

⚡️ Installing Packages

```
!pip install pdf2image
!pip install pytesseract
!pip install KeyBERT
!pip install transformers
!pip install torch
!pip install bert-extractive-summarizer
!pip install sentencepiece
!pip install sentence-transformers
!pip install dgl
!pip install pykeen

Collecting pdf2image
  Downloading pdf2image-1.16.3-py3-none-any.whl (11 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from pdf2image) (9
Installing collected packages: pdf2image
Successfully installed pdf2image-1.16.3
Collecting pytesseract
  Downloading pytesseract-0.3.10-py3-none-any.whl (14 kB)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from pte
Requirement already satisfied: Pillow>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from pytes
Installing collected packages: pytesseract
Successfully installed pytesseract-0.3.10
Collecting KeyBERT
  Downloading keybert-0.8.3.tar.gz (29 kB)
  Preparing metadata (setup.py) ... done
Collecting sentence-transformers>=0.3.8 (from KeyBERT)
  Downloading sentence-transformers-2.2.2.tar.gz (85 kB)
  _____ 86.0/86.0 kB 4.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: scikit-learn>=0.22.2 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from KeyBER
Requirement already satisfied: rich>=10.4.0 in /usr/local/lib/python3.10/dist-packages (from KeyBERT
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from
Collecting transformers<5.0.0,>=4.6.0 (from sentence-transformers>=0.3.8->KeyBERT)
  Downloading transformers-4.35.0-py3-none-any.whl (7.9 MB)
  _____ 7.9/7.9 MB 38.8 MB/s eta 0:00:00
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from sentence-transf
Requirement already satisfied: torch>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from sentenc
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (from sentence
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (from sentence-transf
Collecting sentencepiece (from sentence-transformers>=0.3.8->KeyBERT)
  Downloading sentencepiece-0.1.99-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.3 M
  _____ 1.3/1.3 MB 43.4 MB/s eta 0:00:00
Collecting huggingface-hub>=0.4.0 (from sentence-transformers>=0.3.8->KeyBERT)
  Downloading huggingface_hub-0.19.1-py3-none-any.whl (311 kB)
  _____ 311.1/311.1 kB 23.7 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from hug
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggin
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from hugg
Requirement already satisfied: mdurl~0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.6.0->
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.6.
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.6.0-
Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages (from torch>
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from tr
Collecting tokenizers<0.15,>=0.14 (from transformers<5.0.0,>=4.6.0->sentence-transformers>=0.3.8->Ke
  Downloading tokenizers-0.14.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.8 MB)
  _____ 3.8/3.8 MB 88.8 MB/s eta 0:00:00
```

```
Collecting safetensors>=0.3.1 (from transformers<5.0.0,>=4.6.0->sentence-transformers>=0.3.8->KeyBERT)
  Downloading safetensors-0.4.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.3 MB)
    1 3/1 3 MB 66 6 MB/s eta 0:00:00
```

```
!pip install rouge
!pip install textacy
!pip install pdfplumber
!pip install rdflib
!pip install bert-score
!pip install autocorrect
!pip install sentence-transformers
!pip install faiss-cpu --no-cache
```

```
!apt-get install poppler-utils
!sudo apt-get install tesseract-ocr
!pip install pdfplumber
!pip install autocorrect
!pip install rouge
!pip install rdflib
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
poppler-utils is already the newest version (22.02.0-2ubuntu0.2).
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tesseract-ocr is already the newest version (4.1.1-2.1build1).
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
Requirement already satisfied: pdfplumber in /usr/local/lib/python3.10/dist-packages (0.10.3)
Requirement already satisfied: pdfminer.six==20221105 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: Pillow>=9.1 in /usr/local/lib/python3.10/dist-packages (from pdfplumber)
Requirement already satisfied: pypdfium2>=4.18.0 in /usr/local/lib/python3.10/dist-packages (from pdfpl
Requirement already satisfied: charset-normalizer>=2.0.0 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: cryptography>=36.0.0 in /usr/local/lib/python3.10/dist-packages (from pd
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.10/dist-packages (from cryptography)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.12->c
Requirement already satisfied: autocorrect in /usr/local/lib/python3.10/dist-packages (2.6.1)
Requirement already satisfied: rouge in /usr/local/lib/python3.10/dist-packages (1.0.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from rouge) (1.16.0)
Collecting rdflib
  Downloading rdflib-7.0.0-py3-none-any.whl (531 kB)
    531.9/531.9 kB 7.9 MB/s eta 0:00:00
Collecting isodate<0.7.0,>=0.6.0 (from rdflib)
  Downloading isodate-0.6.1-py2.py3-none-any.whl (41 kB)
    41.7/41.7 kB 5.4 MB/s eta 0:00:00
Requirement already satisfied: pyparsing<4,>=2.1.0 in /usr/local/lib/python3.10/dist-packages (from rdf
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from isodate<0.7.0,>=0.6
Installing collected packages: isodate, rdflib
Successfully installed isodate-0.6.1 rdflib-7.0.0
```

▼ Importing Packages

```
import cv2
import os
from pdf2image import convert_from_path
import numpy as np
from google.colab.patches import cv2_imshow
import pytesseract
import numpy as np
import pandas as pd
import re
```

```
from keybert import KeyBERT
import spacy
from transformers import AutoModel, AutoTokenizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
from summarizer import Summarizer, TransformerSummarizer
from sentence_transformers import SentenceTransformer
from transformers import pipeline
import torch
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer
import math
import IPython
import pickle
import pandas as pd
import matplotlib.pyplot as plt
import pdfplumber
import numpy as np
import itertools
import unicodedata
from re import sub
import matplotlib.pyplot as plt
import matplotlib.image as img
import string
from string import punctuation
import time
from pprint import pprint
from sentence_transformers import SentenceTransformer
# import faiss
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
from autocorrect import Speller
spell = Speller()
from sentence_transformers import CrossEncoder
from rouge import Rouge
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet
from nltk.probability import FreqDist
from nltk.parse.corenlp import CoreNLPParser, CoreNLPDependencyParser
from nltk.tree import ParentedTree
from nltk.stem.snowball import SnowballStemmer
# Use English stemmer.
stemmer = SnowballStemmer("english")
from collections import Counter
from tqdm import tqdm
import copy
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import spacy
from spacy.matcher import Matcher
from spacy.matcher import DependencyMatcher
from spacy.tokens import Span
from spacy import displacy
from rdflib.namespace import DC, DCTERMS, DOAP, FOAF, OWL, RDF, RDFS, SKOS, VOID, XMLNS, XSD
from rdflib import URIRef, BNode, Literal, Namespace, Graph
from rdflib.extras import describer
import dgl
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from sklearn.model_selection import train_test_split
import networkx as nx
import matplotlib.pyplot as plt
```

```
from sklearn.metrics.pairwise import cosine_similarity
from pykeen.triples import TriplesFactory
from pykeen.pipeline import pipeline
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import dgl.function as fn
```

▼ Data Extraction

```
def detect_figure_without_text(input_image_path):
    # Read the input image
    image = cv2.imread(input_image_path)

    # Convert the image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian blur to reduce noise and improve edge detection
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)

    # Detect edges using Canny edge detection
    edges = cv2.Canny(blurred, 50, 150)

    # Find contours in the edge-detected image
    contours, _ = cv2.findContours(edges.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Filter and sort contours based on area
    valid_contours = [contour for contour in contours if cv2.contourArea(contour) > 1000]
    sorted_contours = sorted(valid_contours, key=cv2.contourArea, reverse=True)

    # Check if at least one figure is detected
    if len(sorted_contours) <= 2 and len(sorted_contours)>0:
        largest_contour = sorted_contours[0]
        x, y, w, h = cv2.boundingRect(largest_contour)
        if h > 10:
            # Crop the region containing the figure
            figure_only = image[y:y+h, x:x+w]
            output_image_path = os.path.join("figures", os.path.basename(input_image_path))
            # Display the figure without text
            cv2.imshow(figure_only) # Display the image using cv2_imshow
            cv2.imwrite(output_image_path, figure_only)
            print(f"Saved figure to {output_image_path}")
    elif len(sorted_contours) >2:
        largest_contour = sorted_contours[0]
        if cv2.boundingRect(largest_contour)[3]>10:
            canvas = np.zeros_like(gray)
            cv2.drawContours(canvas, sorted_contours[:4], -1, (255, 255, 255), thickness=cv2.FILLED)
            x, y, w, h = cv2.boundingRect(canvas)
            output_image_path = os.path.join("figures", os.path.basename(input_image_path))
            # Display the figure without text
            figure_only = image[y:y+h, x:x+w]
            cv2.imshow(figure_only) # Display the image using cv2_imshow
            cv2.imwrite(output_image_path, figure_only)
            print(f"Saved figure to {output_image_path}")
        else:
            print("No figure detected in the image.")
    else:
        print("No figure detected in the image.")

def returnBBox(input_image_path):
    # Read the input image
    image = cv2.imread(input_image_path)
```

```
# Convert the image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Apply Gaussian blur to reduce noise and improve edge detection
blurred = cv2.GaussianBlur(gray, (5, 5), 0)

# Detect edges using Canny edge detection
edges = cv2.Canny(blurred, 50, 150)

# Find contours in the edge-detected image
contours, _ = cv2.findContours(edges.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Filter and sort contours based on area
valid_contours = [contour for contour in contours if cv2.contourArea(contour) > 1000]
sorted_contours = sorted(valid_contours, key=cv2.contourArea, reverse=True)

# Check if at least one figure is detected
if len(sorted_contours) <= 2 and len(sorted_contours)>0:
    largest_contour = sorted_contours[0]
    return cv2.boundingRect(largest_contour)
elif len(sorted_contours) >2:
    largest_contour = sorted_contours[0]
    if cv2.boundingRect(largest_contour)[3]>10:
        canvas = np.zeros_like(gray)
        cv2.drawContours(canvas, sorted_contours[:4], -1, (255, 255, 255), thickness=cv2.FILLED)
        return cv2.boundingRect(canvas)
    else:
        return 0, 0, 0,0
else:
    return 0, 0 ,0 ,0
```

```
# Specify the path to your PDF file
pdf_file_path = 'PMBOK6-2017.pdf'

# Define the range of pages you want to convert
start_page = 431
end_page = 494
images = convert_from_path(
    pdf_file_path,
    first_page=start_page,
    last_page=end_page,
    poppler_path='/usr/bin' # Path to Poppler in Google Colab
)
# Convert pages from start_page to end_page to images

# Specify the path to the folder where the converted images will be saved
output_folder = 'output_images'

# Create the output folder if it doesn't exist
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

# Iterate through the images and save them
for i, image in enumerate(images):
    # Save each image with a unique name in the output folder
    page_number = start_page + i
    image_path = os.path.join(output_folder, f'page_{i}.jpg')
    image.save(image_path, 'JPEG')

# Detect and display figures in the saved image
detect_figure_without_text(image_path)
```

```
# Iterate through the images in the folder and detect/display figures
for filename in os.listdir(output_folder):
    if filename.endswith('.jpg'):
        image_path = os.path.join(output_folder, filename)
        detect_figure_without_text(image_path)
```

```
def extract_text_without_figure(input_image_path, figure_bbox):
    # Read the input image
    image = cv2.imread(input_image_path)
    # Convert the image to grayscale
```

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Apply Gaussian blur to reduce noise and improve OCR accuracy
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
# Create a mask to hide the region of the figure
mask = np.zeros_like(blurred)
x, y, w, h = figure_bbox
mask[y:y+h, x:x+w] = 255
# Apply the mask to the image
inverted_mask = cv2.bitwise_not(mask)
masked_image = cv2.bitwise_and(blurred, blurred, mask=inverted_mask)
cv2_imshow(masked_image)
# Perform OCR using pytesseract on the masked image
extracted_text = pytesseract.image_to_string(masked_image)

# Print the extracted text
print("Extracted Text (without figure):")
return extracted_text

extracted_texts = []
filenames = os.listdir(output_folder)

# Define a custom sorting key function to extract the number from filenames
def get_number_from_filename(filename):
    return int(filename.split('_')[1].split('.')[0])

# Sort the filenames based on the extracted numbers
sorted_filenames = sorted(filenames, key=get_number_from_filename)

for filename in sorted_filenames:
    if filename.endswith('.jpg'):
        image_path = os.path.join(output_folder, filename)
        figure_bbox = returnBBox(image_path)
        extracted_texts.append(extract_text_without_figure(image_path, figure_bbox))
```

11

PROJECT RISK MANAGEMENT

Project Risk Management includes the processes of conducting risk management planning, identification, analysis, response planning, response implementation, and monitoring risk on a project. The objectives of project risk management are to increase the probability and/or impact of positive risks and to decrease the probability and/or impact of negative risks, in order to optimize the chances of project success.

The Project Risk Management processes are:

11.1 Plan Risk Management—The process of defining how to conduct risk management activities for a project.

11.2 Identify Risks—The process of identifying individual project risks as well as sources of overall project risk, and documenting their characteristics.

11.3 Perform Qualitative Risk Analysis—The process of prioritizing individual project risks for further analysis or action by assessing their probability of occurrence and impact as well as other characteristics.

11.4 Perform Quantitative Risk Analysis—The process of numerically analyzing the combined effect of identified individual project risks and other sources of uncertainty on overall project objectives.

11.5 Plan Risk Responses—The process of developing options, selecting strategies, and agreeing on actions to address overall project risk exposure, as well as to treat individual project risks.

11.6 Implement Risk Responses—The process of implementing agreed-upon risk response plans.

```
chapters = []
subchapters = []
subsubchapters = []
```

```
dataset = {"Chapter": [], "Text": []}
```

```
current_chapter = ""
text_accumulator = ""
```

```
lines = " ".join(extracted_texts)
lines
```

395

```
# Regular expression pattern to match strong formatting
strong_format_pattern = r'<strong>(.*)</strong>'
```

```
# Load spaCy model
nlp = spacy.load("en_core_web_sm") # You can replace "en_core_web_sm" with the appropriate language model
```

```
def preprocess_text(text):
```

```
    # Supprimer les mots 'Gide'
```

```
    text_without_gide = re.sub(r'\b Gide \b', '', text, flags=re.IGNORECASE)
```

```
    # Supprimer les numéros avant 'Gide'
```

```
    text_without_numbers_before_gide = re.sub(r'\b\d+\b\s*Gide\b', ' Gide', text_without_gide, flags=re.IGNORECASE)
```

```
    # Supprimer le texte enclosed dans les balises <strong>
```

```
    text_without_strong_formatting = re.sub(strong_format_pattern, '', text_without_numbers_before_gide)
```

```
    # Remplacer "tt" par "t"
```

```
    text_without_double_t = re.sub(r'tt', 't', text_without_strong_formatting)
```

```
    # Tokenize le texte avec spaCy
```

```
    doc = nlp(text_without_double_t)
```

```
    # Supprimer les espaces supplémentaires, les caractères spéciaux et les stopwords
    cleaned_tokens = []
```

```

for token in doc:
    # Supprimer la ponctuation, les caractères spéciaux et les stopwords
    if not token.is_punct and not token.is_stop:
        # Supprimer les caractères spéciaux
        cleaned_token = re.sub(r'[-_!'"â€¢@#]', '', token.text)

        #if cleaned_token and cleaned_token != 'm': # Vérifier si le jeton n'est pas vide et n'est pas 'm'
        cleaned_tokens.append(token.text)
return ' '.join(cleaned_tokens)

processed_text = preprocess_text(lines)
print(processed_text)

```

PROJECT RISK MANAGEMENT

Project Risk Management includes processes conducting risk management planning identification analysis response planning response implementation monitoring risk project objectives project risk management increase probability and/or impact positive risks decrease probability and/or impact negative risks order optimize chances project success

Project Risk Management processes

11.1 Plan Risk Management process defining conduct risk management activities project

11.2 Identify Risks process identifying individual project risks sources overall project risk documenting characteristics

11.3 Perform Qualitative Risk Analysis process prioritizing individual project risks analysis action assessing probability occurrence impact characteristics

11.4 Perform Quantitative Risk Analysis process numerically analyzing combined effect identified individual project risks sources uncertainty overall project objectives

11.5 Plan Risk Responses process developing options selecting strategies agreeing actions address overall project risk exposure treat individual project risks

11.6 Implement Risk Responses process implementing agreed risk response plans

11.7 Monitor Risks process monitoring implementation agreed risk response plans tracking identified risks identifying analyzing new risks evaluating risk process effectiveness project

Figure 11.1 provides overview Project Risk Management processes Project Management Risk processes presented discrete processes defined interfaces practice overlap interact ways completely detailed PMBOK® Guide

395

Figure 11.1 Project Risk Management Overview

396 1 Gide

KEY CONCEPTS PROJECT RISK MANAGEMENT

projects risky unique undertakings varying degrees complexity aim deliver benefits context constraints assumptions responding stakeholder expectations conflicting changing Organizations choose project risk controlled intentional manner order create value balancing risk reward

Project Risk Management aims identify manage risks addressed project management processes managed risks potential case project deviate plan fail achieve defined project objectives Consequently effectiveness Project Risk Management directly relates project success

Risk exists levels project project contains individual risks affect achievement project objectives important consider riskiness overall project arises combination individual project risks sources uncertainty Project Risk Management processes address levels risk projects defined follows

```

# Define regular expressions to match subchapters and subsubchapters
chapter_pattern = r'11\.\.\d\s.*|\d+\.\d+\.\d\s.*|\d+\.\d+\.\d+\.\d\s.*'
lines = processed_text.splitlines()
num_lines = len(lines)
# subchapter_matches = re.findall(subchapter_pattern, page_text)
# subsubchapter_matches = re.findall(subsubchapter_pattern, page_text)
# subchapter_matches = [match.strip() for match in subchapter_matches]
# subsubchapter_matches = [match.strip() for match in subsubchapter_matches]
line_index = 0
for line_index in range(2,num_lines):
    line = lines[line_index].strip()
    # Check if the line matches a subchapter pattern
    if re.match(chapter_pattern, line):
        if text_accumulator:
            dataset["Chapter"].append(current_chapter)
            dataset["Text"].append(text_accumulator)
            text_accumulator = ""
        current_chapter = line
    elif len(line) >= 25 and not line.startswith('*') and not (line.startswith('Figure ') and len(line) < 80):
        text_accumulator += " " + line
if text_accumulator:
    dataset["Chapter"].append(current_chapter)
    dataset["Text"].append(text_accumulator)

print(dataset["Text"][0])

```

Project Risk Management includes processes conducting risk management planning identification analysis r

```

import pandas as pd
df = pd.DataFrame(dataset)
df = df.applymap(lambda x: x.lstrip())
df

```

	Chapter	Text
0		Project Risk Management includes processes cond...
1	11.2 Identify Risks process identifying indivi...	docmenting characteristics
2	11.3 Perform Qalitative Risk Analysis process ...	action assessing probability occrrence impact ...
3	11.4 Perform Qantitative Risk Analysis process...	indvidual project risks sorces ncertainty over...
4	11.5 Plan Risk Responses process developing op...	address overall project risk exposre treat ind...
...
92	11.7.3.1 WORK PERFORMANCE INFORMATION	Described Section 4.5.1.3 Work performance inf...
93	11.7.3.2 CHANGE REQUESTS	Described Section 4.3.3.4 Monitor Risks proces...
94	11.7.3.3 PROJECT MANAGEMENT PLAN	change project management plan goes

```
df.tail(20)
```

	Chapter	Text
77	11.6.1.2 PROJECT DOCUMENTS	Project documents considered inputs process incl...
78	11.6.1.3 ORGANIZATIONAL PROCESS ASSETS	organizational process assets influence Implementation...
79	11.6.2.1 EXPERT JUDGMENT	Described Section 4.1.2.1 Expertise considered...
80	11.6.2.2 INTERPERSONAL TEAM SKILLS	Interpersonal team skills process include limit...
81	11.6.2.3 PROJECT MANAGEMENT INFORMATION SYSTEM...	Described Section 4.3.2.2 Project management i...
82	11.6.3.1 CHANGE REQUESTS	Described Section 4.3.3.4 Implementation risk ...
83	11.6.3.2 PROJECT DOCUMENTS UPDATES	Project documents updated result carrying process...
84	11.7 MONITOR RISKS	Monitor Risks process monitoring implementation...
85	11.7.1.1 PROJECT MANAGEMENT PLAN	Described Section 4.2.3.1 Project management p...
86	11.7.1.2 PROJECT DOCUMENTS	Project documents considered inputs process incl...
87	11.7.1.3 WORK PERFORMANCE DATA	Described Section 4.3.3.2 Work performance dat...
88	11.7.1.4 WORK PERFORMANCE REPORTS	Described Section 4.5.3.1 Work performance rep...

```
# Enlever les virgules entre les mots dans la colonne 'Concepts'
df['Chapter'] = df['Chapter'].str.replace(r',', '')
```

```
# Enlever les virgules entre les mots dans la colonne 'Description'
df['Text'] = df['Text'].str.replace(r',', '')
```

```
# Remplacer "tt" par "t" dans la colonne 'Concepts'
df['Chapter'] = df['Chapter'].str.replace(r'tt', 't')
```

```
# Remplacer "tt" par "t" dans la colonne 'Description'
df['Text'] = df['Text'].str.replace(r'tt', 't')
```

```
df.to_excel("pmbok.xls")
```

```
import nltk
import pandas as pd
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from nltk.stem import WordNetLemmatizer

# Download the NLTK resources if you haven't already
nltk.download('punkt')
nltk.download('wordnet')

# Instantiate the WordNet lemmatizer
lemmatizer = WordNetLemmatizer()

# Tokenize the text into words and convert them to lowercase
words = [lemmatizer.lemmatize(word.lower()) for word in word_tokenize(processed_text)]

# Calculate the total number of words
total_words = len(words)
```

```
# Calculate the frequency distribution of words
freq_dist = FreqDist(words)

# Calculate word frequencies as percentages
keywords = [(word, freq / total_words * 100) for word, freq in freq_dist.items()]

# Sort keywords by percentage in descending order
keywords = sorted(keywords, key=lambda x: x[1], reverse=True)

# Convert the list of tuples into a DataFrame
data = pd.DataFrame(keywords, columns=['keyword', 'score'])

# Print the top 10 keywords with their percentages
print(data.head(10))
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
   keyword      score
0      risk  9.559924
1    project  5.917573
2   process  1.896018
3  response  1.726375
4 management  1.686458
5  analysis  1.427003
6     plan  1.407045
7   section  1.367129
8 described  1.187506
9 individual  1.107674
```

```
import pandas as pd
import nltk
import spacy
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize

# Download NLTK resources (if not already downloaded)
nltk.download('punkt')
nltk.download('stopwords')

# Load spaCy English language model
nlp = spacy.load('en_core_web_sm')

# Function to extract keywords from Description column and apply lemmatization using spaCy
def extract_keywords(description_text):
    if not isinstance(description_text, str) or not description_text.strip():
        return [] # Return an empty list for non-string or empty input

    print(f"Original Text: {description_text}")

    # Tokenize into sentences and words
    sentences = sent_tokenize(description_text)
    words = word_tokenize(description_text)
    print(f"Tokenized Words: {words}")

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    filtered_words = [word for word in words if word.lower() not in stop_words]
    print(f"Filtered Words: {filtered_words}")

    # Lemmatization using spaCy
    lemmatized_words = []
    doc = nlp(" ".join(filtered_words))
    for token in doc:
        lemmatized_words.append(token.lemma_)
```

```
print(f"Lemmaized Words: {lemmatized_words}")

# Extract nouns (NN, NNS, NNP, NNPS)
keywords = [word for word in lemmatized_words if nltk.pos_tag([word])[0][1] in ['NN', 'NNS', 'NNP', 'NNPS']]
print(f"Extracted Keywords: {keywords}")

return keywords

# Apply the extract_keywords function to the 'Description' column of your DataFrame
df['Keywords'] = df['Text'].apply(extract_keywords)

# Print the updated DataFrame with extracted keywords
df
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Original Text: Project Risk Management includes processes conducting risk management p:
Tokenized Words: ['Project', 'Risk', 'Management', 'includes', 'processes', 'conducting']
Filtered Words: ['Project', 'Risk', 'Management', 'includes', 'processes', 'conducting']
Lemmatized Words: ['project', 'Risk', 'Management', 'include', 'process', 'conducte'],
Extracted Keywords: ['project', 'Risk', 'Management', 'inclde', 'process', 'condcte']
Original Text: documenting characteristics
Tokenized Words: ['documenting', 'characteristics']
Filtered Words: ['documenting', 'characteristics']
Lemmatized Words: ['docmente', 'characteristic']
Extracted Keywords: ['docmente']
Original Text: action assessing probability occurrence impact characteristics
Tokenized Words: ['action', 'assessing', 'probability', 'occurrence', 'impact', 'characte
Filtered Words: ['action', 'assessing', 'probability', 'occurrence', 'impact', 'characte
Lemmatized Words: ['action', 'assess', 'probability', 'occurrence', 'impact', 'characte
Extracted Keywords: ['action', 'assess', 'probability', 'occurrence', 'impact']
Original Text: individual project risks sources uncertainty overall project objectives
Tokenized Words: ['individual', 'project', 'risks', 'sources', 'uncertainty', 'overall'
Filtered Words: ['individual', 'project', 'risks', 'sources', 'uncertainty', 'overall',
Lemmatized Words: ['individual', 'project', 'risk', 'sorce', 'ncertainty', 'overall',
Extracted Keywords: ['individual', 'project', 'risk', 'sorce', 'ncertainty', 'project']
Original Text: address overall project risk exposure treat individual project risks
Tokenized Words: ['address', 'overall', 'project', 'risk', 'exposure', 'treat', 'individu
Filtered Words: ['address', 'overall', 'project', 'risk', 'exposure', 'treat', 'individu
Lemmatized Words: ['address', 'overall', 'project', 'risk', 'exposure', 'treat', 'individu
Extracted Keywords: ['address', 'project', 'risk', 'exposure', 'treat', 'individual']
Original Text: tracking identified risks identifying analyzing new risks evaluating r
Tokenized Words: ['tracking', 'identified', 'risks', 'identifying', 'analyzing', 'new'
Filtered Words: ['tracking', 'identified', 'risks', 'identifying', 'analyzing', 'new'
Lemmatized Words: ['track', 'identify', 'risk', 'identify', 'analyze', 'new', 'risk'
Extracted Keywords: ['track', 'risk', 'analyze', 'risk', 'evaluate', 'risk', 'process']
Original Text: Plan Risk Management process defining conduct risk management activities
Tokenized Words: ['Plan', 'Risk', 'Management', 'process', 'defining', 'conduct', 'ris
Filtered Words: ['Plan', 'Risk', 'Management', 'process', 'defining', 'conduct', 'ris
Lemmatized Words: ['plan', 'risk', 'Management', 'process', 'define', 'conduct', 'ris
Extracted Keywords: ['plan', 'risk', 'Management', 'process', 'define', 'conduct', 'ris
Original Text: Described Section 4.1.3.1 project charter documents high level project
Tokenized Words: ['Described', 'Section', '4.1.3.1', 'project', 'charter', 'documents'
Filtered Words: ['Described', 'Section', '4.1.3.1', 'project', 'charter', 'documents'
Lemmatized Words: ['describe', 'Section', '4.1.3.1', 'project', 'charter', 'document'
Extracted Keywords: ['describe', 'Section', 'project', 'charter', 'document', 'level'
Original Text: Described Section 4.2.3.1 planning Project Risk Management approved st
Tokenized Words: ['Described', 'Section', '4.2.3.1', 'planning', 'Project', 'Risk',
Filtered Words: ['Described', 'Section', '4.2.3.1', 'planning', 'Project', 'Risk', 'I
Lemmatized Words: ['describe', 'Section', '4.2.3.1', 'planning', 'Project', 'Risk',
Extracted Keywords: ['describe', 'Section', 'planning', 'Project', 'Risk', 'Management
Original Text: Project documents considered inputs process include limited stakeholder i
Tokenized Words: ['Project', 'documents', 'considered', 'inputs', 'process', 'includ
Filtered Words: ['Project', 'documents', 'considered', 'inputs', 'process', 'includ
Lemmatized Words: ['project', 'document', 'consider', 'inpt', 'process', 'includ
Extracted Keywords: ['project', 'document', 'inpt', 'process', 'includ', 'stakeholder
Original Text: enterprise environmental factors influence Plan Risk Management proces
Tokenized Words: ['enterprise', 'environmental', 'factors', 'influence', 'Plan', 'Ris
Filtered Words: ['enterprise', 'environmental', 'factors', 'influence', 'Plan', 'Ris
Lemmatized Words: ['enterprise', 'environmental', 'factor', 'influence', 'Plan', 'Ris
Extracted Keywords: ['enterprise', 'factor', 'influence', 'Plan', 'Risk', 'Management
Original Text: organizational process assets influence Plan Risk Management process i
Tokenized Words: ['organizational', 'process', 'assets', 'influence', 'Plan', 'Risk',
Filtered Words: ['organizational', 'process', 'assets', 'influence', 'Plan', 'Risk',
Lemmatized Words: ['organizational', 'process', 'asset', 'influence', 'Plan', 'Risk',
Extracted Keywords: ['process', 'asset', 'influence', 'Plan', 'Risk', 'Management', 'i
Original Text: Described Section 4.1.2.1 Expertise considered individuals groups specia
Tokenized Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'indiv
Filtered Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'indiv
Lemmatized Words: ['describe', 'Section', '4.1.2.1', 'Expertise', 'consider', 'individ
Extracted Keywords: ['describe', 'Section', 'Expertise', 'individual', 'grop', 'knowle
Original Text: Data analysis techniques process includes limited stakeholder analysis
Tokenized Words: ['Data', 'analysis', 'techniques', 'process', 'includes', 'limited',
Filtered Words: ['Data', 'analysis', 'techniques', 'process', 'includes', 'limited', 'l
```

Lemmatized Words: ['datum', 'analysis', 'techniques', 'process', 'includes', 'limited']
Extracted Keywords: ['datum', 'analysis', 'techniques', 'process', 'includes', 'stakeholders']
Original Text: risk management plan developed project kick meeting specific planning
Tokenized Words: ['risk', 'management', 'plan', 'developed', 'project', 'kick', 'meet', 'planning']
Filtered Words: ['risk', 'management', 'plan', 'developed', 'project', 'kick', 'meet', 'planning']
Lemmatized Words: ['risk', 'management', 'plan', 'develop', 'project', 'kick', 'meet', 'planning']
Extracted Keywords: ['risk', 'management', 'plan', 'project', 'kick', 'meet', 'planning']
Original Text: risk management plan component project management plan describes risk
Tokenized Words: ['risk', 'management', 'plan', 'component', 'project', 'management']
Filtered Words: ['risk', 'management', 'plan', 'component', 'project', 'management']
Lemmatized Words: ['risk', 'management', 'plan', 'component', 'project', 'management']
Extracted Keywords: ['risk', 'management', 'plan', 'component', 'project', 'management']
Original Text: Identify Risks process identifying individual project risks sources over time
Tokenized Words: ['Identify', 'Risks', 'process', 'identifying', 'individual', 'project']
Filtered Words: ['Identify', 'Risks', 'process', 'identifying', 'individual', 'project']
Lemmatized Words: ['identify', 'Risks', 'process', 'identify', 'individual', 'project']
Extracted Keywords: ['Risks', 'process', 'individual', 'project', 'risk', 'source', 'plan']
Original Text: Described Section 4.2.3.1 Project management plan components include limited
Tokenized Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan']
Filtered Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan']
Lemmatized Words: ['describe', 'Section', '4.2.3.1', 'Project', 'management', 'plan']
Extracted Keywords: ['describe', 'Section', 'Project', 'management', 'plan', 'components']
Original Text: Project documents considered inputs process include limited Assumption logic
Tokenized Words: ['Project', 'documents', 'considered', 'inputs', 'process', 'include', 'logic']
Filtered Words: ['Project', 'documents', 'considered', 'inputs', 'process', 'include', 'logic']
Lemmatized Words: ['project', 'document', 'consider', 'input', 'process', 'include', 'logic']
Extracted Keywords: ['project', 'document', 'input', 'process', 'include', 'Assumption', 'logic']
Original Text: Described Section 12.2.3.2 project requires external procurement resources
Tokenized Words: ['Described', 'Section', '12.2.3.2', 'project', 'requires', 'external']
Filtered Words: ['Described', 'Section', '12.2.3.2', 'project', 'requires', 'external']
Lemmatized Words: ['Described', 'Section', '12.2.3.2', 'project', 'require', 'external']
Extracted Keywords: ['Section', 'project', 'require', 'procurement', 'resource', 'agreement']
Original Text: Described Section 12.3.1.4 project requires external procurement resources
Tokenized Words: ['Described', 'Section', '12.3.1.4', 'project', 'requires', 'external']
Filtered Words: ['Described', 'Section', '12.3.1.4', 'project', 'requires', 'external']
Lemmatized Words: ['Described', 'Section', '12.3.1.4', 'project', 'require', 'external']
Extracted Keywords: ['Section', 'project', 'require', 'procurement', 'resource', 'procurement']
Original Text: enterprise environmental factors influence Identify Risks process include
Tokenized Words: ['enterprise', 'environmental', 'factors', 'influence', 'Identify', 'Risks']
Filtered Words: ['enterprise', 'environmental', 'factors', 'influence', 'Identify', 'Risks']
Lemmatized Words: ['enterprise', 'environmental', 'factor', 'influence', 'Identify', 'Risks']
Extracted Keywords: ['enterprise', 'factor', 'influence', 'Risks', 'process', 'include']
Original Text: organizational process assets influence Identify Risks process include
Tokenized Words: ['organizational', 'process', 'assets', 'influence', 'Identify', 'Risks']
Filtered Words: ['organizational', 'process', 'assets', 'influence', 'Identify', 'Risks']
Lemmatized Words: ['organizational', 'process', 'asset', 'influence', 'Identify', 'Risks']
Extracted Keywords: ['process', 'asset', 'influence', 'Risks', 'process', 'include']
Original Text: Described Section 4.1.2.1 Expertise considered individuals groups specific
Tokenized Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'individuals']
Filtered Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'individuals']
Lemmatized Words: ['describe', 'Section', '4.1.2.1', 'Expertise', 'consider', 'individuals']
Extracted Keywords: ['describe', 'Section', 'Expertise', 'individual', 'group', 'knowledge']
Original Text: Data gathering techniques process include limited Brainstorming goal brain
Tokenized Words: ['Data', 'gathering', 'techniques', 'process', 'include', 'limited', 'brainstorming']
Filtered Words: ['Data', 'gathering', 'techniques', 'process', 'include', 'limited', 'brainstorming']
Lemmatized Words: ['datum', 'gather', 'techniques', 'process', 'include', 'limited', 'brainstorming']
Extracted Keywords: ['datum', 'gather', 'techniques', 'process', 'include', 'goal', 'brainstorming']
Original Text: Data analysis techniques process include limited Root case analysis Root
Tokenized Words: ['Data', 'analysis', 'techniques', 'process', 'include', 'limited', 'Root']
Filtered Words: ['Data', 'analysis', 'techniques', 'process', 'include', 'limited', 'Root']
Lemmatized Words: ['datum', 'analysis', 'techniques', 'process', 'include', 'limited', 'Root']
Extracted Keywords: ['datum', 'analysis', 'techniques', 'process', 'include', 'Root']
Original Text: Interpersonal team skills process includes limited facilitation Section
Tokenized Words: ['Interpersonal', 'team', 'skills', 'process', 'includes', 'limited']
Filtered Words: ['Interpersonal', 'team', 'skills', 'process', 'includes', 'limited']
Lemmatized Words: ['interpersonal', 'team', 'skill', 'process', 'include', 'limited']
Extracted Keywords: ['team', 'skill', 'process', 'include', 'facilitation', 'Section']
Original Text: sources overall project risk skilled facilitator help participants remain
Tokenized Words: ['sources', 'overall', 'project', 'risk', 'skilled', 'facilitator']
Filtered Words: ['sources', 'overall', 'project', 'risk', 'skilled', 'facilitator']
Lemmatized Words: ['source', 'overall', 'project', 'risk', 'skilled', 'facilitator']
Extracted Keywords: ['source', 'project', 'risk', 'facilitator', 'help', 'participants']
Original Text: prompt list predetermined list risk categories rise individual project

Original Text: prompt list predetermined list risk categories
Tokenized Words: ['prompt', 'list', 'predetermined', 'list', 'risk', 'categories', '']
Filtered Words: ['prompt', 'list', 'predetermined', 'list', 'risk', 'categories', 'r']
Lemmatized Words: ['prompt', 'list', 'predetermine', 'list', 'risk', 'category', 'ri']
Extracted Keywords: ['prompt', 'list', 'predetermine', 'list', 'risk', 'category', '']
Original Text: ndertake risk identification project team condct specialized meeting
Tokenized Words: ['ndertake', 'risk', 'identification', 'project', 'team', 'condct', '']
Filtered Words: ['ndertake', 'risk', 'identification', 'project', 'team', 'condct', '']
Lemmatized Words: ['ndertake', 'risk', 'identification', 'project', 'team', 'condct']
Extracted Keywords: ['ndertake', 'risk', 'identification', 'project', 'team', 'condct']
Original Text: risk register captres details identified individual project risks resl
Tokenized Words: ['risk', 'register', 'captres', 'details', 'identified', 'individual']
Filtered Words: ['risk', 'register', 'captres', 'details', 'identified', 'individual']
Lemmatized Words: ['risk', 'register', 'captre', 'detail', 'identify', 'individual']
Extracted Keywords: ['risk', 'register', 'captre', 'detail', 'individual', 'project']
Original Text: risk report presents information sorces overall project risk smmary i
Tokenized Words: ['risk', 'report', 'presents', 'information', 'sorces', 'overall', '']
Filtered Words: ['risk', 'report', 'presents', 'information', 'sorces', 'overall', '']
Lemmatized Words: ['risk', 'report', 'present', 'information', 'sorce', 'overall', '']
Extracted Keywords: ['risk', 'report', 'present', 'information', 'sorce', 'project']
Original Text: Project documents pdated reslt process inclde limited Assmption log De
Tokenized Words: ['Project', 'documents', 'pdated', 'reslt', 'process', 'inclde', 'li']
Filtered Words: ['Project', 'docsments', 'pdated', 'reslt', 'process', 'inclde', 'lim']
Lemmatized Words: ['project', 'docment', 'pdate', 'reslt', 'process', 'inclde', 'lim']
Extracted Keywords: ['project', 'docment', 'pdate', 'reslt', 'process', 'inclde', 'A']
Original Text: Perform Qalitative Risk Analysis process prioritizing individual projec
Tokenized Words: ['Perform', 'Qalitative', 'Risk', 'Analysis', 'process', 'prioritiz']
Filtered Words: ['Perform', 'Qalitative', 'Risk', 'Analysis', 'process', 'prioritizi']
Lemmatized Words: ['perform', 'Qalitative', 'Risk', 'Analysis', 'process', 'prioriti']
Extracted Keywords: ['perform', 'Risk', 'Analysis', 'process', 'prioritize', 'individ']
Original Text: Described Section 4.2.3.1 Project management plan components inclde r
Tokenized Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan']
Filtered Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan']
Lemmatized Words: ['describe', 'Section', '4.2.3.1', 'Project', 'management', 'plan']
Extracted Keywords: ['describe', 'Section', 'Project', 'management', 'plan', 'componen']
Original Text: Project documents considered inpts process inclde limited Assmption lo
Tokenized Words: ['Project', 'documents', 'considered', 'inpts', 'process', 'inclde']
Filtered Words: ['Project', 'documents', 'considered', 'inpts', 'process', 'inclde']
Lemmatized Words: ['project', 'document', 'consider', 'inpt', 'process', 'inclde', 'li']
Extracted Keywords: ['project', 'document', 'inpt', 'process', 'inclde', 'Assmption']
Original Text: enterprise environmental factors inflence Perform Qalitative Risk Anal
Tokenized Words: ['enterprise', 'environmental', 'factors', 'inflence', 'Perform', 'Q']
Filtered Words: ['enterprise', 'environmental', 'factors', 'inflence', 'Perform', 'Qa']
Lemmatized Words: ['enterprise', 'environmental', 'factor', 'inflence', 'Perform', 'Qa']
Extracted Keywords: ['enterprise', 'factor', 'inflence', 'Risk', 'Analysis', 'inclde']
Original Text: organizational process assets inflence Perform Qalitative Risk Analys
Tokenized Words: ['organizational', 'process', 'assets', 'inflence', 'Perform', 'Qal']
Filtered Words: ['organizational', 'process', 'assets', 'inflence', 'Perform', 'Qali']
Lemmatized Words: ['organizational', 'process', 'asset', 'inflence', 'Perform', 'Qal']
Extracted Keywords: ['process', 'asset', 'inflence', 'Risk', 'Analysis', 'inclde', 'li']
Original Text: Described Section 4.1.2.1 Expertise considered individuals grops specia
Tokenized Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'ind']
Filtered Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'indi']
Lemmatized Words: ['describe', 'Section', '4.1.2.1', 'Expertise', 'consider', 'indiv']
Extracted Keywords: ['describe', 'Section', 'Expertise', 'individual', 'grop', 'knowle']
Original Text: Data gathering techniques process inclde limited interviews Strctred se
Tokenized Words: ['Data', 'gathering', 'techniques', 'process', 'inclde', 'limited']
Filtered Words: ['Data', 'gathering', 'techniques', 'process', 'inclde', 'limited', 'li']
Lemmatized Words: ['datum', 'gather', 'techniques', 'process', 'inclde', 'limited', 'li']
Extracted Keywords: ['datum', 'gather', 'techniques', 'process', 'inclde', 'interview']
Original Text: Data analysis techniques process inclde limited Risk data qality asses
Tokenized Words: ['Data', 'analysis', 'techniques', 'process', 'inclde', 'limited', 'li']
Filtered Words: ['Data', 'analysis', 'techniques', 'process', 'inclde', 'limited', 'Ri']
Lemmatized Words: ['datum', 'analysis', 'techniques', 'process', 'inclde', 'limited', 'Ri']
Extracted Keywords: ['datum', 'analysis', 'techniques', 'process', 'inclde', 'risk', 'li']
Original Text: Interpersonal team skills process inclde limited facilitation Section
Tokenized Words: ['Interpersonal', 'team', 'skills', 'process', 'inclde', 'limited']
Filtered Words: ['Interpersonal', 'team', 'skills', 'process', 'inclde', 'limited']
Lemmatized Words: ['interpersonal', 'team', 'skill', 'process', 'inclde', 'limited']
Extracted Keywords: ['team', 'skill', 'process', 'inclde', 'facilitation', 'Section']
Original Text: help participants remain focsed risk analysis task follow method assoc
Tokenized Words: ['help', 'participants', 'remain', 'focsed', 'risk', 'analysis', 'ta']
Filtered Words: ['help', 'participants', 'remain', 'focsed', 'risk', 'analysis', 'ta']

Lemmatized Words: ['help', 'participant', 'remain', 'focse', 'risk', 'analysis', 'ta:
Extracted Keywords: ['help', 'participant', 'remain', 'focse', 'risk', 'analysis', 't:
Original Text: Risks project categorized sorces risk e.g. risk breakdown strctre RBS
Tokenized Words: ['Risks', 'project', 'categorized', 'sorces', 'risk', 'e.g', '.', 'r:
Filtered Words: ['Risks', 'project', 'categorized', 'sorces', 'risk', 'e.g', '.', 'r:
Lemmatized Words: ['risk', 'project', 'categorize', 'sorce', 'risk', 'e.g', '.', 'ri:
Extracted Keywords: ['risk', 'project', 'categorize', 'sorce', 'risk', 'e.g', 'risk'
Original Text: Data representation techniques process inclde limited Probability impac
Tokenized Words: ['Data', 'representation', 'techniques', 'process', 'incld', 'limite
Filtered Words: ['Data', 'representation', 'techniques', 'process', 'incld', 'limite
Lemmatized Words: ['datum', 'representation', 'techniques', 'process', 'incld', 'limi
Extracted Keywords: ['datum', 'representation', 'techniques', 'process', 'incld', 'Pi
Original Text: ndertake qalitative risk analysis project team condct specialized mee
Tokenized Words: ['ndertake', 'qalitative', 'risk', 'analysis', 'project', 'team', 'co
Filtered Words: ['ndertake', 'qalitative', 'risk', 'analysis', 'project', 'team', 'co
Lemmatized Words: ['ndertake', 'qalitative', 'risk', 'analysis', 'project', 'team',
Extracted Keywords: ['ndertake', 'qalitative', 'risk', 'analysis', 'project', 'team'
Original Text: Project documents pdated reslt carrying process inclde limited Assmption
Tokenized Words: ['Project', 'documents', 'pdated', 'reslt', 'carrying', 'process', 'in
Filtered Words: ['Project', 'documents', 'pdated', 'reslt', 'carrying', 'process', 'in
Lemmatized Words: ['project', 'document', 'pdate', 'reslt', 'carry', 'process', 'incl
Extracted Keywords: ['project', 'document', 'pdate', 'reslt', 'carry', 'process', 'incl
Original Text: Perform Qantitative Risk Analysis process nmerically analyzing combin
Tokenized Words: ['Perform', 'Qantitative', 'Risk', 'Analysis', 'process', 'nmericall
Filtered Words: ['Perform', 'Qantitative', 'Risk', 'Analysis', 'process', 'nmericall
Lemmatized Words: ['perform', 'qantitative', 'Risk', 'Analysis', 'process', 'nmerica
Extracted Keywords: ['perform', 'qantitative', 'Risk', 'Analysis', 'process', 'analy
Original Text: Described Section 4.2.3.1 Project management plan components inclde l
Tokenized Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan'
Filtered Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan',
Lemmatized Words: ['describe', 'Section', '4.2.3.1', 'Project', 'management', 'plan'
Extracted Keywords: ['describe', 'Section', 'Project', 'management', 'plan', 'componen
Original Text: Project documents considered inpts process inclde limited Assumption lo
Tokenized Words: ['Project', 'documents', 'considered', 'inpts', 'process', 'inclde',
Filtered Words: ['Project', 'documents', 'considered', 'inpts', 'process', 'inclde',
Lemmatized Words: ['project', 'document', 'consider', 'inpt', 'process', 'inclde', 'l
Extracted Keywords: ['project', 'document', 'inpt', 'process', 'inclde', 'Assumption',
Original Text: enterprise environmental factors inflence Perform Qantitative Risk An
Tokenized Words: ['enterprise', 'environmental', 'factors', 'inflence', 'Perform', 'Q
Filtered Words: ['enterprise', 'environmental', 'factors', 'inflence', 'Perform', 'Q
Lemmatized Words: ['enterprise', 'environmental', 'factor', 'influence', 'Perform', 'Q
Extracted Keywords: ['enterprise', 'factor', 'influence', 'Risk', 'Analysis', 'proces
Original Text: organizational process assets inflence Perform Qantitative Risk Analy
Tokenized Words: ['organizational', 'process', 'assets', 'inflence', 'Perform', 'Qan
Filtered Words: ['organizational', 'process', 'assets', 'inflence', 'Perform', 'Qan
Lemmatized Words: ['organizational', 'process', 'asset', 'inflence', 'Perform', 'Qan
Extracted Keywords: ['process', 'asset', 'inflence', 'Risk', 'Analysis', 'process',
Original Text: Described Section 4.1.2.1 Expertise considered individuals grops specia
Tokenized Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'ind
Filtered Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'indi
Lemmatized Words: ['describe', 'Section', '4.1.2.1', 'Expertise', 'consider', 'indi
Extracted Keywords: ['describe', 'Section', 'Expertise', 'individual', 'grop', 'knowle
Original Text: Interviews Section 5.2.2.2 generate inpts qantitative risk analysis di
Tokenized Words: ['Interviews', 'Section', '5.2.2.2', 'generate', 'inpts', 'qantitat:
Filtered Words: ['Interviews', 'Section', '5.2.2.2', 'generate', 'inpts', 'qantitat:
Lemmatized Words: ['Interviews', 'section', '5.2.2.2', 'generate', 'inpt', 'qantitat:
Extracted Keywords: ['Interviews', 'section', 'generate', 'inpt', 'qantitative', 'ri:
Original Text: Interpersonal team skills process inclde limited facilitation Section
Tokenized Words: ['Interpersonal', 'team', 'skills', 'process', 'inclde', 'limited',
Filtered Words: ['Interpersonal', 'team', 'skills', 'process', 'inclde', 'limited',
Lemmatized Words: ['interpersonal', 'team', 'skill', 'process', 'inclde', 'limited',
Extracted Keywords: ['team', 'skill', 'process', 'inclde', 'facilitation', 'Section'
Original Text: Quantitative risk analysis reqires inpts qantitative risk analysis mode
Tokenized Words: ['Quantitative', 'risk', 'analysis', 'reqires', 'inpts', 'qantitative
Filtered Words: ['Quantitative', 'risk', 'analysis', 'reqires', 'inpts', 'qantitative
Lemmatized Words: ['quantitative', 'risk', 'analysis', 'require', 'inpt', 'qantitative
Extracted Keywords: ['quantitative', 'risk', 'analysis', 'require', 'inpt', 'qantitative
Original Text: Data analysis techniques process inclde limited Simlation Qantitative
Tokenized Words: ['Data', 'analysis', 'techniques', 'process', 'inclde', 'limited', '
Filtered Words: ['Data', 'analysis', 'techniques', 'process', 'inclde', 'limited', 'S
Lemmatized Words: ['datum', 'analysis', 'techniques', 'process', 'inclde', 'limited',
Extracted Keywords: ['datum', 'analysis', 'techniques', 'process', 'inclde', 'Simlati
Original Text: Project documents considered otots process inclde limited risk report

Tokenized Words: ['Project', 'documents', 'considered', 'otpts', 'process', 'inclde',
Filtered Words: ['Project', 'documents', 'considered', 'otpts', 'process', 'inclde',
Lemmatized Words: ['project', 'document', 'consider', 'otpt', 'process', 'inclde', 'l:
Extracted Keywords: ['project', 'document', 'otpt', 'process', 'inclde', 'risk', 'repo
Original Text: Plan Risk Responses process developing options selecting strategies a
Tokenized Words: ['Plan', 'Risk', 'Responses', 'process', 'developing', 'options', '
Filtered Words: ['Plan', 'Risk', 'Responses', 'process', 'developing', 'options', 'se
Lemmatized Words: ['plan', 'risk', 'Responses', 'process', 'develop', 'option', 'selc
Extracted Keywords: ['plan', 'risk', 'Responses', 'process', 'option', 'select', 'sti
Original Text: Described Section 4.2.3.1 Project management plan components inclde l:
Tokenized Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan'
Filtered Words: ['Described', 'Section', '4.2.3.1', 'Project', 'management', 'plan',
Lemmatized Words: ['describe', 'Section', '4.2.3.1', 'Project', 'management', 'plan'
Extracted Keywords: ['describe', 'Section', 'Project', 'management', 'plan', 'compone
Original Text: Project documents considered inpts process inclde limited Lessons learn
Tokenized Words: ['Project', 'documents', 'considered', 'inpts', 'process', 'inclde',
Filtered Words: ['Project', 'documents', 'considered', 'inpt', 'process', 'inclde', 'l:
Lemmatized Words: ['project', 'document', 'consider', 'inpt', 'process', 'inclde', 'l:
Extracted Keywords: ['project', 'docment', 'inpt', 'process', 'inclde', 'Lessons', 'l:
Original Text: enterprise environmental factors inflence Plan Risk Responses process
Tokenized Words: ['enterprise', 'environmental', 'factors', 'influence', 'Plan', 'Ris
Filtered Words: ['enterprise', 'environmental', 'factors', 'influence', 'Plan', 'Ris
Lemmatized Words: ['enterprise', 'environmental', 'factor', 'influence', 'Plan', 'Ris
Extracted Keywords: ['enterprise', 'factor', 'influence', 'Plan', 'Risk', 'Responses'
Original Text: organizational process assets inflence Plan Risk Responses process inc
Tokenized Words: ['organizational', 'process', 'assets', 'influence', 'Plan', 'Risk',
Filtered Words: ['organizational', 'process', 'assets', 'influence', 'Plan', 'Risk',
Lemmatized Words: ['organizational', 'process', 'asset', 'influence', 'Plan', 'Risk',
Extracted Keywords: ['process', 'asset', 'influence', 'Plan', 'Risk', 'Responses', 'pi
Original Text: Described Section 4.1.2.1 Expertise considered individuals grops specia
Tokenized Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'ind:
Filtered Words: ['Described', 'Section', '4.1.2.1', 'Expertise', 'considered', 'indi:
Lemmatized Words: ['describe', 'Section', '4.1.2.1', 'Expertise', 'consider', 'indi:
Extracted Keywords: ['describe', 'Section', 'Expertise', 'individual', 'grop', 'knowle
Original Text: Data gathering techniques process inclde limited interviews Section 5.
Tokenized Words: ['Data', 'gathering', 'techniques', 'process', 'inclde', 'limited',
Filtered Words: ['Data', 'gathering', 'techniques', 'process', 'inclde', 'limited', 'l:
Lemmatized Words: ['datum', 'gather', 'techniques', 'process', 'inclde', 'limited', 'l:
Extracted Keywords: ['datum', 'gather', 'techniques', 'process', 'inclde', 'intervie
Original Text: Interpersonal team skills process incldes limited facilitation Section
Tokenized Words: ['Interpersonal', 'team', 'skills', 'process', 'incldes', 'limited'
Filtered Words: ['Interpersonal', 'team', 'skills', 'process', 'incldes', 'limited',
Lemmatized Words: ['interpersonal', 'team', 'skill', 'process', 'incldes', 'limited',
Extracted Keywords: ['team', 'skill', 'process', 'incldes', 'facilitation', 'Section'
Original Text: alternative strategies considered dealing threats follows Escalate Esc
Tokenized Words: ['alternative', 'strategies', 'considered', 'dealing', 'threats', 'f:
Filtered Words: ['alternative', 'strategies', 'considered', 'dealing', 'threats', 'fo:
Lemmatized Words: ['alternative', 'strategy', 'consider', 'deal', 'threat', 'follow'
Extracted Keywords: ['alternative', 'strategy', 'deal', 'threat', 'Escalate', 'Escala
Original Text: alternative strategies considered dealing opportnities follows Escalat
Tokenized Words: ['alternative', 'strategies', 'considered', 'dealing', 'opportnities'
Filtered Words: ['alternative', 'strategies', 'considered', 'dealing', 'opportnities',
Lemmatized Words: ['alternative', 'strategy', 'consider', 'deal', 'opportnity', 'follo
Extracted Keywords: ['alternative', 'strategy', 'deal', 'opportnity', 'Escalate', 'e:
Original Text: responses designed se certain events occr risks appropriate project te
Tokenized Words: ['responses', 'designed', 'se', 'certain', 'events', 'occr', 'risks'
Filtered Words: ['responses', 'designed', 'se', 'certain', 'events', 'occr', 'risks'
Lemmatized Words: ['response', 'design', 'se', 'certain', 'event', 'occr', 'risk', 'i:
Extracted Keywords: ['response', 'design', 'se', 'event', 'occr', 'risk', 'appropriat
Original Text: Risk responses planned implemented individual project risks address ove
Tokenized Words: ['Risk', 'responses', 'planned', 'implemented', 'individual', 'projec
Filtered Words: ['Risk', 'responses', 'planned', 'implemented', 'individual', 'projec
Lemmatized Words: ['risk', 'response', 'plan', 'implement', 'individual', 'project',
Extracted Keywords: ['risk', 'response', 'plan', 'implement', 'individual', 'project'
Original Text: nmber alternative risk response strategies considered Data analysis te
Tokenized Words: ['nmber', 'alternative', 'risk', 'response', 'strategies', 'consid
Filtered Words: ['nmber', 'alternative', 'risk', 'response', 'strategies', 'consid
Lemmatized Words: ['nmber', 'alternative', 'risk', 'response', 'strategy', 'consid
Extracted Keywords: ['nmber', 'alternative', 'risk', 'response', 'strategy', 'Data',
Original Text: Decision making techniques select risk response strategy inclde limited
Tokenized Words: ['Decision', 'making', 'techniques', 'select', 'risk', 'response', 's:
Filtered Words: ['Decision', 'making', 'techniques', 'select', 'risk', 'response', 's:


```

Tokenized Words: ['Project', 'documents', 'considered', 'inpts', 'process', 'inclde',
Filtered Words: ['Project', 'documents', 'considered', 'inpts', 'process', 'inclde'],
Lemmatized Words: ['project', 'document', 'consider', 'inpt', 'process', 'inclde', '1:
Extracted Keywords: ['project', 'docment', 'inpt', 'process', 'inclde', 'Isse', 'log
Original Text: Described Section 4.3.3.2 Work performance data contains data project
Tokenized Words: ['Described', 'Section', '4.3.3.2', 'Work', 'performance', 'data',
Filtered Words: ['Described', 'Section', '4.3.3.2', 'Work', 'performance', 'data', '4
Lemmatized Words: ['describe', 'Section', '4.3.3.2', 'work', 'performance', 'datum',
Extracted Keywords: ['describe', 'Section', 'work', 'performance', 'datum', 'contain
Original Text: Described Section 4 5.3.1 Work performance reports provide information
Tokenized Words: ['Described', 'Section', '4', '5.3.1', 'Work', 'performance', 'repo
Filtered Words: ['Described', 'Section', '4', '5.3.1', 'Work', 'performance', 'repo
Lemmatized Words: ['describe', 'Section', '4', '5.3.1', 'work', 'performance', 'repo
Extracted Keywords: ['describe', 'Section', 'work', 'performance', 'report', 'provide
Original Text: Data analysis techniques process inclde limited Technical performance
Tokenized Words: ['Data', 'analysis', 'techniques', 'process', 'inclde', 'limited', 'T
Filtered Words: ['Data', 'analysis', 'techniques', 'process', 'inclde', 'limited', 'Te
Lemmatized Words: ['datum', 'analysis', 'techniques', 'process', 'inclde', 'limited',
Extracted Keywords: ['datum', 'analysis', 'techniques', 'process', 'inclde', 'performa
Original Text: Described Section 8.2.2.5 Risk adits type adit consider effectiveness
Tokenized Words: ['Described', 'Section', '8.2.2.5', 'Risk', 'adits', 'type', 'adit'
Filtered Words: ['Described', 'Section', '8.2.2.5', 'Risk', 'adits', 'type', 'adit',
Lemmatized Words: ['Described', 'Section', '8.2.2.5', 'risk', 'adit', 'type', 'adit'
Extracted Keywords: ['Section', 'risk', 'adit', 'type', 'adit', 'effectiveness', 'ris
Original Text: Meetings process inclde limited risk reviews Risk reviews schedld req
Tokenized Words: ['Meetings', 'process', 'inclde', 'limited', 'risk', 'reviews', 'Ris
Filtered Words: ['Meetings', 'process', 'inclde', 'limited', 'risk', 'reviews', 'Ris
Lemmatized Words: ['meeting', 'process', 'inclde', 'limited', 'risk', 'review', 'ris
Extracted Keywords: ['meeting', 'process', 'inclde', 'risk', 'review', 'risk', 'revie
Original Text: Described Section 4.5.1.3 Work performance information incldes informa
Tokenized Words: ['Described', 'Section', '4.5.1.3', 'Work', 'performance', 'informa
Filtered Words: ['Described', 'Section', '4.5.1.3', 'Work', 'performance', 'informa
Lemmatized Words: ['Described', 'Section', '4.5.1.3', 'work', 'performance', 'informa

```

```

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

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]     date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True

# Fonction pour obtenir des synonymes pour un mot
def get_synonyms(word):
    synonyms = []
    for syn in wordnet.synsets(word):
        for lemma in syn.lemmas():
            synonyms.append(lemma.name())
    return list(set(synonyms)) # Using set to remove duplicates and then converting back to l

# Function to find synonyms for words in 'Keywords' column
def find_keywords_synonyms(keywords):
    synonyms_list = []
    for word in keywords:
        synonyms = get_synonyms(word)
        synonyms_list.append(synonyms)
    return synonyms_list

```

```
# Appliquez la fonction à chaque élément de la colonne "description" pour créer la nouvelle colonne "synonyme"
df['Synonyme'] = df['Keywords'].apply(find_keywords_synonyms)

# Affichez la DataFrame mise à jour.
df
```

	Chapter	Text	Figures	Keywords	Synonyme
0	Project Risk Management includes processes cond...	None	[project, Risk, Management, include, process, c...]	[[task, externalize, throw, undertaking, pictu...]	
1	11.2 Identify Risks process identifying indivi...	documenting characteristics	None	[docmente]	[[[]]]
2	11.3 Perform Qualitative Risk Analysis process ...	action assessing probability occrrence impact ...	None	[action, assess, probability, occrrence, impact]	[[action_mechanism, legal_action, activity, fu...]
3	11.4 Perform Quantitative Risk Analysis process...	indivial project risks sorces ncertainty over...	None	[individual, project, risk, sorce, ncertainty, ...]	[[[], [task, externalize, throw, undertaking, p...]
4	11.5 Plan Risk Responses process developing op...	address overall project risk exposre treat ind...	None	[address, project, risk, exposre, treat, indivi...]	[[savoir-faire, speech, come_up_to, handle, pl...]
...
92	11.7.3.1 WORK PERFORMANCE INFORMATION	Described Section 4.5.1.3 Work performance inf...	None	[Section, work, performance, information, incl...]	[[section, segment, subdivision, part, plane_s...]
93	11.7.3.2 CHANGE REQUESTS	Described Section 4.3.3.4 Monitor Risks proces...	None	[describe, Section, Monitor, Risks, process, r...]	[[describe, draw, delineate, report, name, lin...]
		change project		[change, . . .]	

```
import pandas as pd
from nltk.corpus import wordnet
```

```
# Fonction pour obtenir des synonymes pour un mot, limités à 7 synonymes
def get_synonyms(word):
    synonyms = []
    for syn in wordnet.synsets(word):
        for lemma in syn.lemmas():
            if len(synonyms) >= 7:
                break
            synonym = lemma.name()
            if synonym != word:
                synonyms.append(synonym)
    return synonyms
```

```
# Fonction pour trouver jusqu'à 7 synonymes pour chaque mot dans la liste de mots-clés
def find_keywords_synonyms(keywords):
    synonyms_list = []
    for word in keywords:
        synonyms = get_synonyms(word)
```

```

        synonyms_list.append(synonyms)
    return synonyms_list

# Supposons que vous avez déjà un DataFrame df avec une colonne "Keywords".
# Appliquez la fonction à chaque élément de la colonne "Keywords" pour créer la nouvelle colonne "Synonyme".
df['Synonyme_Limit'] = df['Keywords'].apply(find_keywords_synonyms)

# Affichez la DataFrame mise à jour.
df

```

	Chapter	Text	Figures	Keywords	Synonyme	Synonyme_Limit
0		Project Risk Management includes processes cond...	None	[project, Risk, Management, include, process, c...]	[[task, externalize, throw, undertaking, pictu...]	[[undertake, labor, ...]
1	11.2 Identify Risks process identifying individ...	documenting characteristics	None	[documente]	[]	[]
2	11.3 Perform Qualitative Risk Analysis process ...	action assessing probability occurrence impact ...	None	[action, assess, probability, occurrence, impact]	[[action_mechanism, legal_action, activity, milita...]	[[activity, especially, fu...]
3	11.4 Perform Quantitative Risk Analysis process...	individual project risks sources uncertainty over...	None	[individual, project, risk, source, uncertainty, ...]	[], [task, externalize, throw, undertaking, pictu...]	[], [understand, project, ...]
4	11.5 Plan Risk Responses process developing options	address overall project risk exposure treat individ...	None	[address, project, risk, exposure, treat, individ...	[[savoir-faire, speech, come_up_to, handle, pl...]	[[compute, reference, ...]
...
92	11.7.3.1 WORK PERFORMANCE INFORMATION	Described Section 4.5.1.3 Work performance info...	None	[Section, work, performance, information, incl...]	[[section, segment, subdivision, part, plane_s...]	subdivision, section
93	11.7.3.2 CHANGE REQUESTS	Described Section 4.3.3.4 Monitor Risks process...	None	[describe, Section, Monitor, Risks, process, r...]	[[describe, draw, delineate, report, name, lin...]	[[description, report, tracking, ...]
94	11.7.3.3 PROJECT MANAGEMENT PLAN UPDATES	change project management plan goes organization...	None	[change, project, management, plan, organization...]	[[switch, exchange, alter, variety, shift, change...]	[modification, ...]

```

import pandas as pd
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

```

```

# Fonction pour extraire les verbes d'une description
def extract_verbs(text):
    # Tokenisation des mots

```

```
words = nltk.word_tokenize(text)

# Analyse grammaticale pour obtenir les POS (Part-of-Speech) tags
tags = nltk.pos_tag(words)

# Extraction des verbes (VB, VBD, VBG, VBN, VBP, VBZ)
verbs = [word for word, pos in tags if pos in ['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ']]

return verbs

# Appliquer la fonction à la colonne "description" pour créer la nouvelle colonne "auxiliaires"
df['Auxiliaires'] = df['Text'].apply(extract_verbs)

# Afficher la DataFrame mise à jour
df
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]   date!
```

	Chapter	Text	Figures	Keywords	Synonyme	Synon...
0	Project Risk Management includes processes	None	[project, Risk, Management, include, process, ...]	[[task, externalize, throw, undertaking, pictu...]	[[undertake, labor, ...]	

```
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
```

```
# Remplacez 'facebook/bart-large-cnn' par le modèle que vous souhaitez utiliser
model_name = "facebook/bart-large-cnn"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
```

Downloading	1.58k/1.58k [00:00<00:00,
(...)lve/main/config.json: 100%	65.5kB/s]
Downloading	899k/899k [00:00<00:00,
(...)olve/main/vocab.json: 100%	8.87MB/s]
Downloading	456k/456k [00:00<00:00,
(...)olve/main/merges.txt: 100%	5.79MB/s]
Downloading	1.36M/1.36M [00:00<00:00,

```
def generate_better_description(description):
    if not pd.isnull(description) and description != "":
        input_text = description
        input_ids = tokenizer.encode(input_text, return_tensors="pt", max_length=1024, truncation=True, padding=True)
        output = model.generate(input_ids, max_length=100, num_return_sequences=1, no_repeat_ngram_size=2, temperature=0.7)
        # Extrait la description améliorée du résultat de la génération
        description_amelioree = tokenizer.decode(output[0], skip_special_tokens=True)
        return description_amelioree
    else:
        # Gérez le cas où la description est manquante
        return description
```

```
df['DescriptionAmelioree'] = df['Text'].apply(generate_better_description)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/generation/configuration_utils.py:367: UserWarning
  warnings.warn(
```

```
df
95      DOCUMENTS      updated reslt      None      update. reslt.      throw. undertakina.      labor.
```

```
from transformers import BertTokenizer, BertLMHeadModel
```

```
# Load the pre-trained model and tokenizer
model_name = "bert-base-uncased"
model = BertLMHeadModel.from_pretrained(model_name)
tokenizer = BertTokenizer.from_pretrained(model_name)

# Define a function to get the best synonyms for a word using the model
def get_best_synonyms(word):
    input_text = f"Replace the word '{word}' with:"
    input_ids = tokenizer.encode(input_text, return_tensors="pt")
    outputs = model.generate(input_ids, max_length=20, num_return_sequences=1)
    decoded_output = tokenizer.decode(outputs[0], skip_special_tokens=True)
```

```

return decoded_output

# Apply the function to each element in the "Keywords" column to create the new column "Best_Synonyms"
df['Best_Synonyms'] = df['Keywords'].apply(lambda x: [get_best_synonyms(word) for word in x])

# Display the updated DataFrame
print(df)

If you want to use `BertLMHeadModel` as a standalone, add `is_decoder=True`.

df["Figures"] = None

!pip install xlwt

Collecting xlwt
  Downloading xlwt-1.3.0-py2.py3-none-any.whl (99 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100.0/100.0 kB 1.8 MB/s eta 0:00:00
Installing collected packages: xlwt
Successfully installed xlwt-1.3.0

df.to_excel("pmbok_without_fig.xls")

<ipython-input-44-659d04d61180>:1: FutureWarning: As the xlwt package is no longer maintained, the xlwt
df.to_excel("pmbok_without_fig.xls")

```

```
!zip -r figures.zip figures
```

>Dataframe

```

df = pd.read_excel('PMI.xls')
df.drop(columns=['Unnamed: 0'], inplace=True,)

pattern = r'([a-zA-Z\d]+(\.[a-zA-Z\d]+)*)\s(.+)'
indices = []
sections = []

for row in df['Section']:
    match = re.match(pattern, row)
    if match:
        index, _, section = match.groups()
        indices.append(index)
        sections.append(section)
    else:
        indices.append(None)
        sections.append(None)

df['Index'] = indices
df['Section'] = sections
df = df.fillna('')
df

```

	Section	Text	Page	Figure	figure_text	Index
0	purpose of the practice standard for project r...	the purpose of the practice standard for p...	14	images/image14_0.png	Purpose Practice Principles of the Specializat...	1.1
1	project risk management defi nition	the definition of project risk management , ...	16			1.2
2	role of project risk management in project man...	project risk management is not an optional a...	16			1.3
3	good risk management practice	project risk management is a valuable compon...	17			1.4
4	critical success factors for project risk mana...	figure 1 - 2 . critical success factors for ...	18	images/image18_0.png	Recognize the Value of Risk Management Integra...	1.5
...

```
df["Fusion"] = df["Section"].astype(str) + " " + df["Text"].astype(str)
df
```

Section	Text	Page	Figure	figure_text	Index
0 purpose of the practice standard for project r...	the purpose of the practice standard for p...	14	images/image14_0.png	Purpose Practice Principles of the Specializat...	1.1 pu the star pi

■ *Manageability.* The ease with which the risk owner (or owning organization) can manage the occurrence or impact of the risk.

▼ Keywords

```
def extract_keywords(text):
    # Initialize the KeyBERT model
    keybert_model = KeyBERT()

    # Extract keywords from the input text
    keywords = keybert_model.extract_keywords(
        text, keyphrase_ngram_range=(1, 4), stop_words="english"
    )

    # Join the extracted keywords into a single string, separated by commas
    return [keyword[0] for keyword in keywords]

df["Fusion"] = df.Text.apply(extract_keywords)
df = df.rename(columns={"Fusion": "Keywords"})
df.head()
```

Section	Text	Page	Figure	figure_text	Index	Keywords
0 purpose of the practice standard for project r...	the purpose of the practice standard for p...	14	images/image14_0.png	Purpose Practice Principles of the Specializat...	1.1	[project ri manageme practi standard pr]
1 project risk management defini tion	the definition of project risk management , ...	16			1.2	[project ri manageme define project risk]
2 role of project risk management in project man...	project risk management is not an optional a...	16			1.3	[projec project ri manageme project risk]
3 good risk management practice	project risk management is a valuable	17			1.4	[project ri manageme proces]

effort on the areas of highest risk exposure, or by developing generic risk responses to address groups of related risks.

▼ Definition

▼ *Probability and Impact Matrix.* A probability and impact matrix is a grid for mapping the probability of each risk occurring against its potential impact.

```
model = TransformerSummarizer(transformer_type="XLNet",transformer_model_key="xlnet-base-cased")

generated_summaries = []

# Iterate through the DataFrame and generate summaries for each row
for text in df['Text']:
    summary = ''.join(model(text, min_length=250))
    generated_summaries.append(summary)
```

```
# Add the list of generated summaries as a new column 'Definition' in the DataFrame
df['Definition'] = generated_summaries
```

```
df["Text"][3]
```

' project risk management is a valuable component of project management and it enhances the value of the other project management processes . as with all of these processes , project risk management should be conducted in a manner consistent with existing organizational practices and policies . in addition , like the other processes involved in project management , project risk management should be conducted in a way that is appropriate to the project . project risk management should recognize the business challenges as well as the multi - cultural environment associated with an increasingly global environment including many joint venture projects and customers , suppliers , and workforces spread around the globe . changes in the project management plan that result from the project risk management process may require decisions at the appropriate level of management to reassess personnel , establish or modify budgets , make commitments to others outside the project , interact with regulators , and comply with the rules of accountin

```
df["Definition"][3]
```

'project risk management should recognize the business challenges as well as the multi - cultural environment associated with an increasingly global environment including many joint venture projects and customers , suppliers , and workforces spread around the globe . changes in the project management plan that result from the project risk management process may require decisions at the appropriate level of management to reassess personnel , establish or modify budgets , make commitments to others outside the project , interact with regulators , and comply with the rules of accountin

◀ 中 Synonyms

```
def extract_synonyms(row):
    model = SentenceTransformer("sentence-transformers/all-MiniLM-L6-v2")
    keywords = row["Keywords"]
    synonyms = row["Synonyms"]
    section = row["Section"]
    for keyword in keywords:
        similarity = cosine_similarity(
            np.array(model.encode([section])),
            np.array(model.encode([keyword])))
        )
        if 0.8 <= similarity <= 0.95 :
            keywords.remove(keyword)
            synonyms.append(keyword)
            print(f"moving {keyword} to synonyms list")

    row["Keywords"] = ", ".join(keywords)
    row["Synonyms"] = ", ".join(list(set(synonyms)))

    return row

df['Synonyms'] = ""
df['Synonyms'] = df.apply(lambda _: [], axis=1)
df = df.apply(extract_synonyms, axis=1)
df
```

moving project risk management practice to synonyms list
moving project risk management recognized to synonyms list
moving project risk management defined to synonyms list
moving definition project risk management to synonyms list
moving objectives project risk management to synonyms list
moving project project risk management to synonyms list
moving project risk management requires to synonyms list
moving risks project risk management to synonyms list
moving success factors project risk to synonyms list
moving project risk management to synonyms list
moving definition project risk given to synonyms list
moving definition project risk encompasses to synonyms list
moving definition project risk to synonyms list
moving project objectives individual risk to synonyms list
moving project risk levels to synonyms list
moving overall project risk represents to synonyms list
moving risk attitudes project stakeholders to synonyms list
moving role project risk management to synonyms list
moving project risk management specific to synonyms list
moving project risk management process to synonyms list
moving project risk management activities to synonyms list
moving project risk management process to synonyms list
moving determined project risk management to synonyms list
moving project risk management processes to synonyms list
moving project risk management step to synonyms list
moving step project risk management to synonyms list
moving project risk management implementation to synonyms list
moving plan risk management process to synonyms list
moving project risk management plan to synonyms list
moving project risk management carried to synonyms list
moving project risk management process to synonyms list
moving required project risk management to synonyms list
moving managing project risk to synonyms list
moving risk management plan plan to synonyms list
moving identified project risk relate to synonyms list
moving risk relate project objective to synonyms list
moving identified project risk to synonyms list
moving techniques available risk identification to synonyms list
moving historical reviews based occurred to synonyms list
moving historical review approaches to synonyms list
moving current assessment techniques rely to synonyms list
moving current assessments rely detailed to synonyms list
moving review approaches current assessment to synonyms list
moving identify risks process recorded to synonyms list
moving identify risks process to synonyms list
moving perform qualitative risk analysis to synonyms list
moving qualitative risk analysis to synonyms list
moving methods qualitative risk analysis to synonyms list
moving successful qualitative risk analysis to synonyms list
moving successful qualitative risk to synonyms list
moving quality information risks required to synonyms list
moving qualitative risk analysis to synonyms list
moving perform qualitative risk analysis to synonyms list
moving qualitative risk analysis to synonyms list
moving qualitative risk analysis process to synonyms list
moving perform qualitative risk analysis to synonyms list
moving quantitative risk analysis characteristics to synonyms list
moving quantitative risk analysis to synonyms list
moving appropriately quantitative risk analysis to synonyms list
moving calculation effect risks typically to synonyms list
moving quantitative method addresses uncertainty to synonyms list
moving uncertainty specifically methods able to synonyms list
moving data gathering to synonyms list
moving plan risk responses process to synonyms list
moving plan risk responses process to synonyms list
moving success plan risk responses to synonyms list
moving important success plan risk to synonyms list
moving responses process consider risks to synonyms list
moving risks risk response strategies to synonyms list
moving risk response strategies applied to synonyms list
moving risk response plan to synonyms list
moving risk management project team to synonyms list

moving practice standard project risk to synonyms list
 moving pmi standards member advisory to synonyms list
 moving prioritization risks linked to synonyms list
 moving prioritization risks to synonyms list
 moving techniques given to synonyms list
 moving delphi technique uses facilitated to synonyms list
 moving analysis fmea fault tree to synonyms list
 moving fmea fault tree to synonyms list
 moving force field analysis typically to synonyms list
 moving industry knowledge base special to synonyms list
 moving industry knowledge to synonyms list
 moving nominal group to synonyms list
 moving root cause analysis seeks to synonyms list
 moving example root cause analysis to synonyms list
 moving perform qualitative risk analysis to synonyms list
 moving perform qualitative risk to synonyms list
 moving perform quantitative risk analysis to synonyms list
 moving quantitative risk analysis to synonyms list
 moving perform quantitative risk to synonyms list
 moving decision tree analysis usually to synonyms list
 moving decision tree choosing to synonyms list
 moving decision tree to synonyms list
 moving expected monetary value emv to synonyms list
 moving monte carlo simulation detailed to synonyms list
 moving carlo simulation detailed to synonyms list
 moving plan risk responses to synonyms list
 moving plan risk responses given to synonyms list
 moving techniques plan risk responses to synonyms list
 moving scenario analysis risk to synonyms list
 moving risk audits carried to synonyms list
 moving objective risk reassessment ensure to synonyms list
 moving risk reassessment ensure risk to synonyms list
 moving reassessment ensure risk management to synonyms list

	Section	Text	Page	Figure	figure_text	Index
0	purpose of the practice standard for project r...	the purpose of the practice standard for p...	14	images/image14_0.png	Purpose Practice Principles of the Specializat...	1.1 f mai pm
1	project risk management defi nition	the definition of project risk management , ...	16			1.2 f mai f ma
2	role of project risk management in project man...	project risk management is not an optional a...	16			1.3 f mai p
3	good risk management practice	project risk management is a valuable compon...	17			1.4 f ma proce
4	critical success factors for project risk mana...	figure 1 - 2 . critical success factors for ...	18	images/image18_0.png	Recognize the Value of Risk Management Integra...	1.5 facti mai facti
...
173	risk audits	risk audits are carried out in order to eval...	116		d.6.1.2	f ev ma
		the objective		Otherwise_Reassess		obj

	risk reassessment	of risk reassessment	is to ens...	116	images/image116_0.png	From Plan Risk Project Ris...	d.6.1.3	reas
174	status meetings	risks can and should be on the agenda at all...		116			d.6.1.4	ris project ris
175	trend analysis	the evolution of the variance values over ti...		117			d.6.1.5	ev eval
176								

▲ Assumption Log. Described in Section 4.1.2.2. Assumptions may form inputs to the quantitative risk analysis if:

```
!pip install xlwt
import xlwt
import pandas as pd
df=pd.read_excel("df_Before_new_concepts.xls")
```

Requirement already satisfied: xlwt in /usr/local/lib/python3.10/dist-packages (1.3.0)

Concepts - Relations - Triplets

```
def Get_triplets(text):
    triplets = []
    relation, subject, relation, object_ = "", "", "", ""
    text = text.strip()
    current = "x"
    for token in (
        text.replace("<s>", "").replace("<pad>", "").replace("</s>", "").split()
    ):
        if token == "<triplet>":
            current = "t"
            if relation != "":
                triplets.append(
                    {
                        "head": subject.strip(),
                        "type": relation.strip(),
                        "tail": object_.strip(),
                    }
                )
            relation = ""
            subject = ""
        elif token == "<subj>":
            current = "s"
            if relation != "":
                triplets.append(
                    {
                        "head": subject.strip(),
                        "type": relation.strip(),
                        "tail": object_.strip(),
                    }
                )
            object_ = ""
        elif token == "<obj>":
            current = "o"
            relation = ""
        else:
            if current == "t":
                subject += " " + token
            elif current == "s":
                object_ += " " + token
            elif current == "o":
                relation += " " + token
```

```

        relation += " " + token
    if subject != "" and relation != "" and object_ != "":
        triplets.append(
            {"head": subject.strip(), "type": relation.strip(), "tail": object_.strip()})
    )
return triplets

def Get_triplets_from_text(text):

    triplet_extractor = pipeline(
        "text2text-generation",
        model="Babelscape/rebel-large",
        tokenizer="Babelscape/rebel-large",
    )
    extracted_text = triplet_extractor.tokenizer.batch_decode(
        [
            triplet_extractor(text, return_tensors=True, return_text=False)[0][
                "generated_token_ids"
            ]
        ]
    )
    extracted_triplets = Get_triplets(extracted_text[0])
    return extracted_triplets

    Interpersonal and team skills that can be used for this process include but are not limited to facilitation (see

def extract_concepts_relations(df):
    new_concepts = []
    new_relations = []
    new_objects = []
    Indexation = []

    for index, row in df.iterrows():
        print(index)
        model = SentenceTransformer("sentence-transformers/all-MiniLM-L6-v2")
        text = row["Text"]
        triples = Get_triplets_from_text(text)
        for triple in triples:
            subject = triple["head"]
            object_ = triple["tail"]
            relation = triple["type"]
            new_concepts.append(subject)
            new_relations.append(relation)
            new_objects.append(object_)
            Indexation.append(row["Index"])
        print(row["Index"])

    hierarchy = {
        "Concept": new_concepts,
        "Relation": new_relations,
        "Object": new_objects,
        "Index": Indexation,
    }
    return pd.DataFrame(hierarchy)

```

432

Part 1 - Guide

```
Hierarchy_relations = extract_concepts_relations(df)
```

```

0
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
1.1
1
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
1.2
1.2
2
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
1.3

```

1.3
3
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
1.4
1.4
4
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
1.5
5
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
1.6
1.6
6
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.1
2.1
7
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.2
8
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.3
2.3
9
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.4
10
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.5
2.5
11
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.6
2.6
12
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.7
2.7
13
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-
2.8
2.8
2.8
2.8
2.8
2.8
? *

uncertainty have the most potential impact on project outcomes. It correlates variations in project outcomes with

Hierarchy_relations

	Concept	Relation	Object	Index
0	project management	subclass of	risk management	1.1
Hierarchy_relations.to_excel("Predicates_with_Index.xls")				
...				
353	risk profile	facet of	risk	d 6 1 5
import sentence_transformers from sentence_transformers import SentenceTransformer model = SentenceTransformer('all-MiniLM-L6-v2')				
import pandas as pd import numpy as np from sklearn.metrics.pairwise import cosine_similarity				
# Iterate through new concepts and similar concepts				
for new_concept in Hierarchy_relations["Concept"]:				
similar_concepts = [concept for concept in df["Section"] if 0.7 <= cosine_similarity(np.array(model.encode([new_concept])), np.array(model.encode([concept])))) < 1]				
if similar_concepts: for similar_concept in similar_concepts: index = df.index[df["Section"] == similar_concept].tolist()[0] synonyms = df["Synonyms"][index] if isinstance(synonyms, list) and new_concept not in synonyms: synonyms.append(new_concept) else: new_row = pd.DataFrame({'Section': [new_concept], 'Synonyms': [[]]}) df = pd.concat([df, new_row], ignore_index=True)				
df				

Unnamed: 0	Section	Text	Page	Figure	figure_text	Index
0	0.0 purpose of the practice standard for project risk management	the purpose of the practice standard for project risk management	14.0	images/image14_0.png	Purpose Practice Principles of the Specializat...	1.1
1	1.0 project risk management definition	the definition of project risk management , ...	16.0	NaN	NaN	1.2
2	2.0 role of project risk management in project management	project risk management is not an optional a...	16.0	NaN	NaN	1.3
3	3.0 good risk management practice	project risk management is a valuable compon...	17.0	NaN	NaN	1.4
	critical success	figure 1 - 2 . critical			Recognize the Value of	

```
df.to_excel("df_After_new_concepts.xls")
```

```
df.to_excel("df_After_new_concepts.xls")
```

```
critical chain
```

```
df
```

	Section	Text	Page	Figure	figure_text	Index	Keywords
0	purpose of the practice standard for project risk management	the purpose of the practice standard for project risk management	14.0	images/image14_0.png	Purpose Practice Principles of the Specializat...	1.1	stan projec managen pmi proje
1	project risk management definition	the definition of project risk management , ...	16.0		Nan	Nan	projec managen projec manager
	role of project risk	project risk					projec

▼ (M) Modeling

```
df = pd.read_excel("DataframeF.xls",index_col=0)
df
```

	Section	Text	Page	Figure	figure_text	Index	Keyw
0	purpose of the practice standard for project risk management	the purpose of the practice standard for project risk management	14	images/image14_0.png	Purpose Practice Principles of the Specializat...	1.1	star project manager pmi proj...
1	project risk management definition	the definition of project risk management	16		NaN	NaN	1.2 project manager project management
2	project risk management definition	the definition of project risk management	16		NaN	NaN	1.2 project manager project management

```
Triplets_columns = df[["Concept", "Relation", "Object"]]
```

```
Triplets_columns
```

	Concept	Relation	Object
0	project management	subclass of	risk management
1	project management	practiced by	project managers
2	project managers	field of this occupation	project management
3	project planning	part of	project management
4	budgeting	part of	project management
...
367	risk	has cause	trigger conditions
368	trigger conditions	has effect	risk
369	risk profile	facet of	risk
370	earned value analysis	part of	earned value management system
371	earned value management system	has part	earned value analysis

★ Node and Edge Embeddings (TransE)

Embeddings for triplets

```
analysis analysis / cv earned
triples_array = df[['Concept', 'Relation', 'Object']].values
np.random.shuffle(triples_array)

# Split the array into training and testing sets
num_training = int(0.8 * len(triples_array))
training_triples = triples_array[:num_training]
testing_triples = triples_array[num_training:]

# Create the TriplesFactory for the training and testing sets
training_tf = TriplesFactory.from_labeled_triples(training_triples)
testing_tf = TriplesFactory.from_labeled_triples(testing_triples)

# Now we use the training triples factory (training_tf) to set up the pipeline
result = pipeline(
```

```
model='TransE',
training=training_tf,
testing=testing_tf,
training_loop='slcwa',
training_kwarg=dict(num_epochs=100),
random_seed=42,
)
trained_model = result.model

# PyKEEN models usually store embeddings in an Embedding class instance.
# The following method is a general way to access these embeddings:
entity_embeddings = trained_model.entity_representations[0](indices=None).detach().cpu().numpy()
relation_embeddings = trained_model.relation_representations[0](indices=None).detach().cpu().numpy()
```

entity embeddings

```
array([[-0.16931446, -0.09050108, -0.00443855, ..., -0.01790261,
       0.14573795,  0.17349783],
      [-0.02693522, -0.26743752,  0.08961374, ..., -0.09891997,
       0.17653653, -0.09478146],
      [-0.21073101, -0.19569781, -0.15753174, ...,  0.18795757,
       0.05537553, -0.16907728],
      ...,
      [ 0.01586115,  0.16630669,  0.17882925, ..., -0.26584014,
       0.16153954, -0.22645517],
      [ 0.04040588,  0.11425882, -0.05078923, ...,  0.2834512 ,
       -0.13226216, -0.01667796],
      [ 0.01695772, -0.01018267, -0.1195078 , ..., -0.22896765,
       0.12467522,  0.15546781], dtype='float32')
```

Training batches on cpu: 9% | 0/1 [00:00<?] ?batch/s]

relation embeddings

```
array([[-0.10599203, -0.15943967,  0.12343119, -0.06920254,  0.05212404,
       -0.06897545,  0.21250081, -0.10560171, -0.18785402, -0.2093673 ,
       -0.1326829 ,  0.18164942,  0.0337041 ,  0.07224312, -0.05849846,
       0.21348193, -0.21290834, -0.00148579,  0.02975143, -0.18701471,
       -0.15411949,  0.16134775,  0.15901563, -0.17982066, -0.16808197,
       -0.17496616, -0.20158361,  0.1011886 ,  0.0556079 , -0.01076821,
       0.06351156,  0.13893394, -0.21037324, -0.04859219,  0.10265786,
       -0.12298391,  0.20285124,  0.2206047 ,  0.07095347,  0.0660888 ,
       -0.07841165,  0.08414713,  0.18670043, -0.1202266 ,  0.0152804 ,
       0.09814979, -0.02753774,  0.0328868 , -0.13365275, -0.07773849],
      [-0.14786123, -0.10197508, -0.19055545,  0.01089966, -0.0244917 ,
       0.16926955, -0.26538733,  0.19831073,  0.1046085 ,  0.09482892,
       -0.20025115, -0.18807414, -0.08566097,  0.29208088, -0.06277578,
       -0.00278492, -0.04787325, -0.06883242,  0.02217213, -0.15436381,
       -0.15386997, -0.20526052,  0.06383024,  0.1685566 , -0.08670653,
       0.00082754, -0.05720341, -0.16035062,  0.1559182 ,  0.11433624,
       0.11739065,  0.03477395,  0.20409706, -0.01408593,  0.04519796,
       0.10934421, -0.13419995,  0.1962179 ,  0.08422501,  0.1881382 ,
       -0.06513438, -0.17260168,  0.12965669,  0.20850901,  0.00073863,
       0.13587187, -0.18278505, -0.22417729, -0.07905335,  0.13021183],
      [ 0.17619738, -0.05781661,  0.16771792, -0.1394533 ,  0.07968388,
       0.19696818,  0.02884366,  0.09526703,  0.21073025,  0.26334792,
       -0.13987021,  0.03067178, -0.18441734,  0.10641287, -0.10343745,
       -0.16313079,  0.08746406,  0.00357865, -0.0497236 ,  0.24499738,
       -0.16936041, -0.03604472, -0.17508261, -0.07809953,  0.10541454,
       0.02063126, -0.00548155,  0.25874904,  0.12605843, -0.00502149,
       0.11582848, -0.01551292, -0.07000191, -0.17413202, -0.2417584 ,
       -0.09775255, -0.071328 ,  0.12844981, -0.1417025 , -0.06825884,
       -0.18439221, -0.11392603,  0.12794358, -0.02409172, -0.13280044,
       0.13286817,  0.13729507, -0.25171894, -0.19582392,  0.10216685],
      [-0.19125909, -0.07514141,  0.1062909 , -0.12866278,  0.15794922,
       0.07629051,  0.11490281,  0.06847138, -0.13907808, -0.20556805,
       -0.14741473,  0.09855652,  0.0142998 ,  0.1584988 , -0.22318865,
       0.14857645, -0.12037501, -0.1992108 , -0.15522368,  0.01301216,
       0.00293806, -0.06217254,  0.15383825, -0.03194583,  0.0463481 ,
       -0.00378607,  0.16202402, -0.10568999,  0.13066936,  0.20403446,
       0.08108306,  0.13709691, -0.02555937, -0.06377178,  0.11675607,
       -0.2007045 , -0.04737509,  0.28980726, -0.01908982, -0.17188361,
       -0.2566112 ,  0.18544371,  0.21929933,  0.04361084, -0.00428461,
       0.18945542, -0.06888676,  0.06624039, -0.19210915, -0.19245285],
      [ 0.05379424,  0.07183735, -0.04768176,  0.10709149,  0.07919171,
       0.04454358,  0.05117799,  0.02354856,  0.09582592,  0.2150382 ,
       0.15437323,  0.08268164, -0.06977146,  0.20762663,  0.13127089,
       0.15213642, -0.03419111,  0.06706186, -0.16521737, -0.2459308 ,
       0.16872448, -0.05022024, -0.10320473, -0.04367388,  0.1539021 ,
       -0.17525971, -0.20955542,  0.10948216, -0.05665122, -0.17018698,
       0.00920147, -0.08138417, -0.06241232, -0.18579617, -0.01268298,
       0.17665799, -0.06979895, -0.22204365, -0.17301103, -0.18907124,
       0.16717322,  0.1388571 ,  0.17697607,  0.19183776,  0.0098919 ,
       0.19427557, -0.11507136,  0.13407363,  0.08041049,  0.246396 ],
      [ 0.15592043, -0.03525617, -0.0760079 , -0.01152039, -0.01517034,
       0.17692624, -0.1633284 ,  0.06874865, -0.16855484, -0.20304659,
       -0.17517935,  0.20162554,  0.14669403,  0.05702971,  0.07831637,
       0.14887308, -0.0541087 ,  0.12121201, -0.07508415,  0.15435846,
       -0.08960126,  0.15658534, -0.21061109, -0.14808203, -0.00730325,
       -0.19642732,  0.000407 ,  0.18603787,  0.08256518,  0.08554597,
       -0.14094287, -0.18673365, -0.22760636, -0.05820551, -0.07705336,
       0.06397451, -0.19301133, -0.10552552, -0.03757003,  0.06867694,
```

▼ ★ Construct the Graph

```
# Create an empty directed graph
knowledge_graph = nx.DiGraph()

# Extract concepts, relationships, and features from the DataFrame
concepts = df['Concept'].tolist()
definitions = df['Definition'].tolist()
relationships = df[['Object', 'Relation']].values.tolist()
```

```
keywords = df['Keywords'].tolist()
synonyms = df['Synonyms'].tolist()
Pages = df['Page'].tolist()
Figures = df['Figure'].tolist()
figure_text = df['figure_text'].tolist()
Indexes = df['Index'].tolist()
Text = df['Text'].tolist()
Sections = df['Section'].tolist()

# Iterate through rows in the DataFrame to add nodes with attributes
for i in range(len(concepts)):
    concept = concepts[i]
    definition = definitions[i]
    keyword = keywords[i]
    synonym = synonyms[i]
    knowledge_graph.add_node(concept, Definition=definition, Keywords=keyword, Synonyms=synonym, Section=Sect

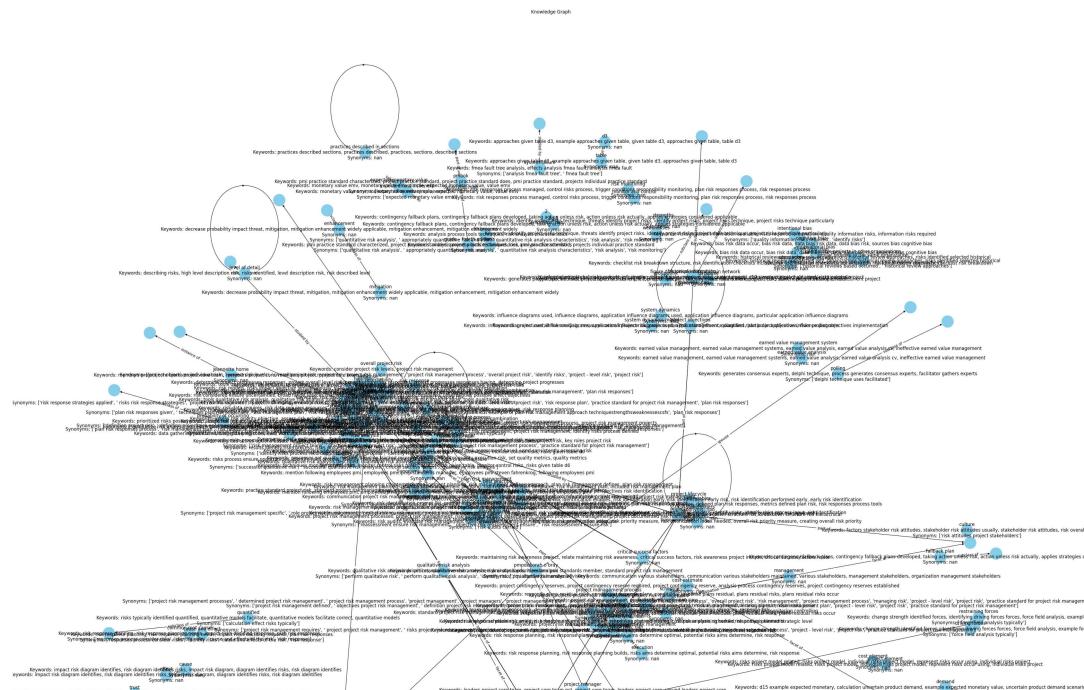
# Add edges (relationships) to the graph using the "Predicate" column
for i, (concept, relation) in enumerate(zip(concepts, relationships)):
    relation_type = relation[1]
    object_concepts = []

    if pd.notna(relation[0]):
        object_concepts = relation[0].split(', ')

    if object_concepts:
        for object_concept in object_concepts:
            knowledge_graph.add_edge(concept, object_concept, relationship=relation_type)

# Visualization
pos = nx.spring_layout(knowledge_graph)
labels = {node: f'{node}\nKeywords: {knowledge_graph.nodes[node]['Keywords']}'}\nSynonyms: {knowledge_graph.nodes[node]['Synonyms']}
edge_labels = {(edge[0], edge[1]): data['relationship'] for edge, data in knowledge_graph.edges.items()}

plt.figure(figsize=(40, 40))
nx.draw(knowledge_graph, pos, with_labels=True, labels=labels, node_size=1000, node_color='skyblue', font_size=10)
nx.draw_networkx_edge_labels(knowledge_graph, pos, edge_labels=edge_labels, font_size=10)
plt.title("Knowledge Graph")
plt.axis('off')
plt.show()
```



Saving the DGL Graph

```
G_dgl = dgl.from_networkx(knowledge_graph)
G_dgl = dgl.add_self_loop(G_dgl)
labels_G_dgl = {node: f'{node}\nKeywords: {knowledge_graph.nodes[node]['Keywords']}\\nSynonyms: {knowledge_graph.nodes[node]['Synonyms']}\\n' for node in knowledge_graph.nodes}
dgl.save_graphs("The DGL Graph", [G_dgl])
```

▼ ★ Message Passing

```

class MeanPoolingMPNN(nn.Module):
    def __init__(self, in_feats, out_feats):
        super(MeanPoolingMPNN, self).__init__()
        self.linear = nn.Linear(in_feats, out_feats)

    def message_func(self, edges):
        return {'m': edges.src['h']}

    def reduce_func(self, nodes):
        return {'h': torch.mean(nodes.mailbox['m'], dim=1)} # Compute mean instead of sum

    def forward(self, g, h):
        h = self.linear(h)
        g.ndata['h'] = h
        g.update_all(self.message_func, self.reduce_func)
        h = g.ndata.pop('h')
        return h

```

▼ GNN Model

```
# Define your model architecture with additional layers and different dropout rates
class ModelGNN(nn.Module):
    def __init__(self, in_feats, hidden_feats, out_feats):
        super(ModelGNN, self).__init__()

        self.gcn = dg1.nn.GraphConv(in_feats, hidden_feats, activation=F.relu)
```

```

self.mpnn = MeanPoolingMPNN(hidden_feats, hidden_feats)
self.gat = dgl.nn.GATConv(hidden_feats, out_feats, num_heads=1)
self.dropout1 = nn.Dropout(0.2)

def forward(self, g, features):
    h = self.gcn(g, features)
    h = self.dropout1(h)
    h = self.mpnn(g, h)
    h = self.dropout1(h)
    h = self.gat(g, h).mean(1)
    h = self.dropout1(h)
    return h

def complex_reasoning(self, g, features):
    # Implement complex reasoning for context-based information
    # Example: Aggregating information across the graph
    with g.local_scope():
        g.ndata['h'] = features
        g.update_all(fn.copy_u('h', 'm'), fn.mean('m', 'neigh'))
    return g.ndata['neigh']

# Assuming G_dgl is your DGL graph object and labels are available
YOUR FEATURE DIMENSION = 371
features = torch.rand(G_dgl.number_of_nodes(), YOUR_FEATURE_DIMENSION)
labels = torch.randint(0, 2, (G_dgl.number_of_nodes(),))

train_indices, test_indices = train_test_split(range(G_dgl.number_of_nodes()), test_size=0.2)
train_mask = torch.zeros(G_dgl.number_of_nodes(), dtype=torch.bool)
test_mask = torch.zeros(G_dgl.number_of_nodes(), dtype=torch.bool)
train_mask[train_indices] = True
test_mask[test_indices] = True

model = ModelGNN(in_feats=YOUR_FEATURE_DIMENSION, hidden_feats=128, out_feats=2) # Increased model complexity

criterion = nn.CrossEntropyLoss()
# After defining your optimizer, you can include weight decay:
optimizer = optim.Adam(model.parameters(), lr=0.0001, weight_decay=1e-5)

# Replace this with loading your actual features
features = torch.rand(G_dgl.number_of_nodes(), YOUR_FEATURE_DIMENSION)
labels = torch.randint(0, 2, (G_dgl.number_of_nodes(),))

# Split the data into training and test sets
train_indices, test_indices = train_test_split(range(G_dgl.number_of_nodes()), test_size=0.1)
train_mask = torch.zeros(G_dgl.number_of_nodes(), dtype=torch.bool)
test_mask = torch.zeros(G_dgl.number_of_nodes(), dtype=torch.bool)
train_mask[train_indices] = True
test_mask[test_indices] = True

# Training loop
num_epochs = 2000
for epoch in range(num_epochs):
    model.train()
    optimizer.zero_grad()
    logits = model(G_dgl, features.float())
    loss = criterion(logits[train_mask], labels[train_mask])
    loss.backward()
    optimizer.step()
    if epoch%100 == 0:
        print(f'Epoch {epoch+1}/{num_epochs}, Loss: {loss.item()}')

# Evaluation with additional metrics
model.eval()
with torch.no_grad():

```

```

logits = model(G_dgl, features.float())
predicted_probs = torch.softmax(logits[test_mask], dim=1)
predicted_classes = torch.argmax(predicted_probs, dim=1)

accuracy = accuracy_score(labels[test_mask].numpy(), predicted_classes.numpy())
precision = precision_score(labels[test_mask].numpy(), predicted_classes.numpy())
recall = recall_score(labels[test_mask].numpy(), predicted_classes.numpy())
f1 = f1_score(labels[test_mask].numpy(), predicted_classes.numpy())

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1 Score: {f1}')
inferred_info = model.complex_reasoning(G_dgl, features)

Epoch 1/2000, Loss: 0.7203724384307861
Epoch 101/2000, Loss: 0.5720620155334473
Epoch 201/2000, Loss: 0.4202962815761566
Epoch 301/2000, Loss: 0.3626067340373993
Epoch 401/2000, Loss: 0.3412257134914398
Epoch 501/2000, Loss: 0.3151753544807434
Epoch 601/2000, Loss: 0.302865594625473
Epoch 701/2000, Loss: 0.26399970054626465
Epoch 801/2000, Loss: 0.2589590847492218
Epoch 901/2000, Loss: 0.27129507064819336
Epoch 1001/2000, Loss: 0.24174940586090088
Epoch 1101/2000, Loss: 0.25519317388534546
Epoch 1201/2000, Loss: 0.244960218667984
Epoch 1301/2000, Loss: 0.2155415415763855
Epoch 1401/2000, Loss: 0.20384465157985687
Epoch 1501/2000, Loss: 0.23759658634662628
Epoch 1601/2000, Loss: 0.18419073522090912
Epoch 1701/2000, Loss: 0.18138006329536438
Epoch 1801/2000, Loss: 0.17888082563877106
Epoch 1901/2000, Loss: 0.19283591210842133
Accuracy: 0.7727272727272727
Precision: 0.8461538461538461
Recall: 0.7857142857142857
F1 Score: 0.8148148148148148

```

Saving GNN model

```

# Print model's state_dict
print("Model's state_dict:")
for param_tensor in model.state_dict():
    print(param_tensor, "\t", model.state_dict()[param_tensor].size())

# Print optimizer's state_dict
print("Optimizer's state_dict:")
for var_name in optimizer.state_dict():
    print(var_name, "\t", optimizer.state_dict()[var_name])

torch.save(model.state_dict(), 'model_state_dict.pth')

Model's state_dict:
gcn.weight      torch.Size([371, 128])
gcn.bias        torch.Size([128])
mpnn.linear.weight      torch.Size([128, 128])
mpnn.linear.bias        torch.Size([128])
gat.attn_l      torch.Size([1, 1, 2])
gat.attn_r      torch.Size([1, 1, 2])
gat.bias        torch.Size([2])
gat.fc.weight    torch.Size([2, 128])
Optimizer's state_dict:
state    {0: {'step': tensor(2000.), 'exp_avg': tensor([[ 2.5289e-04,  6.1868e-04,  1.2899e-04, ... 3.0162e-08, -1.1143e-03], [ 6.8252e-04,  6.9953e-04,  2.3609e-04,  ..., -1.4258e-03, -5.3453e-07, -1.3748e-03], ... ])}

```

```
[ 2.8253e-04,  4.4094e-04,  3.8074e-05,  ..., -4.5938e-04,
-4.4230e-07,  1.2667e-04],
...,
[ 6.3731e-04,  2.8886e-04,  5.9535e-04,  ..., -8.3253e-04,
-2.2292e-08, -5.8562e-05],
[-7.9399e-05,  1.2043e-04, -2.2913e-04,  ...,  1.8658e-05,
3.6083e-07, -9.1161e-04],
[ 1.1844e-03,  1.0081e-03,  9.8796e-06,  ..., -1.8063e-03,
2.9476e-07, -1.4303e-03]], 'exp_avg_sq': tensor([[4.8973e-05, 2.8761e-05, 4.2102e-05, ...
3.5183e-05],
[3.9239e-05, 2.4857e-05, 2.3688e-05,  ..., 5.2214e-05, 4.6693e-10,
3.5058e-05],
[4.5990e-05, 2.9089e-05, 3.2420e-05,  ..., 6.6955e-05, 4.7980e-10,
3.4847e-05],
...,
[4.8446e-05, 2.8193e-05, 3.0451e-05,  ..., 6.3895e-05, 5.6568e-10,
2.1872e-05],
[4.9654e-05, 3.2437e-05, 3.4832e-05,  ..., 7.4065e-05, 5.4212e-10,
1.9473e-05],
[4.3795e-05, 2.6222e-05, 3.0329e-05,  ..., 6.5318e-05, 4.5338e-10,
4.8128e-05]]}, 1: {'step': tensor(2000.), 'exp_avg': tensor([ 1.4881e-03,  8.7707e-04, -6.
9.8487e-05, -5.5247e-04,  2.0296e-03,  1.1381e-05,  4.2402e-03,
-1.4971e-03,  8.1934e-03, -1.0489e-03,  4.9538e-04,  8.3519e-04,
1.3709e-04, -9.4471e-04,  5.0833e-03, -2.1953e-03, -9.4148e-09,
3.1678e-04,  2.2518e-05,  3.9836e-03, -4.0100e-04, -2.5547e-03,
4.0028e-04, -2.2114e-03,  3.7626e-03, -1.4321e-08, -2.8137e-03,
-1.9478e-03,  4.6073e-03, -1.6372e-09,  1.3478e-03, -2.8092e-04,
1.8454e-03,  4.4431e-04,  2.1410e-03, -5.1571e-04,  5.4196e-11,
-4.3182e-04,  1.1246e-03, -4.2846e-03, -6.0923e-04, -6.2892e-04,
0.0000e+00, -1.5359e-03,  1.3120e-04,  1.5037e-03,  2.2969e-06,
3.2599e-03, -7.5292e-04,  1.7992e-03, -2.3366e-08, -6.3890e-04,
4.5543e-03,  8.9489e-04, -1.2279e-03,  6.2672e-04,  0.0000e+00,
2.2817e-03, -9.5113e-04,  6.2071e-04, -7.5590e-04, -1.1179e-03,
-1.9194e-03,  4.0889e-04,  3.6553e-04,  6.2427e-04,  7.5563e-04,
9.5305e-05, -1.1992e-03, -2.6155e-03, -8.5877e-04,  8.9194e-04,
-1.8873e-03, -5.2208e-04, -6.8735e-04,  0.0000e+00, -1.6256e-08,
-1.1050e-04,  1.2453e-09,  3.1444e-04, -6.2935e-04, -4.6266e-04,
2.9124e-03, -9.6963e-04,  0.0000e+00,  8.9451e-04,  6.3480e-04,
-1.1148e-03, -1.4162e-03, -3.5313e-03, -2.3821e-04,  4.4339e-04,
1.4150e-03,  1.3146e-03, -9.7239e-05,  1.7285e-03, -1.7188e-03,
2.2280e-04,  9.0113e-04, -1.9440e-03, -1.4817e-03, -1.6467e-03,
0.0000e+00, -3.2340e-04, -3.5140e-03,  9.0671e-03,  2.6392e-04,
-1.3160e-04, -4.0443e-04,  0.0000e+00, -4.4308e-03, -1.6193e-03.
```

▼ Recommendation System

```
def recommend(query, k=5, query_type='Concept'):
    # Map the query type to the corresponding column in the DataFrame
    query_column = {
        'Concept': 'Concept',
        'Keyword': 'Keywords',
        'Synonym': 'Synonyms'
    }[query_type]

    # Find the index of the query in the specified column
    try:
        query_index = df[df[query_column].str.contains(query, case=False, na=False)].index[0]
    except IndexError:
        print(f"{query} not found in the {query_type} column.")
        return

    # Calculate cosine similarity between the query and all nodes
    query_embedding = model(G_dgl, features.float())[query_index].detach().numpy().reshape(1, -1)
    all_embeddings = model(G_dgl, features.float()).detach().numpy()
    similarities = cosine_similarity(query_embedding, all_embeddings)[0]
```

```
# Get indices of k nodes with the highest similarity
top_indices = similarities.argsort()[-k:][::-1]

# Display recommendations
print(f'Recommendations for '{query}' ({query_type}):')
for idx in top_indices:
    print(f'{concepts[idx]} - Similarity: {similarities[idx]}')
print()

recommend('project risk management', k=5, query_type='Concept')

Recommendations for 'project risk management' (Concept):
project risk management - Similarity: 1.0
brainstorming - Similarity: 0.9999973773956299
project manager - Similarity: 0.9999054074287415
decision tree - Similarity: 0.9998677968978882
project risk management - Similarity: 0.9998677968978882
```

Embedding the query with Bert

```
from networkx.algorithms.bipartite.basic import connected_components
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np
from transformers import AutoTokenizer, AutoModel
import torch

# Load a pre-trained BERT model and tokenizer
model_name = "bert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_name)
modelE = AutoModel.from_pretrained(model_name)

def recommend(query, k=5, query_type='Concept'):
    # Map the query type to the corresponding column in the DataFrame
    query_column = {
        'Concept': 'Concept',
        'Keyword': 'Keywords',
        'Synonym': 'Synonyms'
    }[query_type]

    # Iterate through DataFrame concepts to find the most similar concept to the query
    max_similarity = -1
    most_similar_concept_index = -1
    query_token=tokenizer(query, return_tensors="pt", padding=True, truncation=True)
    query_embedding = modelE(**query_token).last_hidden_state.mean(dim=1)
    for index, row in df.iterrows():
        concept = row[query_column]
        if pd.isna(concept): # Skip NaN values
            continue
        concept_token=tokenizer(concept, return_tensors="pt", padding=True, truncation=True)
        with torch.no_grad():
            concept_embedding = modelE(**concept_token).last_hidden_state.mean(dim=1)

        similarity = cosine_similarity(query_embedding.detach().numpy(), concept_embedding.detach().numpy())
        if similarity > max_similarity:
            max_similarity = similarity
            most_similar_concept_index = index

    if most_similar_concept_index == -1:
        print(f"No similar {query_type} found for query: {query}")
        return

    # Use the embedding of the most similar concept
```

```
query_emb = model(G_dgl, features.float())[most_similar_concept_index].detach().numpy().reshape(1, -1)
all_embeddings = model(G_dgl, features.float()).detach().numpy()

# Calculate cosine similarity
similarities = cosine_similarity(query_emb, all_embeddings)[0]

# Get indices of k nodes with the highest similarity
top_indices = similarities.argsort()[-k:][::-1]

# Display recommendations
print(f"Recommendations for '{query}' ({query_type}):")
for idx in top_indices:
    print(f"{concepts[idx]} - Similarity: {similarities[idx]}")
print()

recommend('project risk management', k=5, query_type='Concept')

Recommendations for 'project risk management' (Concept):
project risk management - Similarity: 1.0
brainstorming - Similarity: 0.9999973773956299
project manager - Similarity: 0.9999054074287415
decision tree - Similarity: 0.9998677968978882
project risk management - Similarity: 0.9998677968978882
```

```
recommend('pr risk', k=5, query_type='Concept')

Recommendations for 'pr risk' (Concept):
risk - Similarity: 1.0
feasible - Similarity: 0.9999673962593079
project manager - Similarity: 0.9999673962593079
project manager - Similarity: 0.9764294624328613
enhancement - Similarity: 0.9725056290626526
```