Elements techniques

Déroulé du projet

PARIS DIGITAL LAB

CentraleSupélec

FUN FRANCE UNIVERSITÉ NUMÉRIQUE

Equipe et contexte du projet

Liwei SUN

Pierre-Louis VEYRENC

Léo SOUQUET
Coach PDL

Mohammed BAHHAD

Antoine VAGLIO

Léa MOUKARZEL
Coach PDL

PARIS DIGITAL LAB

CentraleSupélec a créé en 2015 un dispositif d'Open Innovation : le **Paris Digital Lab**

Des équipes d'étudiants y réalisent des **prototypes de nouveaux services numériques**, en utilisant Intelligence Artificielle, IoT, technologies Web/Mobile…

Chaque prototype est fabriqué en **7 semaines**, par une équipe d'étudiants à temps plein, coachés par des jeunes ingénieurs

En 8 ans :

- 424 étudiants formés à par cette méthode immersive
- 176 entreprises / institutions accompagnées (dont 5 Ministères)
- 372 prototypes réalisés

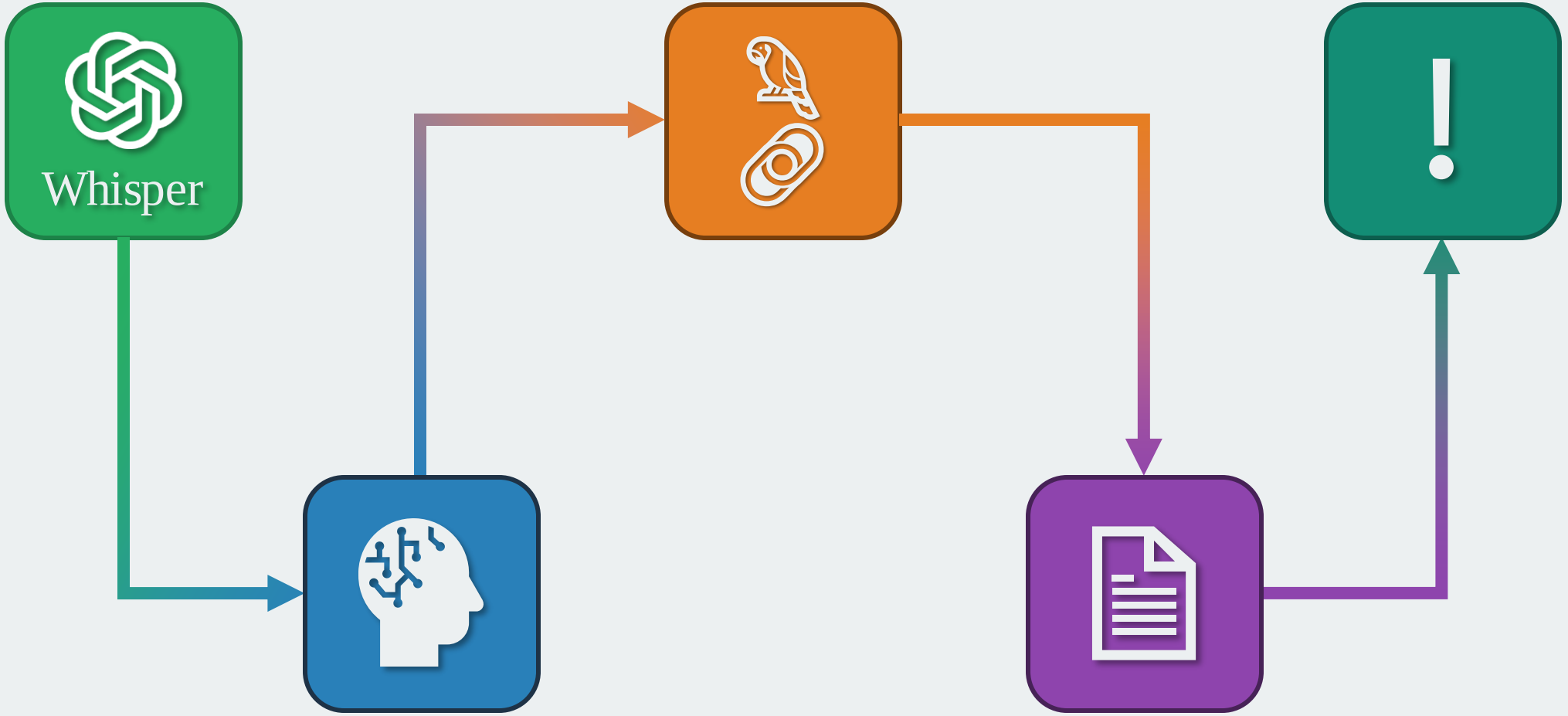
CentraleSupélec

Roadmap

I - Transcription

III – Fonctions avancées

V – Autres aspects marquants



II – Premiers usages LLM

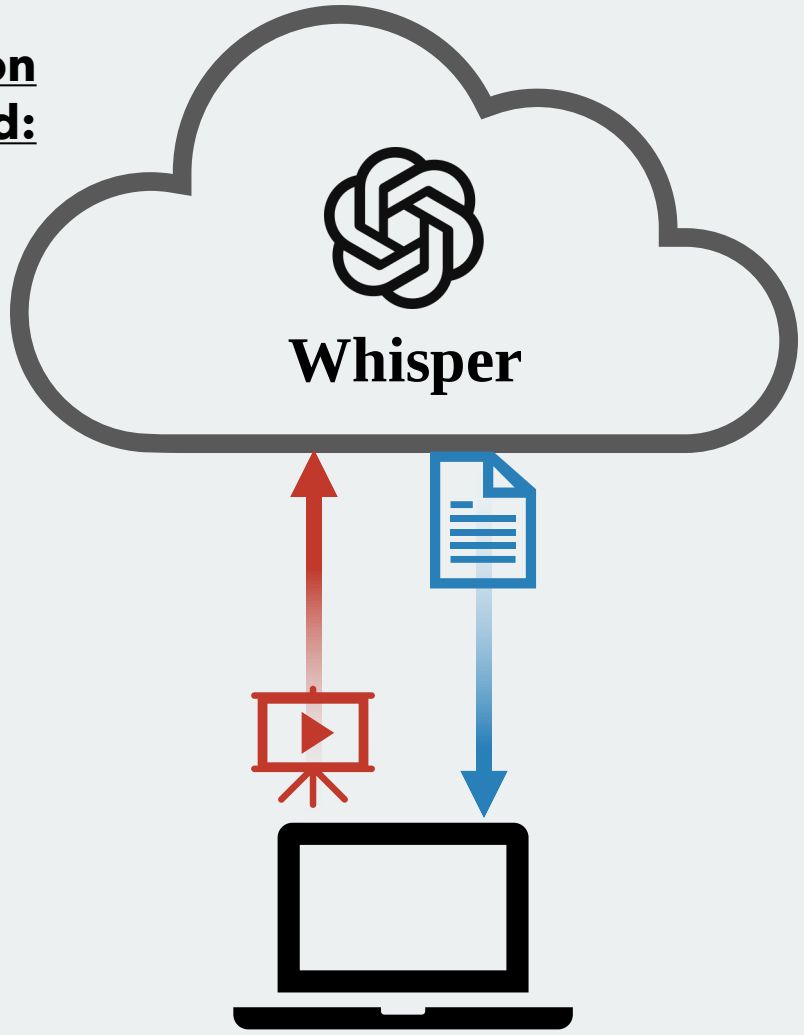
IV – Résumé

1 Transcription

Whisper

Modèle dernier cri d'OpenAI pour la transcription audio

Utilisation cloud:



Le problème:

Model	Usage
Whisper	$0.006 / minute (rounded to the nearest second)

15 cours de 12x12h = 150$

→ trop grosse partie de notre budget

Whisper

Modèle dernier cri d'OpenAI pour la transcription audio

*Le modèle est **open-source** !*

Utilisation en local

CentraleSupélec

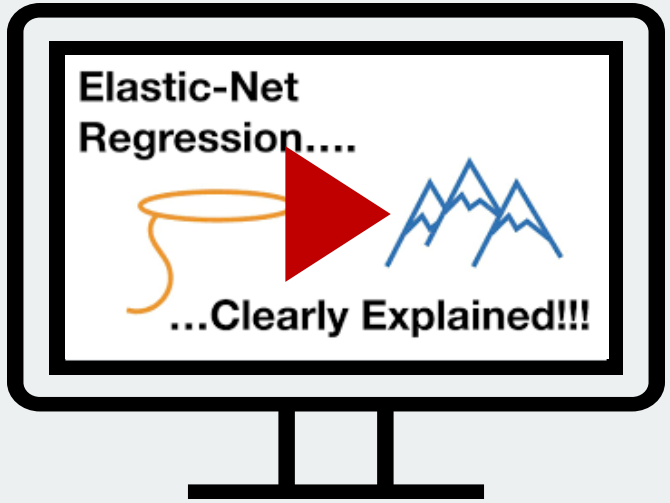
Sur une VM équipée d'une **GTX 1080Ti**
Avec le modèle **large-v2**

1h de vidéo ~= 30mn à transcrire

Plus long, mais « gratuit » !

2 Prise en main des LLM

ChatGPT: incroyable ! ...?



5min18sec

- Titre
- Description
- Tags
- Topics
- Résumé

ChatGPT: incroyable ! ...?

1h11min36sec

- ☒ Titre
- ☒ Description
- ☒ Tags
- ☒ Topics
- ☒ Résumé

Longueur du prompt

```
InvalidRequestError: This model's maximum context length is 4097 tokens, however you requested 7322 tokens (7066 in your prompt; 256 for the completion). Please reduce your prompt; or completion length.
```

La taille limite d'un prompt avec le résultat du modèle est : 4097

34315 est le nombre de tokens dans un des cours de 3h

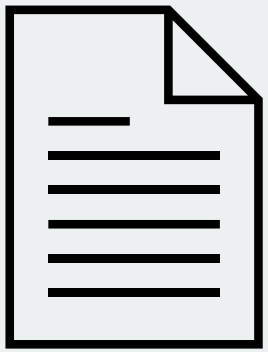
Solution :

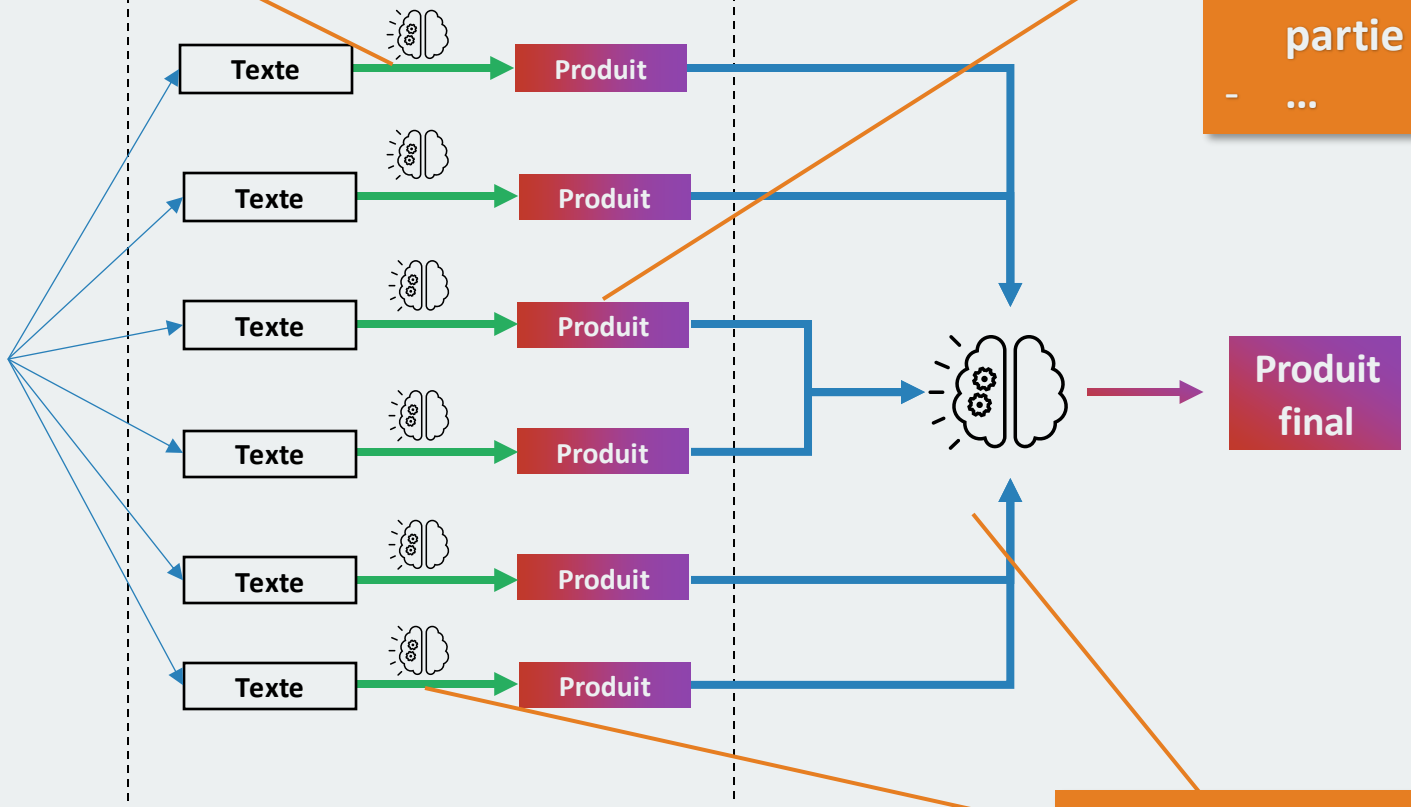
Split - Transform - Combine

Technique *Split-Transform-Combine*

Parties traitées **indépendamment**

Une petite partie du final, ex:
- **Une question de QCM**
- **Un titre de partie**
- **…**

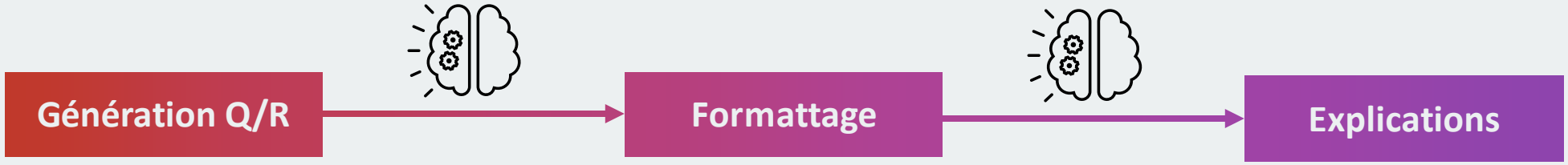




3 Fonctionnalités avancées

Génération de QCM

Fonction plus complexe



Chaîne *d'appels au LLM*

- Aussi faire **STC** là-dessus
- Gérer les ***rate limits*** de l'API
- Gérer les **prompts**, les mettre au bon endroit et bien remplir leurs paramètres
- Etc.

LangChain

Langchain

Framework Python pour créer des apps à base de LLM

Paralléliser et donc
accélérer le traitement

Utile pour créer des apps
robustes et complexes

De nombreuses fonctions
déjà intégrées
(algorithmes de résumé par exemple)

Gestion d'erreurs
(rate limit, erreurs 500 etc)

Permet de structurer
le code grâce à l'**OOP**

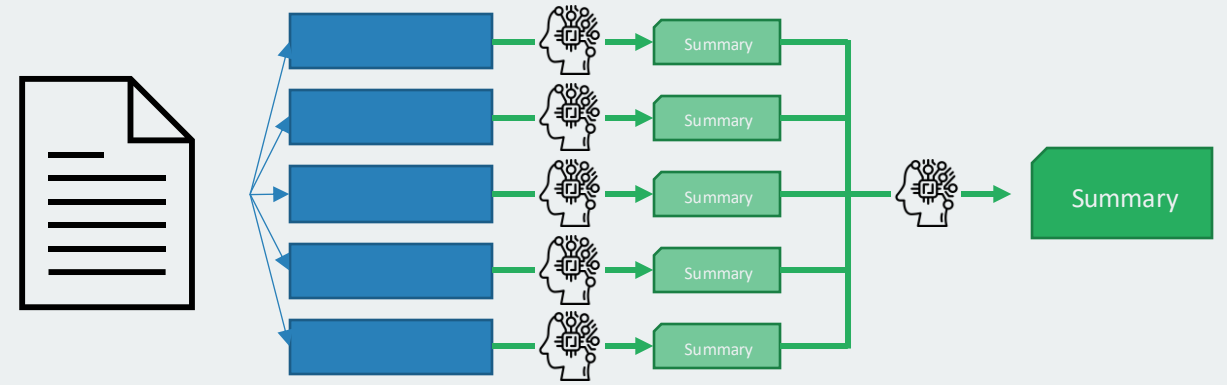
La bibliothèque est encore un peu **bêta:** elle manque de documentation, mais elle est déjà très pratique !

4 Le résumé

Algorithmes de résumé

2 principaux: ***Refine*** *et* ***Map reduce***

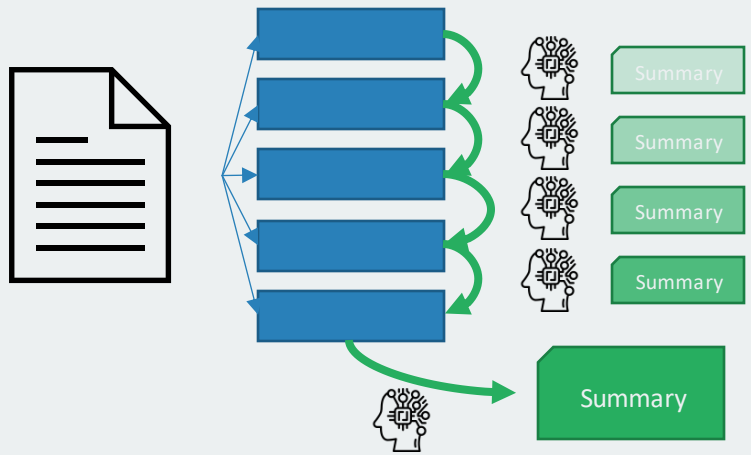
MAP REDUCE



Prompt:

Write a concise summary of the following:
{text}
CONCISE SUMMARY:

REFINE



Prompt:

Your job is to produce a final summary
We have provided an existing summary up to a certain point:
{existing_answer}
We have the opportunity to refine the existing summary
(only if needed) with some more context below.

{text}

Given the new context, refine the original summary
If the context isn't useful, return the original summary.

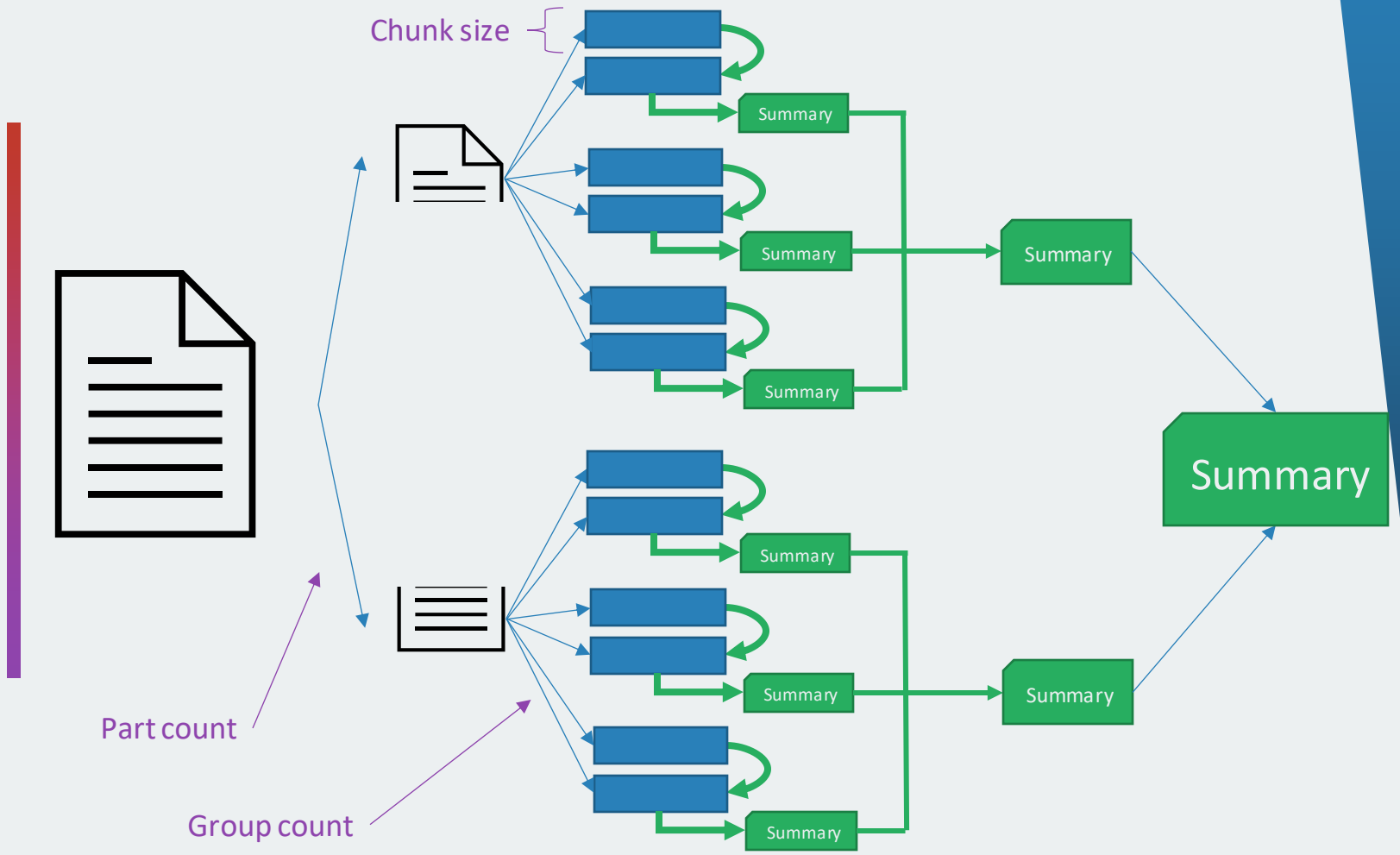
Summarisation hybride

Map Reduce:

- Facilement parallélisable
- Peut être récursif
- Minimise les appels
- Extrait des informations globales

Refinement:

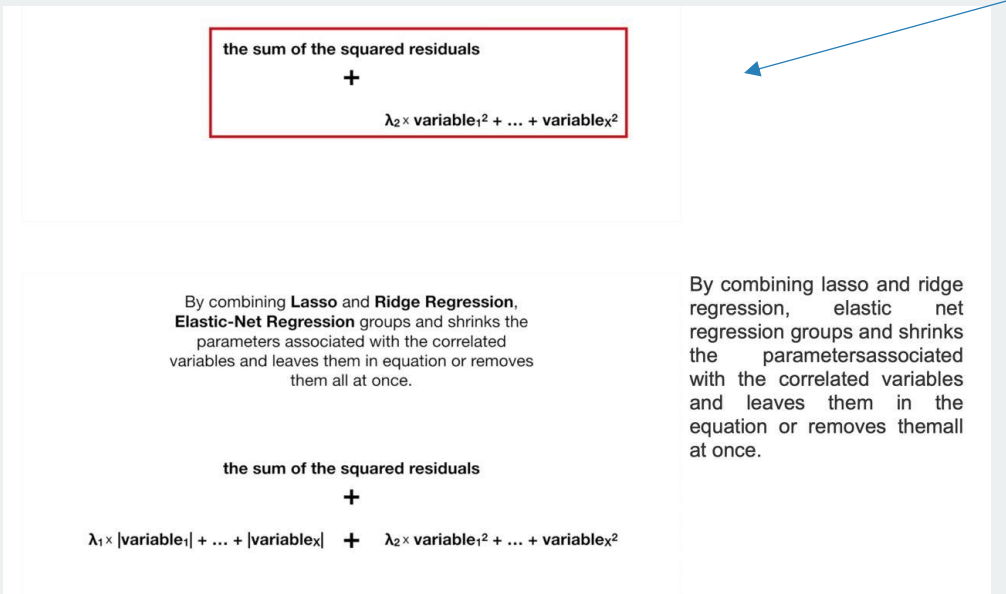
- Intrinsèquement itératif
- Limite la perte de détail
- Plus d'appels aux LLMs

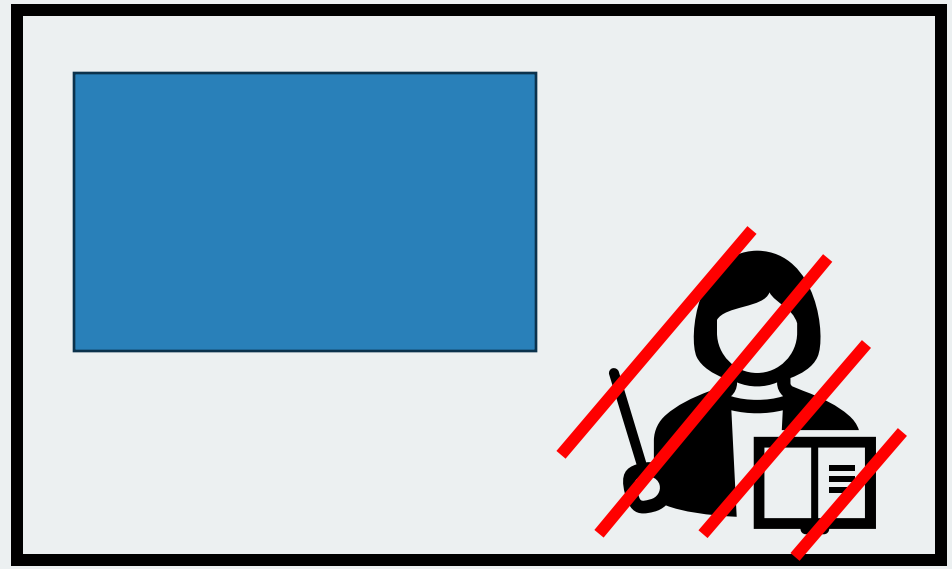


5 Autres aspects marquants

Combinaison retranscription et slides

1. Détection de **changement de slides**
2. Extraction des **timestamps**
3. Détermination des **phrases liées** à ces timestamps
4. Mise en page, et **enregistrement en pdf**





Slide transition when sudden change in non-masked area

Éviter l'hallucination de ChatGPT

- Par exemple : explication des QCMs

Tu es un professeur d'université cherchant à faire apprendre un cours de '{theme}' à ses élèves, intitulé '{title}'.

Prompt de base → **Hallucination** → **Solution**

Renvoie, pour chaque question, le numéro de la phrase qui contient la réponse, au format JSON suivant: {Q0: 8, ...}

Q5: *"Pourquoi le ciel est-il bleu ?"*
Réponse: *"A cause de la diffusion de Rayleigh"*

Dans la retranscription :
-Phrase 13: *"La couleur bleue du ciel est dûe à une phénomène physique nommé 'diffusion de Rayleigh'"*
-Phrase 14: *"L'Union Européenne a été fondée en 1993"*

Renvoi du numéro de la phrase:

Meilleur renvoi: **{Q5: 13}**

Nouveau prompt :

-Voici les questions que tu as générées: {questions}

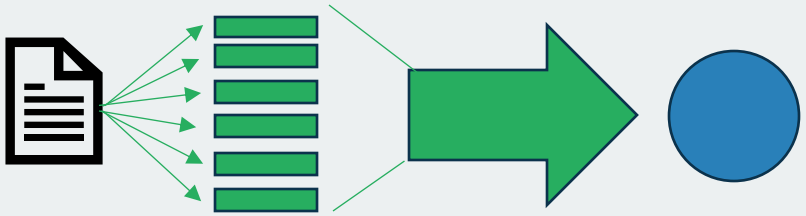
-Voici les phrases de l'extrait de cours: {sentences}

FromTopics

À partir de l'extraction des topics, on demande au LLM de générer le reste des métadonnées :

```python
def description(topic : list):
    response=openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role" : "system", "content": "La réponse que tu donnes est un texte"},
        {"role" : "user", "content" : "Je vais te donner une liste de sujet extrait d'un cours.\
                                       Je te demande de retourner un texte qui présente le cours en 6 phrases maximum. \
                                       Voici les sujets :" + str(topic)}
    ]
    )
    return response["choices"][0]["message"]["content"]
```

- Testé pour la génération de :
  - Titre
  - Description



- **Moins d'appel** par l'utilisation d'un seul appel pour générer une description par exemple.
- Moins de problème de longueur de contexte car les topics sont très courts

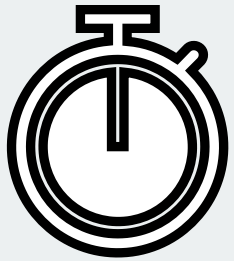
==> Compte sur la créativité du LLM pour générer des textes cohérents

6 Conclusion et perspectives

Conclusion

Pour tout générer sur **18 vidéos (~7h)**

~10$
de coûts d'API



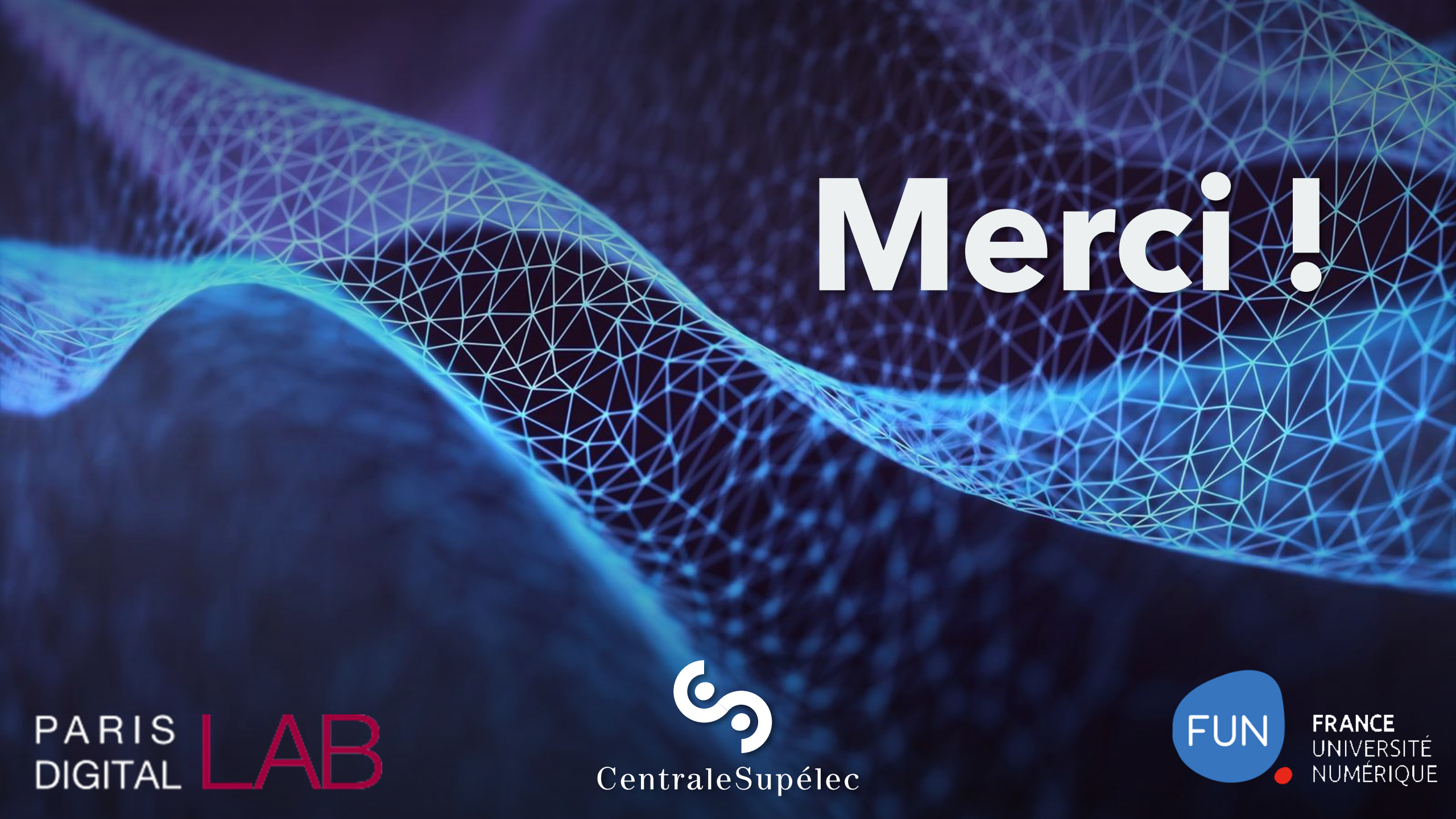
~5h
de traitement

Le facteur limitant est le temps de traitement

Lié à la *rate limit* et le temps de réponse des modèle

Perspectives d'améliorations

- **Chapitrage automatique** de la vidéo
- **Prise de notes facilitée** : extraction de petits clips, mise en place d'un système de "post-it" persistants
- **Implémentation d'un chatbot** pour améliorer l'interaction avec l'étudiant
- **QCM extensible**


Merci !
PARIS DIGITAL LAB
CentraleSupélec
FUN
FRANCE UNIVERSITÉ NUMÉRIQUE