Travail pratique # 3
Expérimentations avec des modèles *transformers*

Automne 2023

Proposé par Luc Lamontagne

OBJECTIF

- Utiliser des modèles préentraînés de type *transformers* pour obtenir, à partir de descriptions textuelles, des informations sur des incidents.
- Choisir un ou des modèles appropriés pour la tâche à accomplir et comprendre comment obtenir les informations avec ce(s) modèle(s) en utilisant un nombre limité d'exemples.
- Mener une expérimentation et évaluer la performance des modèles avec les résultats obtenus.
- Enrichir le jeu de données en annotant de nouveaux exemples (min : 10 par équipe).

INSTRUCTIONS :

- Matériel disponible le 21 novembre 2023.
- À remettre au plus tard le 16 décembre sur MonPortail, le tout compressé en format Zip.
- Ce travail sera noté sur 100 et vaut 20% de la note du cours.
- Format de la remise : Un ou des *notebooks* Jupyter bien documentés. Vous pouvez ajouter un court document en format PDF si cela vous permet de mieux expliquer votre démarche ou de présenter des résultats additionnels.
- Références suggérées :
  - Chapitres 10, 11 et 14 de la 3e édition du livre de Jurafsky et Martin portant sur les *transformers*.
  - Voir sur *HuggingFace* les sections portant sur les systèmes question-réponse et les modèles de langues.
  - Pour les techniques de *prompting*, voir le *Prompt Engineering Guide*.
- Librairies autorisées : Peu de contraintes car l'important est d'apprendre. En cas de doute, me contacter. Par exemple, des modèles HuggingFace avec *PyTorch* ou un service Web donnant accès à un modèle génératif (par ex. une version de GPT).

TÂCHE D'ANALYSE D'INCIDENTS AVEC DES *TRANSFORMERS*

La tâche à accomplir consiste à décrire différentes facettes d'un incident sur un chantier de construction en utilisant les informations contenues dans un texte. Par exemple, pour la description suivante :

> *At approximately 9:30 a.m. on November 17 2010 Employee #1 of Midwest Masonry Inc. was responsible for using the cement mixer. He was struck by the cement mixer that tipped over in the process of mixing cement/concrete. Employee #1 suffered bruises to his upper right thigh and right hip. There was no visible damage to the cement mixer. In addition there were no employees to observe how the cement mixer fell on the employee.*

on souhaiterait qu'un modèle ait la capacité de produire une analyse de l'incident avec les informations suivantes :

Événement: *He was struck by the cement mixer*

Activités: *mixing cement/concrete*

Qui: *Employee #1 of Midwest Masonry Inc.*

Où: ---

Quand: *November 17 2010*

Cause: *cement mixer that tipped over in the process of mixing cement/concrete*

De l'équipement: *cement mixer*

Des blessures: *bruises, bruises to his upper right thigh and right hip*

Des blessés: *Employee #1*

Des parties du corps: *upper right thigh and right hip*

Des décès: ---

À noter que certains champs d'information peuvent être multiples comme les activités, les blessures et les parties du corps. Vous n'avez toutefois pas à extraire plus d'une information par champs. Il est également fréquent que des informations soient manquantes (comme le lieu dans cet exemple).

Pour accomplir cette tâche, vous devez utiliser des modèles *transformers* préentraînés. Vous pouvez choisir parmi les types de modèles suivants[1] :

- Un modèle génératif (comme GPT-2) en utilisant des techniques de *prompting* pour obtenir les différentes informations pertinentes d'une description d'incident.
- Un modèle de langue (un décodeur) préentraîné pour accomplir une tâche de question-réponse ou de génération de texte.
- Un modèle encodeur-décodeur préentraîné pour accomplir une tâche de question-réponse ou de génération de texte.

Les fichiers suivants sont disponibles pour faire votre travail :

- ***incident_analysis.ipynb*** *:* un *notebook* vide pour débuter vos expérimentations. Si vous utilisez un autre fichier, assurez-vous d'utiliser le même nom. Si vous utilisez plusieurs *notebooks*, numérotez-les (par ex. *incident_analysis2.ipynb*).
- ***data/dev_examples.json :*** 100 exemples que vous pouvez utiliser pour mener vos expérimentations. Ces exemples ont été annotés manuellement pour vous donner une indication du type d'information recherché. Il est fort possible que des annotations ne correspondent à votre interprétation des incidents. Ne les modifiez pas. Vous pourrez cependant en discuter dans votre analyse si cela fausse vos résultats.
- ***data/test_examples.json :*** un fichier contenant 1000 exemples non annotés. Vous pouvez les utiliser pour vos expérimentations si cela est utile.
- ***data/new_examples.json :*** vous mettez dans ce fichier un minimum de 10 nouveaux exemples annotés que vous aurez validé manuellement pour vous assurer de leur qualité. Les nouveaux exemples sont tirés du fichier *test_examples.json* **.** Voir les instructions plus bas pour le choix des nouveaux exemples à annoter.

[1] Les extracteurs d'entités nommées (NER) ne sont pas adéquats pour accomplir cette tâche car nous avons un nombre insuffisant d'exemples pour les entraîner. De plus, il n'existe pas de modèles NER préentraînés qui permettent d'extraire ces types d'information (sauf les dates et les mentions de lieux).

- ***evaluation.ipynb*** *:* du code mis à votre disposition pour évaluer la qualité des informations extraites ou générées. Libre à vous de l'utiliser.

Quelques pistes d'exploration possibles pour cette tâche:

a) Comparez l'efficacité de différents *prompts* avec un modèle génératif (min : 2 jeux de *prompts* significativement différents).
b) Évaluation l'impact de différentes formulations de questions avec un modèle génératif de question-réponse (ce qui ressemble au point suivant). (min : 2 jeux de questions différentes)
c) Évaluer la performance de différents modèles (min : 2) pour un jeu de questions/*prompts* (par ex. un modèle extractif et un génératif).
d) Si vous vous sentez plus ambitieux ou êtes familier avec ces techniques, vous pouvez explorer de nouveaux thèmes comme le *in-context prompting* ou le *prompt tuning*. Aucune obligation.

Quelques consignes supplémentaires:

a) Chacune des informations d'une analyse peut être obtenue séparément des autres.
b) Pour les approches de question-réponse, vous devez construire vous-même les questions.
c) Les questions (ou *prompts*) doivent être générales et ne pas s'appliquer spécifiquement à un seul exemple ou une seule catégorie d'exemples (par ex. seulement les incidents d'électrocution).
d) Vous évaluez les performances des modèles avec les métriques *exact match* et *F1*. Vous pouvez utiliser le code disponible dans le fichier *evaluation.ipynb* .
e) Pour les approches génératives et/ou un modèle de langue, il est possible que les évaluations avec F1 (et encore pire avec *exact match*) ne soient pas représentatives de la qualité des résultats. Vous pouvez utiliser, si le temps pour le permet, d'autres métriques d'évaluation comme BLEU, ROUGE ou BLEURT.
f) Vous n'êtes pas tenu de faire un *fine-tuning* des modèles. Cependant, cela est une avenue intéressante pour certains modèles si le temps vous le permet.

CONSTRUCTION DE NOUVEAUX EXEMPLES

Afin d'enrichir la banque d'exemples annotés pour ce corpus, on vous demande d'annoter un minimum de 10 nouveaux exemples que vous sauvegarderez dans le fichier *new_examples.json* **.** Merci de valider les informations manuellement pour vous assurer qu'elles sont pertinentes et correspondent à votre bon jugement. Pour les cas difficiles, faites de votre mieux. À l'impossible, nul n'est tenu.

Quels exemples? Ceux du fichier *test_examples.json* qui ne sont pas annotés**.** Chacun des exemples est numéroté (voir champ *ID*) et vos 10 exemples sont déterminés par votre numéro d'équipe :

[`10 X no_équipe` : `10 X no_équipe + 9`].

Par exemple :

- Équipe 1 - les exemples 10 à 19.
- Équipe 2 - les exemples 20 à 29.
- …
- Équipe 14 - les exemples 140 à 149.
- etc.

Si vous voulez en annoter d'autres, vous pouvez utiliser ceux ayant un *ID* entre 700 à 999. Me consulter en cas de doute.

ÉVALUATION DU TRAVAIL

Choix du (des) modèle(s) et pertinence de l'approche (avec explications pertinentes)	10%
Code et expérimentations (avec explications)	40%
Présentation des résultats. Évaluation. Analyse d'erreurs.	30%
Construction de nouveaux exemples validés manuellement (min : 10 exemples)	10%
Qualité du *notebook* et des nouveaux exemples	10%