📄 Résumeur Intelligent de Documents Business PDF avec Transformers (Fine-tuning Bart)

🎯 Objectif du projet

Ce projet vise à concevoir un système capable de **résumer automatiquement le contenu de documents PDF** grâce à des modèles de type **Transformers**. Il s'adresse à des cas d'usage concrets tels que :

- La lecture rapide de **rapports d'entreprise**, **comptes rendus** ou **études sectorielles**
- La synthèse de documents volumineux pour **gagner du temps**
- L'intégration dans un outil de **veille, d'archivage ou de documentation interne**

Ce projet a été réalisé dans une logique d'apprentissage, avec l'objectif de démontrer :

- La maîtrise des bibliothèques NLP modernes (`transformers`, `pdfplumber`, etc.)
- La compréhension des modèles de résumé génératif (type BART, T5)
- La capacité à construire un **pipeline complet** et fonctionnel

---

🧠 Contexte technique

Les documents PDF représentent un format standard en entreprise mais peu structuré. Leur traitement automatique impose plusieurs défis :

- Extraire correctement le **texte** brut du fichier (éviter les sauts de lignes inutiles, gérer les tableaux)
- Respecter les **limites de longueur** des modèles de NLP (token limits)
- Fournir un **résumé fluide, cohérent et fidèle** au contenu initial

Pour cela, ce projet s'appuie sur :

- `pdfplumber` pour l'extraction de texte
- `facebook/bart-large-cnn` via la librairie `transformers` de Hugging Face pour le résumé
- Un découpage (chunking) du document si nécessaire

📦 Pipeline global

PDF → Extraction texte → Découpage (si besoin) → → Résumé (par chunk)

Importation des Bibliothèques

```
# Installer le nécessaire pour le projet
# !pip install pdfplumber
# !pip install ipywidgets
# !pip install langchain_community
# !pip install pypdf
# !pip install kagglehub
# !pip install pandas
# !pip install datasets
# !pip install transformers[torch]
# !pip install tensorboard
# !pip install evaluate
# !pip install nltk
# !pip install rouge-score
# !pip install absl-py
```

```
# 🔹 Extraction et manipulation de documents
from langchain_community.document_loaders import PyPDFLoader
from bs4 import BeautifulSoup
import re

# 🔹 Traitement de données
import pandas as pd
import numpy as np

# 🔹 Évaluation NLP
import evaluate

# 🔹 Traitement du langage naturel
import nltk
nltk.download("punkt", quiet=True)
nltk.download('punkt_tab', quiet=True)

# 🔹 Deep learning
import torch
import tensorboard

# 🔹 Pour installer le dataset
import kagglehub

# 🔹 Transformers et datasets Hugging Face
from transformers import (
    set_seed,
    AutoTokenizer, AutoModelForSeq2SeqLM,
    BartForConditionalGeneration, BartTokenizer,
    DataCollatorForSeq2Seq,
    Seq2SeqTrainingArguments, Seq2SeqTrainer
```

```
)

from datasets import load_dataset, DatasetDict, Dataset
```

```
WARNING:tensorflow:From c:\Users\tanto\anaconda3\Lib\site-packages\tf_keras\src\losses.p
```

```
seed = 42  # Choisis n'importe quelle valeur fixe
set_seed(seed)

# Redondant mais utile pour bien figer tous les niveaux
torch.manual_seed(seed)
torch.cuda.manual_seed_all(seed)
```

🧰 Fonctions utilitaires

```
def clean_special_characters(text):
    """
    Nettoie les caractères spéciaux du texte en remplaçant les caractères non imprimables
    et les espaces insécables par des espaces normaux.
    """
    text = text.replace("\xa0", " ")  # espace insécable
    text = text.replace("\u200b", "")  # zero-width space
    return text

def normalize_whitespace(text):
    """
    Normalise les espaces dans le texte en remplaçant les espaces multiples par un seul espa
    """
    # Utilise une expression régulière pour remplacer les espaces multiples par un seul espa
    return " ".join(text.split())

def remove_code_blocks(text):
    """
    Supprime les blocs de code délimités par des backticks (```) dans le texte.
    """
    # Utilise une expression régulière pour trouver et supprimer les blocs de code
    return re.sub(r"```.*?```", "", text, flags=re.DOTALL)

def clean_text(text):
    """
    Nettoie le texte en supprimant les caractères spéciaux, les blocs de code,
    et en normalisant les espaces.
    """
    text = clean_special_characters(text)
```

```
    text = remove_code_blocks(text)
    text = normalize_whitespace(text)
    return text
```

```
def preprocess_function(examples, tokenizer):
    """
    Prétraite les exemples en entrée pour la tâche de résumé.

    Args:
        examples (dict): Un dictionnaire contenant les données d'entrée.
        tokenizer (transformers.PreTrainedTokenizer): Le tokenizer utilisé pour l'encodage.

    Returns:
        dict: Un dictionnaire contenant les entrées et les étiquettes tokenisées.
    """

    model_inputs = tokenizer(examples['article'], max_length=1024, truncation=True)

    # Setup the tokenizer for targets
    labels = tokenizer(text_target=examples['highlights'], max_length=128, truncation=True)

    model_inputs['labels'] = labels['input_ids']
    return model_inputs
```

📥 Chargement et extraction du contenu PDF

```
# # Pour charger un fichier PDF ou TXT de manière interactive dans un notebook Jupyter
# import ipywidgets as widgets
# from IPython.display import display

# upload = widgets.FileUpload(accept='.pdf,.txt', multiple=False)  # accepter PDF et TXT

# path = display(upload)
```

```
try:
    # Charger le fichier PDF
    file_path = "18113_LESSON NOTE ON BUSINESS DOCUMENTS.pdf"
    loader = PyPDFLoader(file_path, mode = "single")
    pages = []
    async for page in loader.alazy_load():
        pages.append(page)
except FileNotFoundError:
    print("Le fichier PDF n'a pas été trouvé. Veuillez vérifier le chemin du fichier.")
    # Si le fichier n'est pas trouvé, vous pouvez définir un chemin par défaut ou demander à
```

```
# Extraire le contenu de la page
doc = pages[0].page_content
```

```
print(doc)
```

Afficher la sortie masquée

🧠 Génération du résumé avec un modèle Transformer (Fine-Tuning)

```
# Download latest version
path = kagglehub.dataset_download("banuprakashv/news-articles-classification-dataset-for-nlp

print("Path to dataset files:", path)
```

```
Path to dataset files: /home/nvidia/.cache/kagglehub/datasets/banuprakashv/news-articles
```

```
df = pd.read_csv(path+"/business_data.csv")
```

```
df
```

	headlines	description	content	url	cate
0	Nirmala Sitharaman to equal Morarji Desai's re...	With the presentation of the interim budget on...	Sitharaman, the first full-time woman finance ...	https://indianexpress.com/article/business/bud...	busi
1	'Will densify network, want to be at least no....	'In terms of market share, we aim to double it...	The merger of Tata group's budget airlines Air...	https://indianexpress.com/article/business/avi...	busi
2	Air India group to induct an aircraft every si...	Air India currently has 117 operational aircra...	The Air India group plans to induct one aircra...	https://indianexpress.com/article/business/avi...	busi
	Red Sea woes:	Rising attacks	Indian		

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   headlines    2000 non-null   object
 1   description  2000 non-null   object
 2   content      2000 non-null   object
 3   url          2000 non-null   object
 4   category     2000 non-null   object
dtypes: object(5)
memory usage: 78.2+ KB
```

```
print(df["headlines"][0])
```

```
Nirmala Sitharaman to equal Morarji Desai’s record with her sixth straight budget
```

```
print(df["description"][0])
```

```
With the presentation of the interim budget on February 1, Nirmala Sitharaman will surpa
```

```
print(df["content"][0])
```

```
Sitharaman, the first full-time woman finance minister of the country, has presented fiv
With the presentation of the interim budget on February 1, Sitharaman will surpass the r
Desai, as finance minister, had presented five annual budgets and one interim budget bet
ADVERTISEMENT
As the Parliamentary elections are due, Sitharaman’s interim budget may not contain any
A vote-on-account, once approved by Parliament, will authorise the government to withdra
The new government, which is likely to be formed around June, will come up with a final
After the Modi government came to power in 2014, Arun Jaitley took charge of the finance
ADVERTISEMENT
Piyush Goyal, who was holding the additional charge of the ministry due to ill health of
After the 2019 general elections, in the Modi 2.0 Government, Sitharaman was given the c
That year, Sitharaman did away with the traditional budget briefcase and instead went fo
ADVERTISEMENT
Under Sitharaman, India has weathered the Covid pandemic with an array of policy measure
India is racing to become a USD 5 trillion economy by 2027-28 and USD 30 trillion by 204
The first budget of Independent India was presented by the first finance minister R K Sh
Sitharaman, who will be presenting her sixth budget in a row, is expected to come up wit
```

```
Rakesh Nangia, Chairman, Nangia Andersen India said given the proximity to the elections
ADVERTISEMENT
In the last interim budget for FY 2019-20, while the overall tax structure remained unch
```

```
any(df.isnull())
```

```
True
```

Il y a des Nan donc des cas vides.

Mais pour nous présentement, notre travail se concentre sur **description** et **content**

```
any(df["description"].isnull())
```

```
False
```

```
any(df["content"].isnull())
```

```
False
```

```
ad = (df["content"].apply(lambda x: len(x)))
```

longueur moyenne et longueur mediane des textes

```
ad.mean() ,ad.median()
```

```
(np.float64(1650.582), np.float64(1188.0))
```

```
# Charger le tokenizer et le modèle BART pour la génération de résumés
tokenizer = AutoTokenizer.from_pretrained("facebook/bart-large-cnn")
model = AutoModelForSeq2SeqLM.from_pretrained("facebook/bart-large-cnn")
```

```
# Charger le dataset à partir du fichier CSV
dataset = load_dataset("csv", data_files=path+"/business_data.csv",)
```

```
# Prétraiter le dataset
dataset = dataset.remove_columns(['headlines', 'url', 'category'])
dataset = dataset.rename_column('content','article')
dataset = dataset.rename_column('description', 'highlights')
```

```
dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['highlights', 'article'],
        num_rows: 2000
    })
})
```

```
# 1. Découpe initiale : train (80%) + temp (20%)
split_dataset = dataset["train"].train_test_split(test_size=0.2, seed=seed)

# 2. Découpe de temp en val (10%) + test (10%)
# Ce .test contient les 20%, on les redécoupe à 50/50 → 10% + 10%
temp_split = split_dataset["test"].train_test_split(test_size=0.5, seed=seed)

# 3. Regrouper dans un nouveau DatasetDict
final_dataset = DatasetDict({
    "train": split_dataset["train"],       # 80%
    "validation": temp_split["train"],     # 10%
    "test": temp_split["test"],            # 10%
})

# Vérification
print(final_dataset)
```

```
DatasetDict({
    train: Dataset({
        features: ['highlights', 'article'],
        num_rows: 1600
    })
    validation: Dataset({
        features: ['highlights', 'article'],
        num_rows: 200
    })
    test: Dataset({
        features: ['highlights', 'article'],
        num_rows: 200
    })
})
```

```
# Prétraiter le dataset pour la tâche de résumé
# Utiliser la fonction preprocess_function pour tokeniser les entrées et les étiquettes
tokenized_dataset = final_dataset.map(preprocess_function, batched=True, remove_columns=["hi
```

```
#  Initialiser le DataCollator pour la tâche de résumé
data_collator = DataCollatorForSeq2Seq(tokenizer, model=model)
```

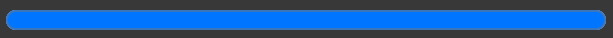
```
# Charger la métrique ROUGE pour l'évaluation
metric = evaluate.load("rouge")
```

```
def compute_metrics(eval_preds):
    preds, labels = eval_preds
    # decode preds and labels
    labels = np.where(labels != -100, labels, tokenizer.pad_token_id)
    decoded_preds = tokenizer.batch_decode(preds, skip_special_tokens=True)
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)

    # rougeLSum expects newline after each sentence
    decoded_preds = ["\n".join(nltk.sent_tokenize(pred.strip())) for pred in decoded_preds]
    decoded_labels = ["\n".join(nltk.sent_tokenize(label.strip())) for label in decoded_labe

    result = metric.compute(predictions=decoded_preds, references=decoded_labels, use_stemme
    return result
```

```
# Définir les arguments d'entraînement pour le modèle Seq2Seq
training_args = Seq2SeqTrainingArguments(
    output_dir="results",
    eval_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    weight_decay=0.01,
    save_total_limit=1,
    num_train_epochs=3,
    fp16=True,
    predict_with_generate=True,
    report_to="tensorboard",
    logging_dir="logs",
    warmup_steps=300,
    label_smoothing_factor=0.1,
    generation_max_length=1000,
    generation_num_beams=4,
    save_strategy="epoch",
)
```

```
# Initialiser le Seq2SeqTrainer avec le modèle, les arguments d'entraînement, le dataset et
trainer = Seq2SeqTrainer(
    model=model,
    args=training_args,
    train_dataset= tokenized_dataset["train"],
    eval_dataset= tokenized_dataset["validation"],
    data_collator=data_collator,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics
)
trainer.train()
```

```
/tmp/ipykernel_183588/2877648183.py:1: FutureWarning: `tokenizer` is deprecated and will
  trainer = Seq2SeqTrainer(
Passing a tuple of `past_key_values` is deprecated and will be removed in Transformers v
```



[1200/1200 04:01, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Rouge1	Rouge2	Rougel	Rougelsum
1	No log	2.292958	0.391335	0.264654	0.338040	0.350953
2	2.192200	2.277229	0.406326	0.287527	0.360051	0.370060
3	1.947600	2.309469	0.425335	0.307998	0.379620	0.390772

```
/home/nvidia/.local/lib/python3.10/site-packages/transformers/modeling_utils.py:3465: Us
  warnings.warn(
TrainOutput(global_step=1200, training_loss=2.0301630147298177, metrics=
{'train_runtime': 241.6931, 'train_samples_per_second': 19.86,
'train_steps_per_second': 4.965, 'total_flos': 6706663897890816.0, 'train_loss':
```

```
# pour visualiser les logs de l'entraînement
%load_ext tensorboard
%tensorboard --logdir ./logs
```

```
ERROR: Could not find `tensorboard`. Please ensure that your PATH
contains an executable `tensorboard` program, or explicitly specify
the path to a TensorBoard binary by setting the `TENSORBOARD_BINARY`
environment variable.
```

```
# Recharge depuis le dossier de sauvegarde
# Assurez-vous que le modèle a été sauvegardé dans le dossier "results/checkpoint-1200" ou u
model = BartForConditionalGeneration.from_pretrained("results/checkpoint-1200")
tokenizer = BartTokenizer.from_pretrained("results/checkpoint-1200")
```

```
/home/nvidia/.local/lib/python3.10/site-packages/transformers/models/bart/configuration_
  warnings.warn(
```

```
# Exemple de texte à résumer
# Utiliser le texte text2 extrait du PDF ou texte1 du dataset test
text1 = final_dataset["test"]["article"][0]
text2 = clean_text(doc)

# Tokenizer le texte
inputs = tokenizer(text1, return_tensors="pt", max_length=1024, truncation=True)
```

```
# Générer le résumé
summary_ids = model.generate(
    inputs["input_ids"],
    attention_mask=inputs["attention_mask"],
    max_length=700,
    min_length=100,
    num_beams=4,
    length_penalty=2.0,
    no_repeat_ngram_size=3,
)

# Décoder le résultat
summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
print("Résumé :", summary)
```

```
Résumé : The 30-share BSE Sensex fell sharply by 505.19 points or 0.77 per cent to close
```

```
print(text1)
```

```
Benchmark stock indices Sensex declined by 505 points at close while Nifty settled lower
The 30-share BSE Sensex fell sharply by 505.19 points or 0.77 per cent to close at 65,28
Ending its eight-day winning streak, the broader Nifty of the National Stock Exchange de
ADVERTISEMENT
Among major Sensex shares, PowerGrid fell the most by 2.76 per cent. IndusInd Bank dropp
ICICI Bank, HDFC Bank, HDFC, ITC, Infosys, L&T, Bajaj Finance, Kotak Bank, HCL Tech and
On the other hand, Tata Motors rose the most by 2.94 per cent, followed by Titan which g
Mahindra & Mahindra, SBI and TCS were also among gainers.
ADVERTISEMENT
“The domestic market succumbed to profit-booking as heat waves from weak global markets
In the broader market, the BSE Midcap declined by 0.76 per cent to 28,999.02 while BSE S
All the BSE sectoral indices except for auto and consumer durables ended in the red with
ADVERTISEMENT
In global markets, Hong Kong, China, Japan and Australia sank up to 1.7 per cent followi
Investors feared that as a sturdy labour market keeps the economy out of a long-feared r
Global oil benchmark Brent crude climbed 0.25 per cent to USD 76.70 a barrel.
Defying a weak trend in the global markets, the 30-share BSE Sensex climbed 339.60 point
Foreign Institutional Investors (FIIs) continued their buying activity as they bought ec
```

Conclusion

Au terme de ce notebook, nous avons mis en place un pipeline complet de fine-tuning pour une tâche de résumé automatique :

- 📄 **Chargement de documents PDF avec LangChain**
- ✂️ **Prétraitement et découpage du texte brut en segments exploitables**
- 🧩 **Construction d’un dataset compatible Hugging Face**
- 🧠 **Fine-tuning d’un modèle de type Bart sur notre jeu de données**
- 📊 **Évaluation à l'oeil nu sur du modèle**

🎯 Résultats

Le modèle entraîné est désormais capable de générer des résumés adaptés à la structure de notre dataset. Il pourra être intégré dans des pipelines de traitement de documents, notamment pour des applications de type :

- Résumé automatique de documents business PDF
- Prétraitement pour des tâches de classification ou de QA

🚀 Pistes d'amélioration

- Utiliser un dataset plus riche avec des résumés de meilleure qualité
- Amélioration du modèle entrainé à travers la recherche de paramètres optimaux

- Une étape de nettoyage des données plus profondes
- Expérimenter avec d'autres modèles (T5, Pegasus, Mistral, etc.)