauge)
    display(output_langauge)
    return languages, input_langauge, output_langauge

def get_code(languages, input_langauge, output_langauge):
    for key, value in languages.items():
        if key == input_langauge.value:
            input_lang = value
        if key == output_langauge.value:
            output_lang = value
    return input_lang, output_lang
```


▼ Main working

```
languages, input_langauge, output_langauge = get_languages()
```



Select Inpu...


English



Select Out...

Sindhi

```
input_code, output_code = get_code(languages, input_langauge, output_langauge)
print(input_code, " to ", output_code)
```



en

to

sd

```

# Get syllabus input from the user
# print("Enter syllabus (text or PDF path):")
print(translate_study_plan("Enter syllabus (text or PDF path):", input_code))
input_syllabus = input().strip()

# Handle PDF input if provided
syllabus = ""
try:
    if input_syllabus.endswith('.pdf'):
        syllabus = read_pdf_input(input_syllabus)
    else:
        syllabus = read_text_input(input_syllabus)

except FileNotFoundError:
    printmd("***Error:** The specified file was not found. Please check the path and try again.")
    raise # Optional: re-raise the exception if you want to stop the program

except Exception as e:
    printmd(f"***Error:** An unexpected error: {e}")
    raise # Optional: re-raise the exception if you want to stop the program

# Convert syllabus text to English
if syllabus:
    syllabus_text = translate_study_plan(syllabus, 'en')
    printmd("---")
    printmd("### Syllabus Content")
    printmd(syllabus_text)
    printmd("---")

# Get the number of days to complete the syllabus
# print("Enter the number of days to complete the syllabus (or press Enter to skip):")
print(translate_study_plan("Enter the number of days to complete the syllabus (or press Enter to skip):", input_code))
days_input = input().strip()
days = int(days_input) if days_input else 7 # Default to 7 days

printmd("---")
printmd("### Summary of the syllabus:")
summary = generate_summary(syllabus_text)

# Translate the summary to the user's preferred language
translated_summ = translate_study_plan(summary, output_code)
printmd(translated_summ)
printmd("---")

printmd("### Study Plan:")
study_plan = generate_study_plan(days, syllabus_text)

# Output the translated study plan
translated_plan = translate_study_plan(study_plan, output_code)
printmd(translated_plan)
printmd("---")

printmd("### Key Topics")
array_topics = extract_topics(syllabus_text)
key_topics = ", ".join(array_topics)
# printmd(key_topics)

# Translate key topics to the user's preferred language
translated_topics = translate_study_plan(key_topics, output_code)
printmd(translated_topics)
printmd("---")

```

↔ Enter syllabus (text or PDF path):  
/content/test.pdf

## Syllabus Content

### Module Contents

#### 1.0 Introduction

Origin & History of NLP; Language, Knowledge and Grammar in language processing

Stages in NLP; Ambiguities and its types in English and Indian Regional Languages; Challenges of NLP; Applications of NLP 2.0 Word Level Analysis

2.1 Basic Terms: Tokenization, Stemming, Lemmatization; Survey of English Morphology, Inflectional Morphology, Derivational Morphology; 2.2 Morphological

Models: Dictionary lookup, finite state morphology; Morphological parsing with FST (Finite State Transducer) 2.3 Grams and its variation: Bigram, Trigram;

Simple (Unsmoothed) N-grams; N- gram Sensitivity to the Training Corpus; Unknown Words: Open versus closed vocabulary tasks; Evaluating N-grams:

Perplexity; Smoothing: Laplace Smoothing

#### 3.0 Syntax analysis

3.1 Part-Of-Speech tagging(POS); Tag set for English (Upenn Treebank); Difficulties /Challenges in POS tagging; Rule-based, Stochastic and Transformation-

based tagging; 3.2 Generative Model: Hidden Markov Model /HMM Viterbi for POS tagging; Issues in HMM POS tagging; Discriminative Model: Maximum

Entropy model, Conditional random Field (CRF); Parsers

4.0 Semantic Analysis 4.1 Introduction, meaning representation; Lexical Semantics; Corpus study; Study of Various language dictionaries like WorldNet

Relations among lexemes & their senses –Homonymy, Polysemy, Synonymy, Hyponymy;Semantic Ambiguity; Word Sense Disambiguation (WSD);Knowledge

based approach( Lesk’s Algorithm)

#### 5.0 Pragmatics

Discourse: Reference Resolution, Reference Phenomena, Syntactic & Semantic constraint on coherence, Anaphora

#### 6.0 Generative AI , Prompt Engineering and Large Language Models

6.1 Introduction to Generative AI , Types of Generative AI Models (Variational Auto Encoders, Generative Adversarial Networks), Advantages and limitations of

Generative AI,ChatGPT. 6.2 Prompt Engineering prompts for LLM interaction, Prompt Templates,Techniques for crafting clear, concise, and informative

prompts, Exploring advanced prompt engineering strategies (zero-shot learning, few-shot learning) , and case studies: successful applications of prompt

engineering. 6.3 LLM architecture (transformers), understanding pre-training and fine-tuning of LLM, Popular LLM examples (GPT-3),Exploring LLM

capabilities: text generation, translation, question answering, code generation etc ,Langchain, Setting up Environment LangChain and LLM, Meta Llama2 ,

Google PaLM2 LLM

Enter the number of days to complete the syllabus (or press Enter to skip):

6

## Summary of the syllabus:

:هتي نصاب جو خلاصو آهي

:ماڻل مواد

### 1. اين ايل پي جو تعارف

- جو اصل ۽ تاريخ NLP
- ٻولي پروسيسنگ ۾ ٻولي، علم، ۽ گرامر
- اين ايل پي ۾ اسٽيج، ابهام ۽ ان جا قسم
- اين ايل پي جون درخواستون

### 2. لفظ جي سطح جو تجزيو

- ٽوڪنائيزيشن، اسٽيمنگ، ۽ لميٽائيزيشن
- مورفولوجي ماڊلز: ڊڪشنري لوڪ اپ ۽ فائنٽ اسٽيٽ مورفولوجي
- ٽيڪنڪ Smoothing ۽ N-grams (Bigram ۽ Trigram)

### 3. نحو جو تجزيو

- ۽ ان جا چٽلينج (POS) پارٽ-آف-اسپيج ٽيگنگ
- ضابطي جي بنياد تي، اسٽوچسٽڪ، ۽ تبديلي جي بنياد تي ٽيگنگ
- (CRF) ۽ مشروط ريٽڊم فيلڊ (HMM) لڪيل مارڪوف ماڊل

### 4. \*\*معنوي تجزيو \*\*

- ليڪسيڪل سيميٽڪس ۽ ڪورپس مطالعي جو تعارف
- ۽ علم تي ٻڌل انداز (WSD) لفظ احساس تڪرار

### 5. عمليات

- بحث ۽ حوالو قرارداد
- هم آهنگي تي نحوي ۽ اصطلاحي پابنديون

### 6. Generative AI ۽ وڏي ٻولي جا ماڊل

- ۽ ان جا قسم AI پيداوار جو تعارف
- (LLMs) فوري انجنيئرنگ ۽ وڏي ٻولي ماڊلز
- ايل ايل ايم آرڪيٽيڪچر، پري ٽريننگ، ۽ فائن ٽيوننگ
- .ايل ايل ايم صلاحيتن کي ڳولڻ ۽ ماحول کي ترتيب ڏيڻ

## Study Plan:

\*\*ڏينهن 1: اين ايل پي جو تعارف ۽ لفظ سطح جي تجزيي \*\*

- ٻولي، علم ۽ گرامر جي اصليت ۽ تاريخ جو مطالعو ڪريو. (2 ڪلاڪ) ، NLP ٻولي پروسيسنگ ۾: (9:00 AM - 12:00 PM) صبح جو
  - ذريعو "Introduction to Natural Language Processing (NLP)" Stanford University by Coursera
  - وسيلو: "قدرتي ٻوليءَ جي پروسيسنگ لاءِ هڪ نئون نمونو" جان سووا طرفان (ڪتاب)
- اين ايل پي جي مرحلن بابت ڄاڻو، مونجهارو، ۽ انگريزي ۽ هندستاني علائقائي ٻولين ۾ ان جي قسمن. (2 ڪلاڪ) (1:00 PM - 3:00 PM) دوپهر
  - (ڪتاب) Collobert et al. "ذريعو: "قدرتي ٻولي پروسيسنگ (تقريباً) شروع کان
  - ذريعو: "قدرتي ٻولي پروسيسنگ ۾ ابهام" ڊيوڊ يارووسڪي طرفان (تحقيق پيپر)
- ۽ ان جي ايپليڪيشنن جي چٽلينجن جو مطالعو ڪريو. (2 ڪلاڪ) NLP: (3:00 PM - 5:00 PM) شام
  - ذريعو: "قدرتي ٻوليءَ جي پروسيسنگ ۾ چٽلينجز" جورافسڪي ۽ مارٽن پاران (ڪتاب)
  - پاران (ارٽيڪل) IBM "وسيلو: "ايپليڪيشن آف نيچرل لينگويج پروسيسنگ

\*\*ڏينهن 2: لفظ جي سطح جو تجزيو (حصو 1) \*\*

- بنيادي اصطلاحن جو مطالعو ڪريو: ٽوڪنائيزيشن، اسٽيمنگ، لميٽائيزيشن، ۽ سروي آف انگلش مورفولوجي. (3 ڪلاڪ) : (9:00 AM - 12:00 PM) صبح جو

- پاران (ٽيوٽوريل) "Tokenization ۽ Lemmatization" NLTK :وسيلو
  - پاران (ڪتاب) OUP "ذريعو: "انگريزي مورفولوجي
- انفليڪشنل ۽ ڊيريويشنل مورفولوجي بابت ڄاڻو. (2 ڪلاڪ) (1:00 PM - 3:00 PM) منجهند جو
  - وسيلو: لسانيات پاران "انفريڪشنل ۽ ڊيريوشنل مورفولوجي" (آرٽيڪل)
  - ذريعو: "انگلش جو مورفولوجي" ڪيمبرج يونيورسٽي پريس پاران (ڪتاب)
- مطالعو مورفولوجيڪل ماڊل: ڊڪشنري لوڪ اپ ۽ محدود رياستي مورفولوجي. (2 ڪلاڪ) (3:00 PM - 5:00 PM) شام
  - ذريعو: "فائنٽ اسٽيٽ مورفولوجي" اسٽينفورڊ يونيورسٽي طرفان (ٽيوٽوريل)
  - پاران (ٽيوٽوريل) NLTK "وسيلو: "لغت ڏس

### ڏينهن 3: لفظ جي سطح جو تجزيو (حصو 2)

- بابت ڄاڻو. (3 ڪلاڪ) morphological parsing سان FST (Finite State Transducer) (9:00 AM - 12:00 PM): صبح جو
  - سان مورفولوجي پارسنگ" اسٽينفورڊ يونيورسٽي طرفان (ٽيوٽوريل) "FST: ذريعو
  - ذريعو: "فائنٽ اسٽيٽ ٽرانسڊيوسرز" پاران آٽوماٽا (آرٽيڪل)
- مطالعو گرام ۽ ان جي تبديلي: بگگرام، ٽرگرام، ۽ سادو (اڻ سموٽ ٿيل) اين-گرام. (2 ڪلاڪ) (1:00 PM - 3:00 PM) منجهند جو
  - (ٽيوٽوريل) "N-Grams" پاران NLTK: ذريعو
  - وسيلو: وڪيپيڊيا طرفان "بيگرام ۽ ٽرگرام" (آرٽيڪل)
- حساسيت بابت ڄاڻو: کليل بمقابله بند لفظي ڪم. (2 ڪلاڪ) N-gram ٽريننگ ڪورس ۽ نامعلوم لفظن کي: (3:00 PM - 5:00 PM) شام
  - ذريعو: اسٽينفورڊ يونيورسٽي طرفان "اين-گرام حساسيت" (ٽيوٽوريل)
  - ذريعو: "کليل ۽ بند لفظي ڪم" لسانيات طرفان (آرٽيڪل)

### ڏينهن 4: نحو جو تجزيو

- ۽ ٽيگ سيٽ جو مطالعو ڪريو. (3 ڪلاڪ) (POS) لاءِ پارٽ آف اسپيچ ٽيگنگ (Upenn Treebank) انگريزي (9:00 AM - 12:00 PM) صبح جو
  - پاران "پارٽ آف اسپيچ ٽيگنگ" (ٽيوٽوريل) NLTK: وسيلو
  - پاران (آرٽيڪل) UPenn "Penn Treebank": ذريعو
- ٽيگنگ ۾ مشڪلاتن ۽ چئلينجن بابت ڄاڻو. (2 ڪلاڪ) POS (1:00 PM - 3:00 PM) منجهند جو
  - وسيلو: جورفسڪي پاران "پارٽ-آف-اسپيچ ٽيگنگ ۾ چيلنجز" ۽ مارٽن (ڪتاب)
  - ٽيگنگ ۾ مشڪلات" لسانيات طرفان (آرٽيڪل) "POS: ذريعو
- قاعدي تي ٻڌل، اسٽوچسٽڪ، ۽ ٽرانسفارميشن تي ٻڌل ٽيگنگ جو مطالعو ڪريو. (2 ڪلاڪ) (3:00 PM - 5:00 PM) شام
  - ذريعو: اسٽينفورڊ يونيورسٽي طرفان "قاعدي تي ٻڌل ٽيگنگ" (ٽيوٽوريل)
  - پاران "اسٽڪاسٽڪ ۽ ٽرانسفارميشن تي ٻڌل ٽيگنگ" (ٽيوٽوريل) NLTK: وسيلو

### ڏينهن 5: نحوي تجزيا ۽ لفظي تجزيا

- ٽيگنگ لاءِ. (3 ڪلاڪ) POS ۽ ويٽربي (HMM) پيدا ٿيندڙ ماڊل بابت ڄاڻو: پوशيده مارڪوف ماڊل (9:00 AM - 12:00 PM) صبح
  - ذريعو: "پوशيده مارڪوف ماڊل" اسٽينفورڊ يونيورسٽي طرفان (ٽيوٽوريل)
  - وڪيپيڊيا طرفان (آرٽيڪل) "Viterbi Algorithm": ذريعو
- (ڪلاڪ 2). (CRF) مطالعي جو امتيازي ماڊل: وڌ ۾ وڌ اينٽراپي ماڊل ۽ مشروط ريٽم فيلڊ: (1:00 PM - 3:00 PM) دوپهر
  - ذريعو: "وڌ کان وڌ اينٽروپي ماڊل" اسٽينفورڊ يونيورسٽي طرفان (ٽيوٽوريل)
  - پاران (ٽيوٽوريل) NLTK "وسيلو: "شرطي بي ترتيب واري فيلڊ
- مطالعو لفظي تجزيو: تعارف، مطلب نمانندگي، ۽ لغوي معنيٰ. (2 ڪلاڪ) (PM - 5:00 شام 3:00) شام
  - ذريعو "Semantic Analysis" Stanford University (tutorial): ذريعو
  - ذريعو: لسانيات طرفان "ليڪسيڪل سيمينٽڪس" (آرٽيڪل)

### AI ڏينهن 6: معنوي تجزيو ۽ پيدا ڪندڙ

- WordNet. (3 ڪلاڪ) ڪورس جي مطالعي ۽ مختلف ٻولين جي لغتن جي مطالعي بابت ڄاڻو جيئن (9:00 AM - 12:00 PM) صبح جو
  - ذريعو: اسٽينفورڊ يونيورسٽي طرفان "ڪورس مطالعو" (ٽيوٽوريل)
  - پاران پرنسٽن يونيورسٽي (آرٽيڪل) "WordNet": ذريعو
- homonymy, polysemy, synonymy, and hyponymy: ليڪسيمز ۽ انهن جي حواس جي وچ ۾ لاڳاپن جو مطالعو ڪريو (1:00 PM - 3:00 PM) دوپهر
  - وسيلو: لسانيات طرفان "ليڪسيڪل رابطا" (آرٽيڪل)
  - پاران "لفظ احساس تڪرار" (ٽيوٽوريل) NLTK: ذريعو
- ماڊل جا قسم، ۽ فائدا ۽ حدون. (2 ڪلاڪ) AI جو مطالعو ڪريو: تعارف، جنريٽيو AI جنريٽو (3:00 PM - 5:00 PM) شام
  - ذريعو: اسٽينفورڊ يونيورسٽي طرفان "جنريٽو اي آئي" (ٽيوٽوريل)
  - ماڊلز" (آرٽيڪل) AI وسيلو: وڪيپيڊيا پاران "پيداوار

## Key Topics

سروي آف انگلش مورفولوجي، اسٽوچسٽڪ ۽ ٽرانسفارميشن، انفليڪشنل مورفولوجي، مشڪلاتون، سيمينٽڪ ايناليسس، وڏي ٻوليءَ جا ماڊل، Generative AI، Meta Llama2، Google PaLM2 LLM، Langchain, LangChain, Laplace Smoothing, Language, FST, Algorithm, Prompt Engineering, Upenn Treebank, Simple, Lexical Semantics, Generative Adversarial Networks, Module Content, HMM, Introduction Anaphora،، علمWord Sense Disambiguation (WSD) Generative AI، NLP، Synonymy، NLP، جا چئلينج NLP، اصل ۽ تاريخ، مارڪوف ماڊل، اين-گرام حساسيت، نحوي ۽ اصطلاحي WorldNet، Parsers