

```

print("Hello World!")

Hello World!

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

pip install PyPDF2

Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.6/232.6 kB 4.6 MB/s eta 0:00:00
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1

import PyPDF2
import os
import json

def pdf_to_json(pdf_path):
    """Converts the content of a PDF to JSON format."""
    content = {}

    with open(pdf_path, 'rb') as pdf_file:
        reader = PyPDF2.PdfReader(pdf_file)
        for page_num in range(len(reader.pages)):
            page = reader.pages[page_num]
            content[f"page_{page_num + 1}"] = page.extract_text()

    return content

def save_as_json(data, output_path):
    """Saves data to a JSON file."""
    with open(output_path, 'w', encoding='utf-8') as json_file:
        json.dump(data, json_file, ensure_ascii=False, indent=4)

if __name__ == "__main__":
    folder_path = "/content/drive/My Drive/BEProjectColab/Resumes"
    output_folder = "/content/drive/My Drive/BEProjectColab/ResumesJSON"

    os.makedirs(output_folder, exist_ok=True)

    for file in os.listdir(folder_path):
        if file.endswith(".pdf"):
            pdf_path = os.path.join(folder_path, file)
            json_content = pdf_to_json(pdf_path)

            output_filename = os.path.splitext(file)[0] + ".json"
            output_path = os.path.join(output_folder, output_filename)
            save_as_json(json_content, output_path)

    print("Conversion complete!")
    print(json_content)

    Conversion complete!
    {'page_1': 'Sourabh Bajaj Email : mail@website.com \nhttp://www.sourabhajaj.com Mobile : +1-123-456-7890 \nEducation \n•Georgia Insti'

```

---

```

pip install nltk

Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)

import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
True

```

```

import json
import os
import string
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

# Initialize lemmatizer and stopwords
lemmatizer = WordNetLemmatizer()
stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    # Convert to lowercase
    text = text.lower()

    # Tokenize
    tokens = word_tokenize(text)

    # Remove '.', ' '
    text = text.replace('.', '')
    text = text.replace(' ', '')

    # Remove punctuation and stopwords, and lemmatize
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words and word not in string.punctuation]

    # Rejoin words after preprocessing
    return ' '.join(tokens)

def preprocess_json_file(json_file_path, preprocessed_folder_path):
    with open(json_file_path, 'r', encoding='utf-8') as file:
        data = json.load(file)

    # Check if the data is a dictionary or a list and preprocess accordingly
    if isinstance(data, dict):
        for key, value in data.items():
            if isinstance(value, str): # Ensure we're working with strings
                data[key] = preprocess_text(value)
    elif isinstance(data, list):
        data = [preprocess_text(text) for text in data if isinstance(text, str)]
    else:
        # If the data is not a dictionary or list, we won't process it
        return

    # Create filename for the preprocessed data
    base_name = os.path.basename(json_file_path)
    preprocessed_json_path = os.path.join(preprocessed_folder_path, base_name)

    with open(preprocessed_json_path, 'w', encoding='utf-8') as file:
        json.dump(data, file, ensure_ascii=False, indent=4)

if __name__ == "__main__":
    folder_path = "/content/drive/My Drive/BEProjectColab/ResumesJSON"
    preprocessed_folder_path = os.path.join(folder_path, "Preprocessed")

    # Create the Preprocessed folder if it doesn't exist
    os.makedirs(preprocessed_folder_path, exist_ok=True)

    for file in os.listdir(folder_path):
        if file.endswith(".json"):
            json_path = os.path.join(folder_path, file)
            preprocess_json_file(json_path, preprocessed_folder_path)

    print("Preprocessing complete!")

    Preprocessing complete!

pip install scikit-learn nltk

Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.3)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.2.0)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)

pip install spacy

```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.10/dist-packages (3.6.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.9)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy) (8.1.12)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (0.9.0)
Requirement already satisfied: pathy<0.10.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (0.10.3)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (4.66.1)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.23.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.10.11)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy) (67.7.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (23.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.3.0)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2023.7.25)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy) (0.7.9)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy) (0.0.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.10/dist-packages (from typer<0.10.0,>=0.3.0->spacy) (8.1.7)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2->spacy) (2.1.3)
```

```
!python -m spacy download en_core_web_sm
```

```
2023-11-21 03:56:16.468229: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempt
2023-11-21 03:56:16.468298: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempt
2023-11-21 03:56:16.468338: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempt
2023-11-21 03:56:16.477996: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-11-21 03:56:17.741631: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Collecting en-core-web-sm==3.6.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.6.0/en_core_web_sm-3.6.0-py3-none-any.whl
```

```
12.8/12.8 MB 36.1 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.7.0,>=3.6.0 in /usr/local/lib/python3.10/dist-packages (from en-core-web-sm==3.6.0) (3.6.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (3.0.9)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (8.1.12)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (2.0.10)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (0.9.0)
Requirement already satisfied: pathy<0.10.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (0.10.3)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (4.66.1)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (1.23.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (1.10.11)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (3.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (67.7.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (23.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm) (3.3.0)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2023.7.25)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy) (0.7.9)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy) (0.0.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.10/dist-packages (from typer<0.10.0,>=0.3.0->spacy) (8.1.7)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2->spacy) (2.1.3)
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
```

```
import json
import os
import spacy
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
# Load the spaCy model
nlp = spacy.load("en_core_web_sm")
```

```
def load_preprocessed_data(json_file_path):
    """Load and concatenate content from JSON file."""
```

```

with open(json_file_path, 'r', encoding='utf-8') as file:
    data = json.load(file)
if isinstance(data, dict):
    return ' '.join(data.values())
elif isinstance(data, list):
    return ' '.join(data)
else:
    return data

def spacy_tokenizer(document):
    """Tokenize, lemmatize and filter out stopwords using spaCy."""
    tokens = nlp(document)
    return [token.lemma_ for token in tokens if not token.is_stop and not token.is_punct]

def extract_top_keywords_with_spacy(folder_path, n_keywords=10):
    """Extract top-N keywords from all documents in a directory using spaCy and TF-IDF."""
    documents = []

    for file in os.listdir(folder_path):
        if file.endswith(".json"):
            json_path = os.path.join(folder_path, file)
            documents.append(load_preprocessed_data(json_path))

    if not documents:
        return []

    vectorizer = TfidfVectorizer(tokenizer=spacy_tokenizer)
    tfidf_matrix = vectorizer.fit_transform(documents)

    feature_names = vectorizer.get_feature_names_out()
    sorted_indices = tfidf_matrix.sum(axis=0).argsort()[0, -n_keywords:].tolist()[0]
    top_keywords = [feature_names[i] for i in sorted_indices]

    return top_keywords

if __name__ == "__main__":
    folder_path = "/content/drive/My Drive/BEProjectColab/ResumesJSON/Preprocessed"
    keywords = extract_top_keywords_with_spacy(folder_path, n_keywords=20)
    print(keywords)

```

```
[ 'pipeline', 'discovery', 'content', 'app', 'management', 'science', 'model', 'service', 'library', 'software', 'experience', 'system' ]
```

/usr/local/lib/python3.10/dist-packages/sklearn/feature\_extraction/text.py:528: UserWarning: The parameter 'token\_pattern' will not be used as it has no effect since 'tokenizer' is not None.  
warnings.warn(

```
print("That's it for Stage 1")
```

That's it for Stage 1