

## Travail pratique # 3

### Expérimentations avec des modèles *transformers* préentraînés

Automne 2022

Proposé par Luc Lamontagne

#### OBJECTIF

- Utiliser des modèles préentraînés de *transformers* pour faire de la classification de textes et choisir des mots avec un modèle de langue masqué, ou faire l'étiquetage de séquences.
- Choisir des modèles appropriés pour chaque tâche à accomplir et comprendre comment faire le *fine-tuning* (affinage) de ces modèles avec un corpus d'entraînement.
- Évaluer les performances de ces modèles sur des jeux de données.

#### Options de remise – Remettre l'une des 2 versions suivantes :

- **Version 1** : Les tâches 1 et 2 décrites dans ce document.
- **Version 2** : La tâche 3 seulement.

#### INSTRUCTIONS :

- Matériel disponible le 25 novembre 2022.
- Ce travail sera noté sur 100 et vaut 20% de la note du cours.
- Rapport et code : À remettre le 16 décembre sur MonPortail, le tout compressé en format Zip.
- Format de la remise : Soit des *notebooks* Jupyter bien documentés (html + notebook) ou un rapport PDF accompagné de code Python. L'un ou l'autre comme dans le travail pratique #2.
- Références : Chapitres 9, 10 et 11 de la 3<sup>e</sup> édition du livre de Jurafsky et Martin.
- Bibliothèques autorisées :
  - Réseaux de neurones : [HuggingFace](#) avec [PyTorch](#) – bibliothèques [transformers](#), [tokenizers](#) et [datasets](#). On suggère l'utilisation de la classe [Trainer](#) pour l'entraînement des modèles (lorsque possible).
  - Tokenisation : Les tokeniseurs correspondant aux *transformers* que vous aurez choisis.
  - Normalisation de textes : aucune normalisation (sauf la tokenisation des textes).

#### TÂCHE 1 – CLASSIFICATION DE QUESTIONS AVEC DES TRANSFORMERS

Utilisez des modèles *transformers* préentraînés pour classer correctement des questions. Réutilisez les fichiers *data/questions-train.txt* (entraînement) et *data/questions-test.txt* (test) du travail #2 pour effectuer votre tâche. Les points à explorer dans cette tâche sont :

- a) Comparez l'efficacité de 2 modèles différents de votre choix (par ex. BERT, DistillBERT, RoBERTa) après *fine-tuning* des modèles sur les données d'entraînement.
- b) (Bonus 10%<sup>1</sup>) Refaite cette fois-ci l'exercice avec 1 seul modèle pour lequel vous entraînez seulement la tête de prédiction du classificateur de texte sans aucun *fine-tuning* des blocs du *transformers*. Comparez vos résultats avec ceux de l'étape précédente. Voir la documentation de

---

<sup>1</sup> Cette partie est facultative.

*HuggingFace* pour voir comment faire ce type d'entraînement avec un *transformer* de classification de texte. Sinon entraînez votre propre classificateur en utilisant les représentations de texte encodées par le *transformer*.

## TÂCHE 2 – COMPLÉTEZ LE PROVERBE AVEC UN *TRANSFORMER*

On reprend la tâche étudiée dans les 2 premiers travaux qui consiste à compléter des proverbes. Pour cette remise, vous faites le *fine-tuning* d'un *transformer* encodeur afin de compléter des proverbes incomplets en choisissant le mot approprié.

Les consignes pour cette tâche sont :

- Vous devez résoudre le problème comme un *cloze test* avec une approche de modèle de langue masqué (*masked language model - MLM*). Autrement dit, un mot du texte est masqué et vous choisissez la meilleure option, parmi les choix du fichier de test, avec une prédiction MLM.
- Vous pouvez utiliser soit un modèle *transformer* entraîné uniquement pour le français ou un modèle multilingue. Consultez la documentation de *HuggingFace* pour les choix de modèles ayant une version MLM (*Fill-Mask*). Vous avez libre choix.
- Le fichier de proverbes est le même que celui du 2<sup>e</sup> travail. Cependant le fichier de test est différent. Si vous rencontrez des problèmes avec ce nouveau fichier de test, merci de me contacter rapidement.
- Présentez les résultats obtenus.
- Comparez les performances du modèle MLM avant et après *fine-tuning*.
- Est-ce que le modèle capture mieux le langage utilisé dans les proverbes après *fine-tuning* ?

## TÂCHE 3 – ÉTIQUETAGE DE SÉQUENCES AVEC UN *TRANSFORMER* – ANALYSE D'ADRESSES POSTALES

En vous inspirant d'un tutoriel<sup>2</sup> qui explique comment faire l'analyse d'adresses postales avec un réseau récurrent LSTM, accomplissez la même tâche avec un modèle *transformer* de votre choix. La tâche consiste à étiqueter les différentes parties d'une adresse afin de repérer le numéro civique, le nom de la rue, la ville, le code postal, etc. (voir Figure 1 tirée du tutoriel).

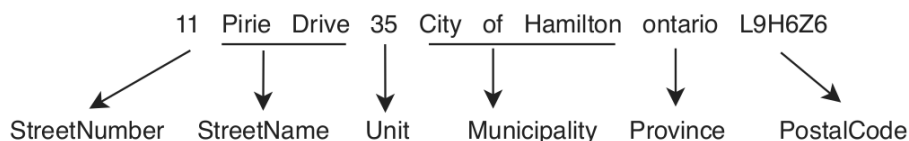


FIGURE 1 – ÉTIQUETAGE D'UNE ADRESSE POSTALE

Vous pouvez réutiliser le code du tutoriel qui vous sera utile. Cependant le problème doit être résolu avec un *transformer*. Il vous faut donc choisir un modèle préentraîné et en faire le *fine-tuning* sur les données d'entraînement.

Quelques points :

- On vous recommande d'utiliser un petit sous-ensemble des données rendues disponibles par les auteurs du tutoriel. Sinon l'entraînement du modèle avec 1 million exemples pourrait prendre plusieurs jours et causer des problèmes à l'exécution, ce qui n'est pas souhaitable en contexte

<sup>2</sup> <https://www.dotlayer.org/en/training-rnn-using-pytorch/>

académique. L'objectif est d'apprendre comment utiliser des *transformers* et non pas de reproduire ce travail intégralement.

- Vous pouvez vous limiter à l'utilisation de données d'une sous-région géographique (par ex. le Québec ou l'Ontario) si vous le souhaitez. Vous pouvez également choisir aléatoirement les exemples.
- Présentez les résultats obtenus avec votre modèle et faites une analyse de quelques erreurs commises par le modèle.

## ÉVALUATION DU TRAVAIL

### Version 1 – Classification de textes et Choix de proverbe avec MLM

|   |              |
|---|--------------|
| Tâche 1.a – Classification de questions – Choix de modèles, présentation des résultats avec <i>fine-tuning</i> , comparaison des 2 modèles, propreté du code. | 40%          |
| T2 – Compléter le proverbe – Choix de modèle, résultats en <i>fine-tuning</i> , analyse d'erreurs, propreté du code.  | 50%          |
| Qualité des <i>notebooks</i> ou du rapport  | 10%          |
| Tâche 1.b – Classification de questions avec entraînement de la tête de prédiction d'un modèle, comparaison avec les modèles de la tâche 1.a.                 | 10%<br>bonus |

### Version 2 – Étiquetage de séquences

|  |     |
|--|-----|
| Approche d'étiquetage et pertinence du choix de <i>transformer</i>   | 30% |
| Présentation des résultats. Évaluation. Analyse de quelques erreurs. | 50% |
| Propreté du code   | 10% |
| Qualité des <i>notebooks</i> ou du rapport                           | 10% |