Prof. Amal Zouaq

École Polytechnique de Montréal
Département Génie Informatique et Génie Logiciel
INF8460 – Traitement automatique de la langue naturelle

TP2 INF8460
Automne 2022

1. DESCRIPTION

Dans ce TP, l'idée est d'effectuer de la recherche de passages de texte dans un corpus à partir d'une question en langue naturelle. Les questions et passages sont en anglais.

Voici un exemple :

Entrée : Question : where is the show the vikings filmed?

Solution - Trouver un passage qui contient la réponse à la question : For the 2012 BBC Documentary series, see Vikings (TV documentary series) . Vikings is an Irish-Canadian historical drama television series written and created by Michael Hirst for the television channel History. Filmed in **Ireland**, it premiered on 3 March 2013 in the United States and Canada. Vikings is inspired by the sagas of Viking Ragnar Lothbrok, one of the best-known legendary Norse heroes and notorious as the scourge of England and France. The show portrays Ragnar as a former farmer who rises to fame by successful raids into England, and eventually becomes king of Denmark, with the support of his family and fellow warriors: his brother Rollo, his son Bjorn Ironside, and his wives—the shieldmaiden Lagertha and the princess Aslaug.

Ici la réponse est en gras dans le texte.

2. LIBRARIES PERMISES

- Jupyter notebook
- NLTK
- Numpy
- Pandas
- Sklearn
- Pour toute autre librairie, demandez à votre chargé de laboratoire et attendez une confirmation par écrit

3. INFRASTRUCTURE

- Vous avez accès aux GPU du local L-4818. Dans ce cas, vous devez utiliser le dossier temp (voir le tutoriel VirtualEnv.pdf)

4. DESCRIPTION DES DONNEES

Dans ce projet, vous utiliserez le jeu de données dans le répertoire *data*.

Nous vous fournissons un ensemble de données qui comprend un corpus (*corpus.csv*) qui contient tous les passages et leurs identificateurs (ID) et un jeu de données qui associe une question, un passage, et une réponse qui est directement extraite du passage.
Ce jeu de données est composé de deux sous-ensembles :

- *Train* : ensemble d'entraînement de la forme <QuestionID, QuestionText, PassageID, Réponse>. Le but est donc de construire un modèle qui retrouve les N passages les plus pertinents pour répondre à la question.
- *Test* : Un ensemble secret qui est utilisé pour évaluer votre modèle. Il est de la forme <QuestionID, Question>. Votre système doit trouver dans le corpus *corpus.csv* le ou les passages les plus pertinents.
- Notez que nous ne fournissons pas d'ensemble de validation car nous n'essayons pas d'entrainer un algorithme d'apprentissage machine dans ce TP. La distinction entre l'ensemble train / test n'est là que pour distinguer des questions pour lesquelles vous connaissez vos paragraphes / réponses et celles pour lesquelles vous ne les connaissez pas.
- Notez qu'il est possible de répondre aux requis du TP sans utiliser la réponse à la question.

5. ETAPES DU TP

A partir du notebook *inf8460_A22_TP2* qui est distribué, vous devez réaliser les étapes suivantes. (Notez que les cellules dans le squelette sont là à titre informatif - il est fort probable que vous rajoutiez des sections au fur et à mesure de votre TP).

5.1. Pré-traitement (5 points)

Les passages et questions de votre ensemble de données doivent d'abord être représentés et indexés pour ensuite pouvoir effectuer une recherche de passage pour répondre à une question. On vous demande donc d'implémenter une étape de pré-traitement des données.

Vous devez effectuer le pré-traitement du corpus (questions, passages) en convertissant le texte en minuscules, en segmentant le texte, en supprimant les mots outils et en racinisant (stemming) le texte. Votre fonction doit ensuite imprimer le nombre total de jetons dans le corpus d'entrainement (questions, passages).

5.2. Représentation de questions et de passages (35 points)

1) (*2.5. points*) *Construction du vocabulaire*. À cette étape, vous devez construire un vocabulaire à partir de votre corpus d'entrainement pré-traité. Notez que les questions et passages de l'ensemble d'entrainement doivent être inclus dans la construction du vocabulaire.

2) (*10 points*) *Création d'un modèle sac de mots pour l'ensemble d'entrainement et de test*. En utilisant sklearn et à partir de votre corpus pré-traité, vous devez représenter chaque passage et question de vos deux ensembles avec votre vocabulaire, en utilisant un modèle sac de mots des n-grammes (n=1-2) qu'ils contiennent et en pondérant ces éléments avec TF-IDF.

3) Maintenant que vous avez une représentation de vos passages et questions, il faut être capable de déterminer quel passage sera le plus pertinent pour la question posée.

a) (*5 points*) Pour cela, vous devez écrire une fonction pour évaluer la similarité cosinus entre la représentation de la question et celle de chaque passage.

b) (*5 points*) Vous devez écrire une fonction *rank_passages*, qui au moyen de la précédente, retourne pour chaque question, le top-N (N=1,5,10, 50) des passages utiles pour répondre à la question (N est un paramètre). Ces passages devront être ordonnés du plus pertinent au moins pertinent. Idéalement le passage à la position 1 sera celui qui contient la réponse à la question.

c) (2.5 *points*) Vous devez exécuter votre fonction *rank_passages* sur toutes les questions de l'ensemble d'entraînement et sauvegarder vos résultats pour l'ensemble d'entrainement selon le même format que `passage_submission.csv` sous le nom `train_passage_submission.csv`

d) (*2.5 points*) Vous devez écrire une fonction *show_top_n_passages* qui affiche le top-5 ainsi que le paragraphe attendu pour une question donnée.
e) *(2.5 points) Évaluation:* En utilisant votre *ensemble d'entrainement*, vous devez implémenter une fonction *evaluate* qui, à partir du fichier `train_passage_submission.csv` et de la comparaison entre la colonne *Gold standard passage* et vos top-N, calcule la précision top-N (N=1,5,10,50) sur l'ensemble de données en paramètre et l'afficher. Notez que nous réutiliserons la même fonction pour l'évaluation de votre ensemble de test, veillez donc à ce que la fonction soit générique et respecte le format du fichier passage_submission.
f) *(5 points*). *Discussion* : Observez les paragraphes obtenus au hasard sur un échantillon de questions. Lorsque le premier passage n'est pas celui qui est attendu (qui contient la réponse), pouvez-vous donner une raison pour sa haute similarité avec la question ? Quels sont les facteurs qui peuvent expliquer, globalement, les erreurs du modèle ? Etayez votre discussion avec des exemples précis.

5.3. Exécution sur l'ensemble de test (5 points)

Vous devez exécuter la fonction `rank_passages` sur toutes les questions de *l'ensemble de test*. A cette étape, vous devez produire un fichier `test_passage_submission.csv` qui contient pour toutes les questions de l'ensemble de test le top-N des passages retournés par votre modèle pour y répondre.
Le fichier doit respecter le format démontré dans `sample_passage_submission.csv`.

5.4. Amélioration (10 points)

Proposez une autre métrique que TF-IDF qui permette d'améliorer les performances obtenues sur l'ensemble d'entrainement. Effectuez une recherche sur Google scholar pour vous donner des idées et indiquez vos références. Démontrez l'amélioration par une exécution de la métrique et un affichage de ses performances comparé à TF-IDF sur l'ensemble d'entrainement. Enfin, vous devez générer un fichier *test_passage_submission_improved.csv* selon le même format précédent avec vos nouvelles réponses sur *l'ensemble de test* selon cette nouvelle métrique.

LIVRABLES

Vous devez remettre sur Moodle un zip contenant les fichiers suivants :

1- Le code : Vous devez compléter le squelette *inf8460_A22_TP2.ipynb* sous le nom *matricule1_matricule2_matricule3_TP1.ipynb*. Ce notebook reprend les différentes questions, et doit contenir les fonctionnalités requises avec des commentaires appropriés. Le code doit être exécutable sans erreur et accompagné des commentaires appropriés dans le notebook de manière à expliquer les différentes fonctions. Les critères de qualité tels que la lisibilité du code et des commentaires sont importants. Tout votre code et vos résultats doivent être exécutables et reproductibles ;
2- Un fichier html représentant votre notebook complètement exécuté sous format html
3- Les fichiers de soumission de données d'entrainement et de test *: train_passage_submission, test_passage_submission.csv et test_passage_submission_improved.csv* ;
4- Un document *contributions.txt* : Décrivez brièvement le pourcentage de contribution et la contribution de chaque membre de l'équipe. Tous les membres sont censés contribuer au développement. Bien que chaque membre puisse effectuer différentes tâches, vous devez vous efforcer d'obtenir une répartition égale du travail.

EVALUATION

Votre TP sera évalué selon les critères suivants :

1. Exécution correcte du code

2. Performance attendue du modèle
3. Organisation du notebook
4. Qualité du code (noms significatifs, structure, performance, gestion d'exception, etc.)
5. Commentaires clairs et informatifs
6. Réponses correctes aux questions de réflexion