Prof. Amal Zouaq

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

TP4 INF8460
Automne 2022

1. DESCRIPTION

Dans ce TP, l'idée est d'utiliser un système de traduction automatique pour générer des requêtes de base de connaissances en SPARQL à partir de questions en langage naturel.

Les bases de connaissances sont une source de données structurées, selon les standards, modèles et langages du Web sémantique, qui permettent un accès efficace à une grande quantité d'informations dans des domaines très variés. Cependant, leur accès est limité par la complexité des requêtes qui ne permet pas au public de s'en servir. Un système de traduction automatique pourrait permettre de générer automatiquement une requête étant donnée une question formulée par un usager en langage naturel.

Dans notre cas, la langue d'entrée sera l'anglais et le langage de requête sera SPARQL (https://www.w3.org/TR/sparql11-query/). Afin de faciliter le travail du modèle, nous vous fournissons une version modifiée des questions dans laquelle certains mots sont remplacés par les URIs de la base de connaissances que l'on retrouve dans la requête.

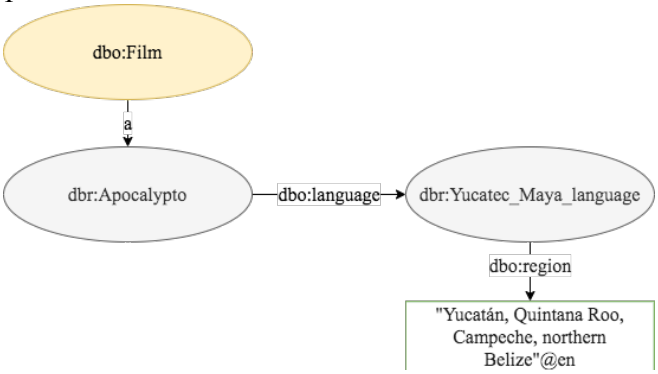
Voici un exemple :

Question originale : *In how many other states do people live, whose languages are spoken in apocalypto?*

Entrée : Question taggée : *In how many other dbp:region do people live, whose dbo:language are spoken in dbr:Apocalypto?*

Sortie attendue : Requête: *select distinct count (?uri) where { dbr:Apocalypto dbo:language ?x . ?x dbp:region ?uri }*

Cette demande renvoie à la partie suivante de la base de connaissances DBPedia:



dbr:Apocalypto a dbo:Film .
dbr:Apocalypto dbo:language dbr:Yucatec_Maya_language .
dbr:Yucatec_Maya_language dbp:region "Yucatán, Quintana Roo, Campeche, northern Belize"@en .)

Réponse : *1*

Dans le cadre de ce TP, on vous demande simplement de générer la requête SPARQL et pas de l'exécuter sur DBpedia.

2. LIBRAIRIES PERMISES

- HuggingFace
- Keras
- NLTK
- SPACY
- Numpy
- Pandas
- Pytorch
- re
- Pour toute autre librairie, demandez à votre chargé de laboratoire via le forum du cours sur Moodle

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)
- Vous pouvez aussi utiliser l'environnement Google Colab : https://colab.research.google.com/

4. ECHÉANCE

Fin de la session. La date précise sera indiquée sur Moodle.

5. KAGGLE

Le TP4-projet se fera sous forme d'une compétition Kaggle. La fin de la compétition sera indiquée sur Kaggle.
Vous devrez utiliser l'environnement Kaggle pour la soumission et l'évaluation de vos modèles.
Pour tester votre système au fur et à mesure, vous aurez le droit à 4 soumissions par jour sur Kaggle. Vous verrez deux types de résultats sur votre « private leaderboard » et votre « public leaderboard » :

- Le « public leaderboard » est calculé sur approximativement 30% des données de test, choisies aléatoirement par Kaggle. Ce score est public et est calculé sur la même tranche de donnée pour tous les participants.
- Le « private leaderboard » est calculé sur approximativement 70% des données de test et n'est visible qu'à la fin de la compétition. Le résultat final sera basé sur ce leaderboard. Si aucune soumission n'est choisie, la soumission avec le meilleur score sur le « public leaderboard » sera utilisée pour calculer le score sur le « private leaderboard ».

Pour l'évaluation, vous devrez soumettre un fichier de données *tp4_submission.csv* du même format que le fichier *sample_submission.csv* (disponible sur le site de la compétition et Moodle). *tp4_submission.csv* devra contenir pour chaque ligne de votre ensemble de test, la requête retournée par votre approche, selon le format indiqué dans la compétition.

6. DESCRIPTION DES DONNEES ET METRIQUES D'EVALUATION

Le corpus est un corpus de 5 000 paires de questions - requêtes sur DBPedia portant sur une grande variété de thèmes plus ou moins spécifiques. Trois documents sont fournis :

- Les 4000 paires de questions – requêtes d'entrainement dans un fichier *train.csv*.
- Les 500 paires de questions – requêtes de validation dans un fichier *validation.csv*.
- Les 500 questions de test pour lesquelles vous devez générer des requêtes SPARQL dans un fichier *test.csv*.

La sortie de votre modèle sera comparée à notre ensemble de référence. Vous serez évalués en utilisant la métrique « accuracy » sur les requêtes prédites par vos modèles dans la compétition Kaggle. Cette métrique vérifie pour chaque requête prédite par votre modèle si elle est exactement égale à la requête attendue.

Nous vous demandons également de rapporter, dans votre notebook uniquement, la métrique BLEU de votre modèle sur l'ensemble de validation. Cette métrique est très utilisée dans les tâches de traduction automatique et nous vous fournissons une fonction qui la calcule. Pour plus d'information voir https://en.wikipedia.org/wiki/BLEU.

7. ETAPES DU TP

A partir du notebook *inf8460_A22_TP4_squelette.ipynb* 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 cellules au fur et à mesure de votre TP).

7.1. Etat de l'art (5%)

Décrivez en trois paragraphes, dans une cellule du notebook, avec les références appropriées dans une cellule à part, les architectures de l'état de l'art pour la génération de requêtes SPARQL. Utilisez le service Google Scholar. Voici quelques mots-clés : Neural Machine Translation SPARQL, SPARQL Sequence to Sequence Model, etc.
En vous basant sur vos recherches, quelles sont les meilleures techniques de l'état de l'art ? Soyez brefs et clairs. Attention à comparer des approches sur les mêmes jeux de données et en utilisant les mêmes métriques dans votre analyse.

7.2. Méthodologie

7.2.1. Architecture proposée (5%)

Proposez une architecture d'encodeur-décodeur non pré-entrainée de type Transformer ou CNN (Convolutional Neural Networks) pour la tâche. Décrivez l'architecture et la méthodologie en quelques lignes dans une cellule de texte.

7.2.2. Mise en place (45%)

Mettez en place la solution proposée dans la partie précédente. Indiquez clairement tous vos hyperparamètres dans une cellule à part.

7.2.3. Évaluation (10%)

Affichez vos courbes de perte sur l'ensemble d'entrainement et de validation.

Affichez les résultats de votre modèle sur l'ensemble de validation au moyen de la métrique de précision globale (accuracy) et de la métrique BLEU.

Générez le fichier *tp4_submission.csv* qui contient les questions de test et leurs requêtes SPARQL.

7.3. Approche(s) avancée(s) (30%)

Reprenez les étapes de la partie précédente avec une ou plusieurs nouvelle(s) architecture(s) plus complexe(s), ou améliorant votre précédente architecture, afin d'obtenir un score plus élevé sur l'ensemble de validation et dans la compétition Kaggle. Démontrez bien cette amélioration dans votre notebook.

7.4. Conclusion (5%)

En quelques phrases précises, discutez des avantages et limites de vos architectures. Analysez les cas d'erreur. Indiquez des pistes d'amélioration futures potentielles**.**

8. LIVRABLES

- Vous devez soumettre vos résultats sur Kaggle avant la date limite de la compétition
- Vous devez remettre sur Moodle avant la date limite:

1- *Le code* : Un Jupyter notebook en Python qui contient le code tel que soumis dans l'environnement Kaggle implanté avec les librairies permises ainsi que votre fichier de soumission de données de test. Le code doit être exécutable sans erreur et accompagné des commentaires appropriés dans le notebook de manière à expliquer les différentes fonctions et étapes dans votre projet. Les critères de qualité tels que la lisibilité du code et des commentaires sont importants. Nous nous réservons le droit de demander une démonstration ou la preuve que vous avez effectué vous-mêmes les expériences décrites. *Attention, en aucun cas votre code ne doit avoir été copié de projets potentiellement existants*.

2- Un fichier *requirements.txt* doit indiquer toutes les librairies / données nécessaires.

3- Un lien *GoogleDrive* vers les modèles nécessaires pour exécuter votre notebook si approprié

4- Le fichier *tp4_submission.csv*

5- Un document *contributions.txt :* Décrivez brièvement 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. En particulier,

tous les membres du projet devraient participer à la conception du projet et participer activement à la réflexion et à l'implémentation du code.

EVALUATION
Votre TP sera évalué sur les points suivants :

Critères :
1. Références appropriées de l'état de l'art
2. Implantation correcte et efficace
3. Exécution correcte du code
4. Performance attendue des modèles / dans la compétition Kaggle
5. Qualité du code (noms significatifs, structure, performance, gestion d'exception, etc.)
6. Commentaires clairs et informatifs
7. Aspect novateur
8. Organisation du notebook
9. Réponses / Analyses correctes

CODE D'HONNEUR
Règle 1 : Le plagiat de code est bien évidemment interdit. Toute utilisation de code permis (par exemple le code exemple du transformateur) doit être référencée adéquatement. L'utilisation de code existant ne peut concerner que les architectures de base (par exemple le Transformateur). Vous ne pouvez pas soumettre un code, écrit par quelqu'un d'autre, faisant de la traduction de la langue naturelle vers SPARQL. Dans le cas contraire, cela sera considéré comme du plagiat.
Règle 2 : Vous êtes libres de discuter des idées et des détails de mise en œuvre avec d'autres équipes. Cependant, vous ne pouvez en aucun cas consulter le code d'une autre équipe INF8460, ou incorporer leur code dans votre TP.
Règle 3 : Vous ne pouvez pas partager votre code publiquement (par exemple, dans un dépôt GitHub public) tant que le cours n'est pas fini.