SAÉ 5.A.01 - Rapport sur le travail réalisé n°2

Aide sur le jeu de données et fine-tuning | 18/11/2023 - 18/12/2023

Pour cette deuxième phase, j'ai apporté mon aide aux autres membres de mon équipe concernant l'import du jeu de données et sur le fine-tuning du modèle GPT-2.

En réalisant des tests sur le jeu de données, nous nous sommes aperçus d'un problème sur le jeu de données que nous avons divisé auparavant sur Google Drive + Collab, dû à des contraintes en termes de performances. J'ai pû constater que chaque article du jeu de données étaient dupliqués six fois lors du chargement de ces derniers. En travaillant avec mon collègue Samuel Dorismond, nous avons réussi à trouver une solution au problème en nous intéressant à la fonction de chargement des données, qui utilisait une boucle d'itérations en trop. De cette manière, nous avons obtenu plus d'espace de stockage pour importer le reste du jeu de données et de le rendre disponible à travers des chunks, plusieurs fichiers divisant l'ensemble des articles. Il nous est désormais possible de charger en entièreté le jeu de données sur un notebook, mais il ne reste pas assez de place en mémoire afin de pouvoir effectuer d'autres actions, tels qu'un entraînement de modèle.

Par la suite, je me suis intéressé au fine-tuning du modèle GPT-2, en souhaitant apporter une assistance à la réalisation du livrable suivant qui consiste en cela. La problématique principale à ce sujet était de savoir comment fournir les données - dans notre cas d'étude, les textes d'articles - au modèle pour l'entraîner, à travers la fonction .fit(). En effet, la difficulté réside dans le format de notre jeu de données, car en raison des problèmes de performance et de mémoire empêchant un import direct du jeu de données depuis l'API TensorFlow, nous avions un jeu de données sous format Python personnalisé via des listes/tableaux et des dictionnaires, au lieu des types de données utilisés par TensorFlow.

En collaboration avec Quentin Vermeersch, membre de mon groupe, nous avons pu trouver à travers la documentation du modèle via la fonction Python help(gpt2_lm) qu'il était possible d'entraîner le modèle GPT-2 en fournissant directement des chaînes de caractères. Nos articles étant sous la forme de tableaux de chaînes de caractères, nous pouvons donc fournir directement les listes de mots de chaque article pour l'entraînement. Ceci étant dit, nous avons également rencontré à nouveau des problèmes liés à la saturation de la mémoire de Google Collab lors de l'entraînement avec un article, qui sont à étudier et résoudre pour la suite du fine-tuning de notre modèle. Néanmoins, nous estimons qu'en théorie, il doit être possible d'entraîner le modèle sur un premier article, puis de sauvegarder le modèle, et de réitérer le processus sur le modèle entraîné résultant et en utilisant le reste des articles suivants.

D'autre part, je me suis intéressé avec mon collègue Aurélien Zulfic à la notion de stop word (mot vide), sur laquelle nous avons mené de courtes recherches pour la réalisation du cloud word sur notre dataset des articles scientifiques. Lors de premiers essais ayant pour objectif de réaliser un graphique du nombre d'occurrences des différents mots contenus dans les articles, nous avions constaté que plusieurs mots communs appelés mots vides tels que "the", "and", ou encore "a", étaient comptabilisé dans le comptage, alors qu'ils ne sont pas utiles pour notre analyse.