Dahmani Anis
Gnankambary Farid
Greffe Jean-Baptiste

ING 3 IA B

Natural Language Processing

Text summarization

A l'attention de M. Hamza Assouane

2023/2024

Sommaire

Introduction

Notre projet s'est focalisé sur la tâche complexe de résumé de textes, plus spécifiquement sur la synthèse des transcriptions provenant de vidéos YouTube. Pour cela, nous avons décomposé notre travail en trois grandes parties distinctes.

Dans la première phase, nous avons concentré nos efforts sur le développement d'une fonctionnalité permettant de récupérer automatiquement la transcription d'une vidéo YouTube en utilisant une API et l'interface Streamlit. Cette fonctionnalité a été conçue de manière intuitive, permettant à l'utilisateur de simplement saisir le lien de la vidéo et d'obtenir en retour la transcription textuelle correspondante. Cette approche a facilité l'accès et l'analyse des contenus audiovisuels, transformant aisément ces données multimédias en données textuelles exploitables.

Dans une deuxième phase, nous avons entrepris la comparaison rigoureuse de différents modèles dans le domaine du traitement du langage naturel, en nous concentrant spécifiquement sur leur performance dans la tâche de résumé de textes. Cette étape a impliqué l'exploration et l'évaluation de diverses architectures de modèles, telles que LED, BART ou encore GPT, en mettant l'accent sur leur capacité à générer des résumés précis et informatifs à partir des transcriptions extraites des vidéos YouTube.

Enfin, dans la dernière phase de notre projet, nous avons utilisé le modèle présentant les performances les plus élevées pour réaliser le résumé automatique des transcriptions issues des vidéos YouTube. Cette étape a été cruciale, car elle a permis de démontrer la pertinence et l'applicabilité du modèle sélectionné pour résumer automatiquement et de manière précise des contenus audiovisuels complexes en texte concis et informatif.

L'ensemble de ces étapes a permis de proposer une solution complète, allant de l'extraction des transcriptions à partir de vidéos YouTube jusqu'à la création automatisée de résumés précis, offrant ainsi une approche robuste pour traiter et résumer efficacement des contenus multimédias variés et riches en information.

1. Etat de l'art

Il existe plusieurs méthodes pour aborder la réalisation de résumé de textes. Dans cette section, nous explorerons plusieurs de ces approches, allant des techniques non basées sur l'apprentissage automatique aux méthodes exploitant les avancées des modèles d'apprentissage automatique.

Pour commencer, nous nous sommes intéressés aux solutions non basées sur l'apprentissage automatique qui existent, telles que le résumé par extraction de phrases clés. Cette approche repose sur l'identification et l'extraction des phrases les plus importantes ou représentatives du texte source. Des approches statistiques sont également employées, comme la pondération des mots selon leur fréquence pour estimer leur importance, ou l'analyse de similarité entre phrases afin d'identifier celles ayant le plus d'importance. Ce sont des méthodes de types extractives.

Par la suite, plusieurs méthodes basées sur l'apprentissage automatique ont été développées. Notre projet s'est concentré sur l'utilisation d'un modèle pré-entraîné, que nous avons affiné par la suite. Parmi ceux-ci, nous avons utilisé le modèle LED : Longformer Encoder-Decoder. Toutefois, le champ est vaste et d'autres modèles sont également efficaces pour cette tâche, tels que :

- Les Réseaux de Neurones Récurrents (RNN) : Ces modèles sont utilisés pour modéliser la séquence de phrases et générer des résumés en prenant en compte l'ordre des mots et la séquentialité des informations.
- Les Réseaux Neuronaux Récurrents à Mémoire à Court et Long Terme (LSTM) : Ils sont une variante des RNN conçue pour capturer les dépendances à long terme, leur permettant de saisir des informations pertinentes sur de plus longues séquences.
- Les Réseaux de Neurones Convolutifs (CNN) : Souvent utilisés pour extraire des caractéristiques et des structures pertinentes dans le texte, ces modèles se concentrent sur la détection de motifs et de relations entre les mots ou groupes de mots.
- Les modèles de Transformers : Parmi eux, on trouve des modèles comme BERT et GPT (Generative Pre-trained Transformer). Ces modèles exploitent une compréhension globale du texte pour générer des résumés, en capturant les relations contextuelles entre les mots et en saisissant la structure du texte de manière efficace. Le modèle LED que nous avons utilisé en est un.

Chacune de ces approches présente ses avantages et ses spécificités, offrant ainsi un éventail de choix pour aborder la tâche complexe du résumé automatique de textes. Ce sont des méthodes abstraites ou bien "abstractives".

2. Datasets utilisés

Notre travail s'est principalement concentré sur l'exploitation du dataset Podcast Summary Assessment Data, disponible sur HuggingFace. Ce corpus de données représente une compilation de résumés de podcasts évalués par des experts humains lors du Podcast Challenge à TREC2020. Ce challenge a réuni un ensemble de spécialistes chargés d'évaluer et de noter ces résumés de podcasts, offrant ainsi un ensemble de données conséquent et évalué manuellement. Ce dataset spécifique comprend un total de 3580 exemples distincts. Chaque exemple est constitué d'une paire comprenant la transcription intégrale d'un podcast associée à son résumé correspondant, le tout étant évalué par des êtres humains experts dans le domaine. Chaque paire de données contient également un score attribué par ces évaluateurs humains. La richesse de ce corpus réside dans sa variété de sujets et de styles de podcasts, offrant ainsi une diversité de contenus allant de l'éducation à la culture en passant par les sciences, la technologie et bien d'autres domaines. De plus, le fait que ces résumés aient été évalués par des experts renforce la qualité et la fiabilité des données, en apportant une dimension subjective humaine à l'évaluation.

Nous avons choisi d'utiliser ce dernier pour réaliser le finetuning de nos modèles, car les transcriptions qu'il contenait ainsi que leurs résumés correspondaient bien à notre objectif d'apprentissage. Les transcriptions qui y étaient présentes étaient en effet très proches de celles d'une vidéo YouTube, ce qui laissait présager de meilleurs résultats qu'avec des datasets plus génériques et provenant d'écrits (journaux, articles…). Également, la présence de résumés vérifiés nous permettait d'envisager notre solution comme celle d'un apprentissage supervisé : la présence de données labellisées permettant un apprentissage et une évaluation plus efficaces.

3. Nos modèles

Inspiré par les travaux de Iz Beltagy, Matthew E. Peters et Arman Cohan, nous avons utilisé le modèle Longformer Encoder-Decoder (LED) qui est un modèle de type longformer. C'est un modèle transformers développé par the Allen Institut for Artificial Intelligence, dans le but notamment de réaliser des résumés de longs documents. Nous avons choisi ce modèle car nous l'avons trouvé intéressant et particulièrement adapté à notre sujet, la durée d'une vidéo YouTube pouvant dépasser plusieurs heures avec donc une transcription assez longue. Ce modèle, disponible sur HuggingFace, peut être finetune sur de nouveaux datasets afin d'obtenir de meilleurs résultats dans des cas d'usage spécifiques. Nous avons donc entraîné le modèle sur le dataset Podcast Summary Assessment Data, dans l'optique d'obtenir de bons résultats de summarization à partir de transcriptions vidéo ou audio. Il est à noter toutefois qu'en raison de contraintes de ressources (espace et puissance de calcul sur GPU de Colab, temps…), nous avons dû sélectionner aléatoirement quelques centaines de données de notre dataset d'entrainement initial. Cette contrainte a eu des répercussions sur nos résultats finaux.

Nous avons par la suite essayé de comparer notre modèle baseline avec d'autres modèles pré-entraînés plutôt connus dans la littérature, sans toutefois les finetuner. Nous avons alors utilisé les modèles suivants : le modèle BART et les modèles GPT-2 medium et large. Bart est un modèle avancé de traitement du langage naturel, basé sur l'architecture des transformers. Conçu par Facebook AI Research, BART combine deux aspects clés : la bidirectionnalité et l'autorégression. Le modèle GPT-2 quant à lui est un LLM développé par OpenAI. Il appartient aussi à la famille des transformers et est renommé pour sa capacité à générer du texte cohérent et réaliste. Ce modèle est pré-entraîné sur de vastes quantités de texte provenant d'Internet, lui permettant de comprendre la structure et la logique du langage naturel dans différentes situations.

Enfin, après l'entrainement de nos différents modèles et la génération de résumés exemples, nous sommes passés à l'évaluation de ces derniers. Etant donné notre tâche initiale, le résumé de texte, il nous a fallu nous intéresser à différentes métriques pour juger de la performance de nos modèles. Plus particulièrement, nous avons utilisé le ROUGE score afin de pouvoir évaluer et comparer nos différents modèles. Nous avons ainsi pu confirmer que notre modèle que nous avons finetuner sur la tâche de résumé de texte était plus performant que les modèles pré-entraînés. Nous devons aussi prendre en considération qu'avoir utilisé un dataset qui ne se basait pas exactement sur des transcriptions de vidéo YouTube mais sur des podcasts.

Pour réaliser la comparaison de nos modèles, nous nous sommes intéressés aux résumés de plusieurs transcripts produits par chacun de nos modèles : LED que nous avons finetuné, GPT2 et BART que nous avons utilisé directement grâce à HuggingFace. Ainsi, il ressort que notre modèle finetuné produit des résumés mieux structurés que ses concurrents : il est capable de produire des phrases mieux structurées, plus longues, tout en capturant les informations essentielles présentes dans le transcript. GPT2 et BART quant à eux, résumaient des phrases plutôt courtes et de façon peu organique. Pour vérifier cela, nous avons comparé le rouge score moyen de chacun des 3 modèles sur notre dataset de test : c'est bien le modèle LED finetuné qui obtenait de meilleures performances. C'est pourquoi nous avons décidé d'utiliser ce dernier pour générer les résumés de notre application.

4. Interface

Nous avons choisi de réaliser notre application en utilisant Streamlit. Il s'agit d'un framework facilitant la création et le partage d'application web. Notre application a été pensé de façon à être très intuitive pour l'utilisateur : celui-ci copie le lien d'une vidéo YouTube, et en un clic, la transcription puis le résumé de la vidéo est affiché.

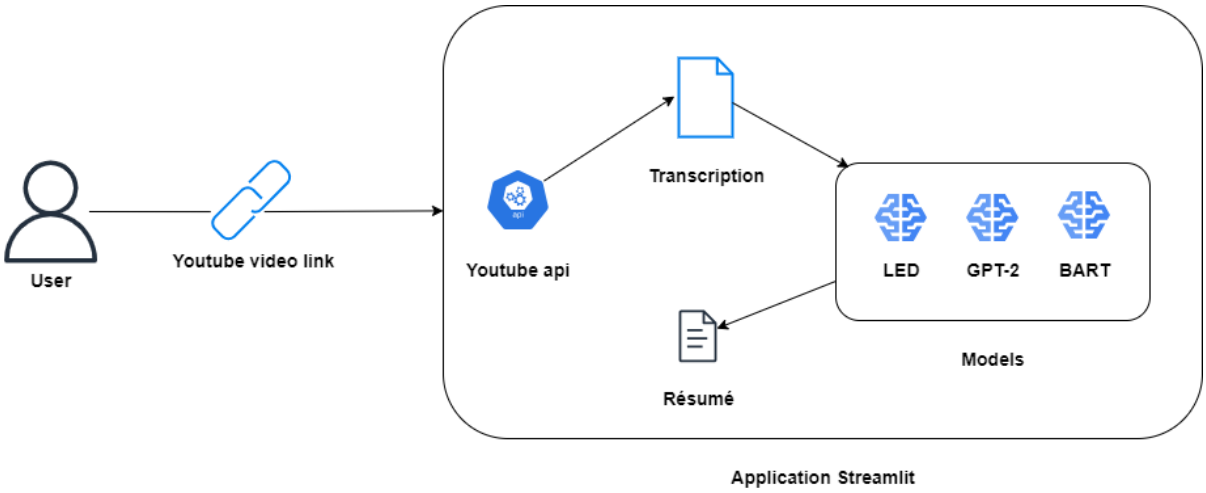

Figure 1 - Diagramme fonctionnel

On utilise l'API YoutubeTranscriptAPI pour générer la transcription de la vidéo YouTube, puis nous réutilisons cette transcription dans notre modèle LED afin de générer le résumé demandé. Dans une version antérieure de l'interface, il était également possible d'uploader son propre contenu vidéo ou audio, mais afin d'éviter à l'utilisateur de devoir télécharger la vidéo YouTube lui-même, nous avons choisi de passer par le lien de la vidéo.

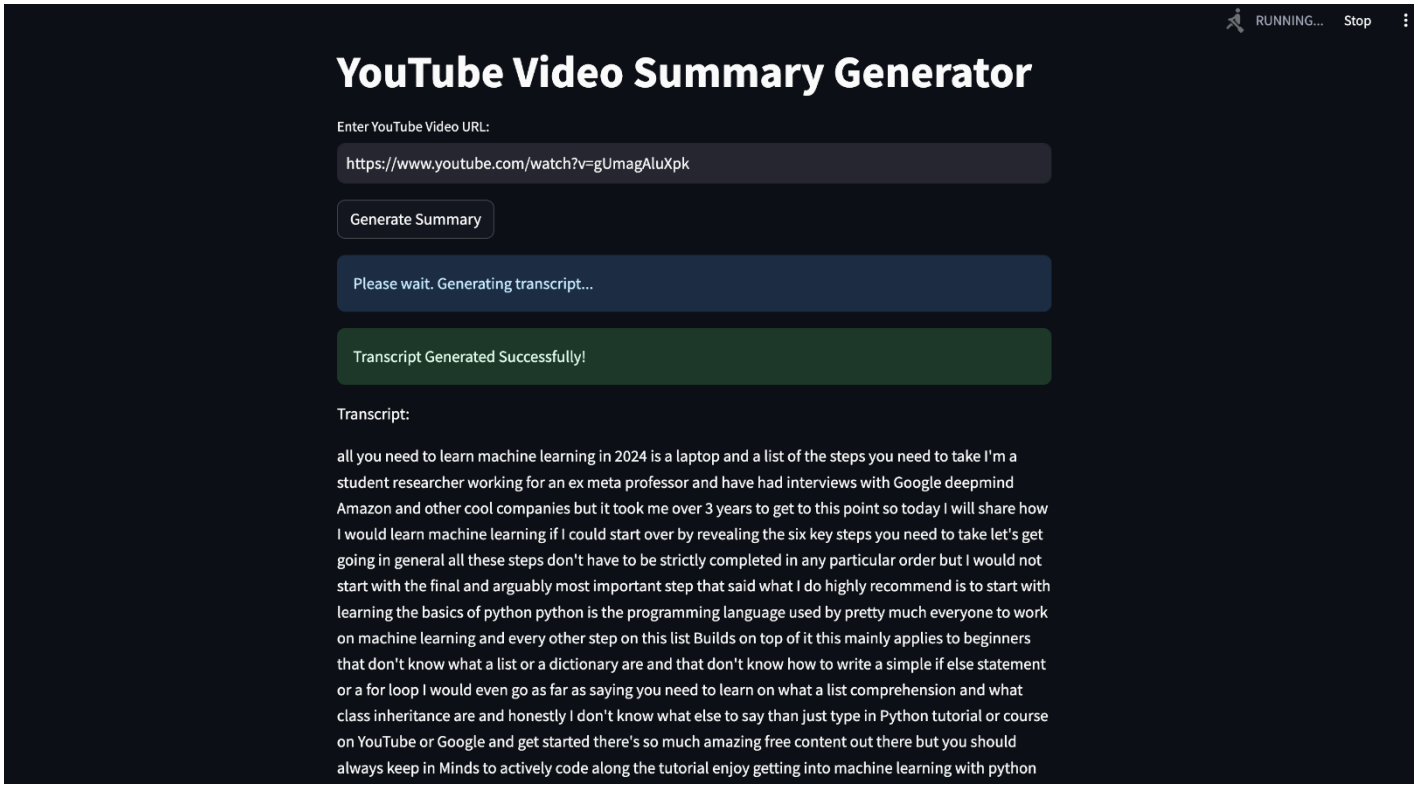

Figure 2 – Interface web Streamlit

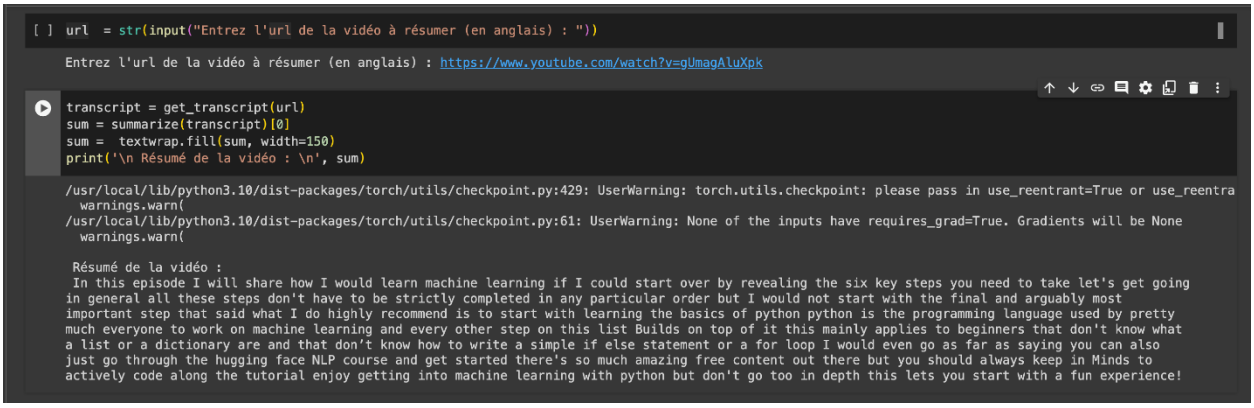

Figure 3 – Résumé d’une vidéo à partir de son URL

5. Limites et ouverture

Même si les résultats obtenus grâce à notre approche sont satisfaisants, il est important pour nous de préciser quelques limites dans notre travail.

Tout d'abord, si notre approche fonctionne correctement pour la langue anglaise, ce n'est toutefois pas le cas des autres langues. En effet, nos modèles ne sont pas multilingues, ce qui nous a poussé à nous cantonner à des datasets et des cas d'usage en anglais. Il en est de même pour la récupération des transcripts des vidéos YouTube, l'API ne permettant d'obtenir que des transcripts de vidéos en langue anglaise. Ainsi, une possible ouverture de notre travail pourrait être d'arriver à obtenir un outil performant sur plusieurs langues, notamment le français ou l'espagnol qui sont très parlées.

Notre travail a aussi été limité par les ressources dont nous disposions. Ayant choisi de travailler avec Google Colab pour une gestion plus efficace, nous avons été limités par les ressources des GPU qui y sont disponibles gratuitement. C'est pourquoi lors de l'entraînement de nos modèles, nous avons délibérément tronqué les datasets, et donc réduit les capacités d'apprentissage de notre modèle. Il pourrait être intéressant de voir quels résultats nous aurions pu obtenir avec un meilleur matériel. L'utilisation de la version gratuite de Google Colab nous a également forcé à trouver des solutions de rechange pour l'hébergement de notre application web. Cet hébergement étant limité, l'application web n'arrive pas à terme lors de la réalisation du résumé.

En outre, les datasets répondant à nos objectifs n'étaient pas très abondants. Nous avons passé une bonne partie de notre travail à en chercher, et nous n'en avons retenu qu'un au final. Notre travail a donc été limité par la disponibilité de ces données. Il en est de même pour l'obtention de transcripts, notamment audio, car l'accès aux APIs de plateformes mainstream où nous pouvions en trouver (Spotify, Deezer…) est limité et souvent payant. Tout ceci aurait pu, a priori, nous permettre d'avoir de meilleurs résultats à l'issue de notre travail.

```
Rouge-1 Recall :  0.4334394365785673
Rouge-1 Precision :  0.29603483189126745
Rouge-1 F1 Score :  0.32039494024024645
Rouge-2 Recall :  0.21695681105081777
Rouge-2 Precision :  0.14400866238137347
Rouge-2 F1 Score :  0.14944280384298564
Rouge-L Recall :  0.39650061950766186
Rouge-L Precision :  0.27280332547705866
Rouge-L F1 Score :  0.2933341539637383
```

Figure 4 – Métriques ROUGE avec dataset réduit

Bilan

Au cours de ce projet, nous avons réaliser un outil permettant de réaliser la transcription puis le résumé de contenus vidéo YouTube en anglais. Nous avons comparé plusieurs modèles et les résultats obtenus avec celui que nous avons-nous-même affiné, le Longformer Encoder-Decoder nous donnait les meilleurs résultats.

Nous avons ensuite réalisé une application intuitive pouvant être hébergé sur un site web afin de permettre à l'utilisateur d'intuitivement obtenir le résumé de la vidéo.

Plusieurs améliorations et pistes d'ouverture sont encore à creuser, notamment le support d'autres langues que l'anglais ou encore un dataset encore plus affiné et pertinent vis-à-vis de l'entraînement de notre modèle.

Bibliographie

- Longformer: The Long-Document Transformer par Iz Beltagy Matthew E. Peters Arman Cohan posté le 2 décembre 2020
- Fine-tune Longformer Encoder-Decoder (LED) for Summarization on pubmed
- Build A Text Summariser Using LLMs with Hugging Face by Gayathri Jujjuru, Published On July 21, 2023
- Streamlit documentation
- How to Launch Streamlit App from Google Colab Notebook