Projet 8 - Participez à une compétition Kaggle

Sources du projet : https://github.com/ben74/nlp_toxic_predictor

Notebook : https://www.kaggle.com/benjaminfontaine/xlm-roberta-toxicity-predictions

---

Table des matières

---

1) Problématique et interprétation 1

2) Modèles 3

3) Nettoyage / Preprocessing 5

4) Méthodologie / Environnement 7

5) Problèmes rencontrés 8

6) Entraînement des modèles 8

7) Leaderboard 11

8) Application 12

9) Axes d'amélioration 12

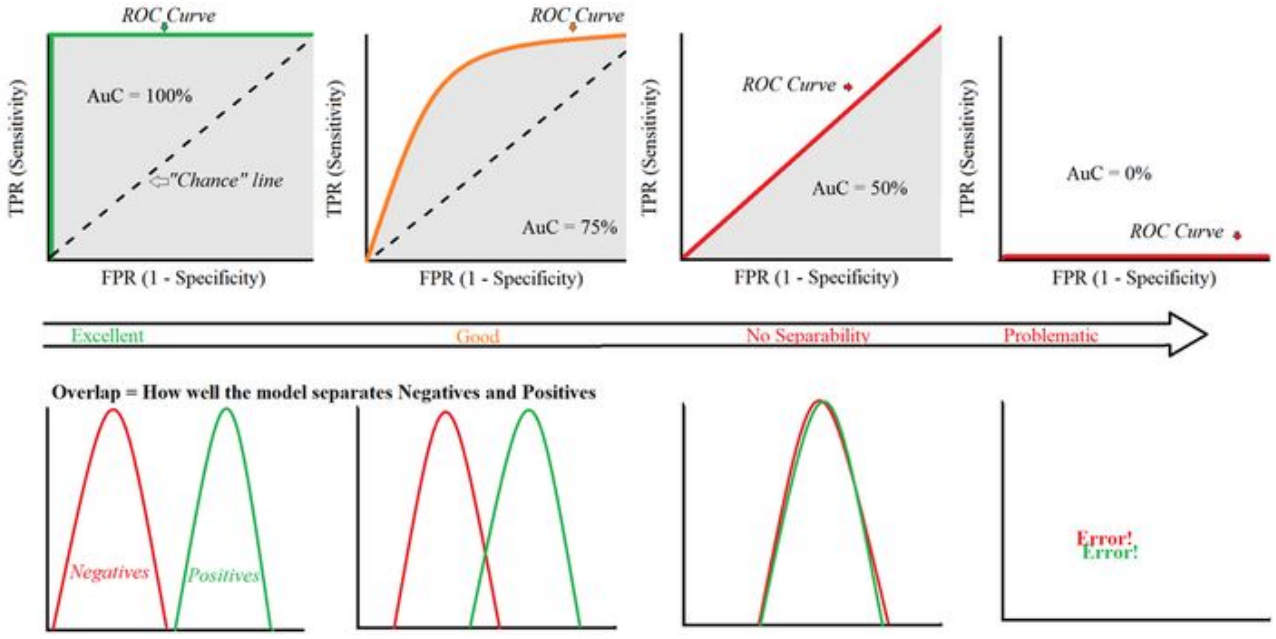
---

1) Problématique et interprétation

Pour ce projet il convient de trouver une compétition kaggle active et d'y participer. Parmi ces dernières une compétition récurrent venait récemment d'ouvrir, ayant pour but la classification de texte par un modèle nlp libre de choix

https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification

Il s'agit d'une compétition récurrente ([jigsaw-toxic-comment-classification-challenge]() et [jigsaw-unintended-bias-in-toxicity-classification]()), organisée par la société Jigsaw qui vent des prestations en api de modération des commentaires pour sites internet dans le but de débusquer si certains d'entre eux sont offensants. L'évaluation des scores finales est réalisée via la métrique ROC_AUC : receiver operating characteristics / area under curve the curve (ratio TP/FP)



https://pessoalex.wordpress.com/2019/03/11/curva-roc-explicada-em-uma-imagem/

Ces trois compétitions disposent de données associées qu'il convient d'agréger, il existe d'autres dataset en recherchant toxic ou jigswa ici : https://www.kaggle.com/search?q=toxic+in%3Adatasets, notamment des versions traduites afin d'augmenter le nb d'échantillons d'entraînement

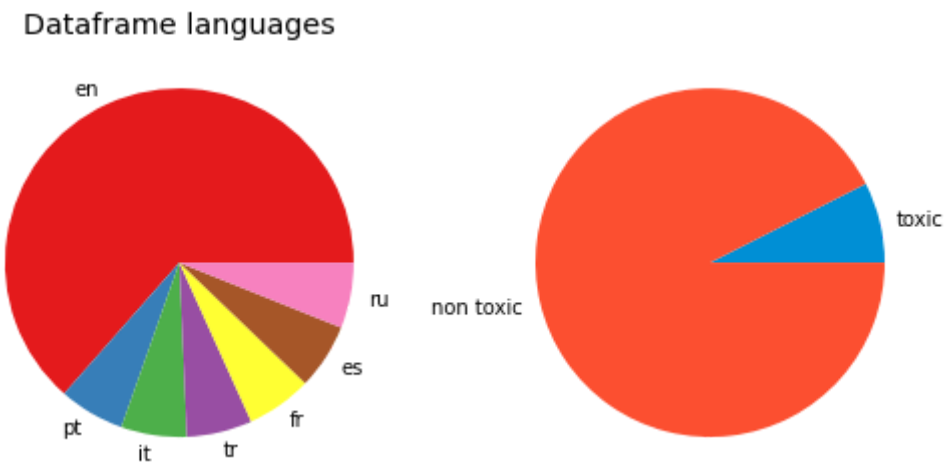
Au travers de leurs mécanisme d'attention, les récents modèles NLP depuis Bert peuvent contextualiser les mots, par exemple, dans la phrase : "Today I have eaten an apple", apple n'aura pas la même signification que "Today apple stock dropped by 10%" dans la première phrase il s'agit du fruit, dans la seconde de la compagnie

---

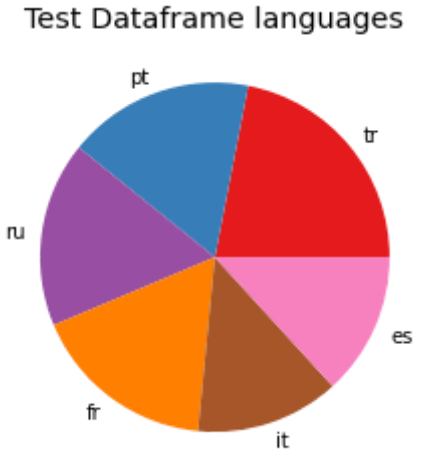
2) Modèles

Nous allons partir sur un modèle de base rapide et fiable précédemment employé dans 1 autre projet Régression Logistique à partir d'un bag of word, cet algorithme complète un score roc_auc de 0.72, mais nous avons besoin de modèles NLP (tels que distillbert, bert, xlmroberta, xlnet) afin de cerner les nuances d'un texte à l'aide du mécanisme d'attention afin d'obtenir de meilleurs résultats sur la "toxicité" d'un commentaire (s'il est grossier, offensant, incriminant ou non, selon le contexte)
Pour faire simple "I am a gay black woman filled with hate" n'a pas la même signification que "I hate gay black filled woman", un algorithme simple telle que la régression logistique ne verra pas la nuance. De même "Cette histoire est complètement débile" ne sera pas forcément offensant alors que "untel est un débile" l'est certainement plus.

Les datasets une fois fusionnés et dédupliqués présentent 3 200 751 enregistrements dont voici les proportions en langues et toxicité des commentaires



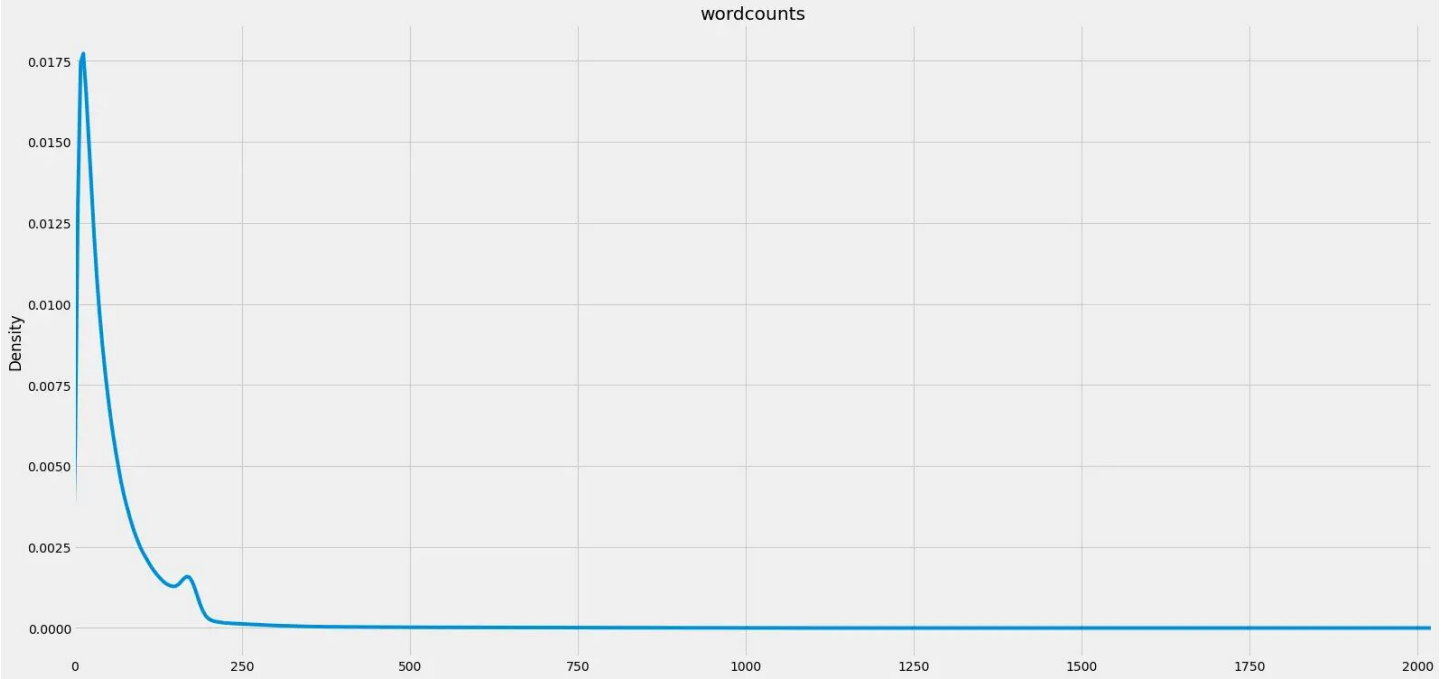
Cependant le dataframe de test pour les soumission et scoring kaggle ne comporte aucun enregistrement en anglais ..



Les textes en russe sont en alphabet phonétique latin, les regex de nettoyage de texte ne supprimeront donc pas les caractères utf8 cyrilliques

```
array(["Net eto ne moe mnenie.Ispol'zuemyi istochnik na samom dele govorit . Vot eshche odin istochnik, kotoryi govorit Pochemu eto vazhno eto ne tak, esli vy rabotaete na traditsionnom PK ili Ma
       'Da, seichas ia rabotaiu nad RFCU',
       "Prosto net dostatochnykh dokazatel'stv, chtoby peremestit' eto kuda libo eshche. Zagolovok veb saita gruppy i bol'shaia chast' ostal'noi chasti saita glasit , i eta kopiia Ink Disease s 1
       "Pozhaluista, predlozhite etu stat'iu dlia rassmotreniia. On ne iavliaetsia postoiannym avtorom stat'i, poetomu ia ne uveren, budet li on riadom, chtoby otvetit' na liubye podniatye vopros
       "Natsional'nyi arkhiv ExtravaSCANza.Vy priglasheny v Natsional'nyi arkhiv ExtravaSCANza, kotoryi prokhodit kazhdyi den' na sleduiushchei nedele s 4 po 7 ianvaria, so sredy po subbotu, v Ko
       "Ia pytalsia naiti legkuiu ssylku, no , pokhozhe, ne imeet bol'shogo znacheniia v Internete. Psikhologicheskaia otsenka v bol'shinstve sluchaev privodit k psikhologicheskomu testirovaniiu.
       "Uchityvaia, chto vy ne predostavliaete nikakogo real'nogo kontenta, no, kazhetsia, tratite vse svoe vremia na formatirovanie statei i vozvrashchenie liudei, ia polagaiu, ia ne dolzhen ozn
       "Chto znachit ia otredaktiroval ssylku ? Ia imel polnoe pravo udalit' neissledovannuiu neproverennuiu informatsiiu o zhivom cheloveke. Vy ne dolzhny byli povtorno dobavliat' eto voobshche.
       'Atren, eto tvoi blog?',
       "Nu, eto nikogda ne govorilo, chto dvizhenie bylo zaversheno. I snova, ob'iavlenie s ofitsial'nogo saita kluba ne iavliaetsia pervoistochnikom, esli vy eto imeete v vidu"],
```

Très peu de posts avec plus de 250 mots ce qui conforte la limitation en entrée de 192 mots



Le fichier "unintended-bias-train" possède des probabilités "toxic" au lieu d'une classification vrai / faux, ces dernières seront arrondies pour l'entraînement du modèle, ce qui donnera de meilleures prédictions au final

	index	comment_text	toxic	lang	f
2537698	1223341	'When you think about it, what clothing isn't ...	0.0	en	jigsaw-unintended-bias-train.csv
366018	195402	Note . He, merci pour la note de l'article Ila...	0.0	fr	jigsaw-toxic-comment-train-google-fr-cleaned.csv
2159793	837567	Read what is actually going on . you wont have...	0.0	en	jigsaw-unintended-bias-train.csv
958898	206513	Editar resumo por CTF83! Alt editar . Eu chego...	0.0	pt	jigsaw-toxic-comment-train-google-pt-cleaned.csv
669783	99919	Vy ne mozhete izmenit' kartu seichas. On byl z...	0.0	ru	jigsaw-toxic-comment-train-google-ru-cleaned.csv
...	...	...	...	...	...
2818992	1511259	That's the way to do it Unifor, stop productio...	0.0	en	jigsaw-unintended-bias-train.csv
2291715	972267	The PQ gov't might start an inquiry, but the f...	0.0	en	jigsaw-unintended-bias-train.csv
434311	55865	Kapattiginiz bir SPI hakkinda daha fazla incel...	0.0	tr	jigsaw-toxic-comment-train-google-tr-cleaned.csv
2200745	879383	Threat to mankind.LOL!!!.Little OTT there friend	0.2	en	jigsaw-unintended-bias-train.csv
2443657	1127393	Any God who thinks killing ten kids on a bet o...	0.3	en	jigsaw-unintended-bias-train.csv

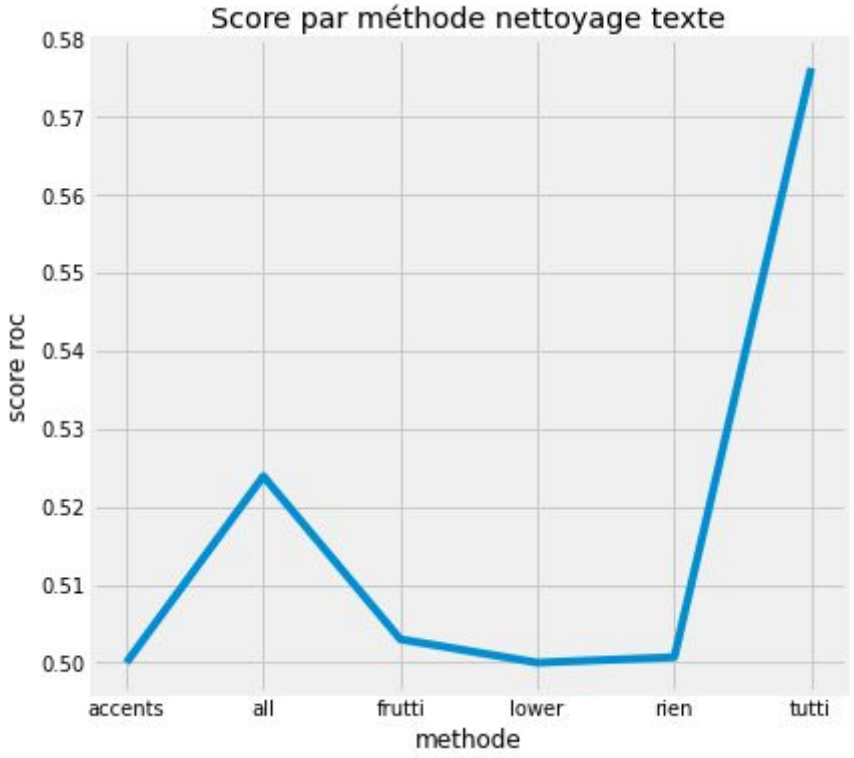
3) Nettoyage / Preprocessing

Une fonction de nettoyage et de tokenisation de ces données a été conçue en amont, afin que nous n'ayons pas à ré-encoder ces valeurs à chaque étape d'entraînement du modèle, ce qui nous permettra de gagner un temps précieux, le kernel final étant limité à 3 heures d'exécution pour valider le score final, une vérification de ce chronomètre permettra de sortir

des boucles d'entraînement afin d'exporter les prédictions peu avant cette échéance.

Afin de tester l'incidence de cette fonction, selon ses paramètres sur les performances, nous avons découpé 30000 enregistrements aléatoire du jeu d'entraînement afin d'entraîner séquentiellement le même modèle avec différents paramètres, testé sur 10000 enregistrements aléatoires, afin d'établir la meilleure méthode de preprocessing de texte.

Chacun de ses paramètres est un chiffre premier, ce qui permet de rapidement cumuler ces filtres, les retrouvant à l'aide de l'opérateur modulo % mod == 0

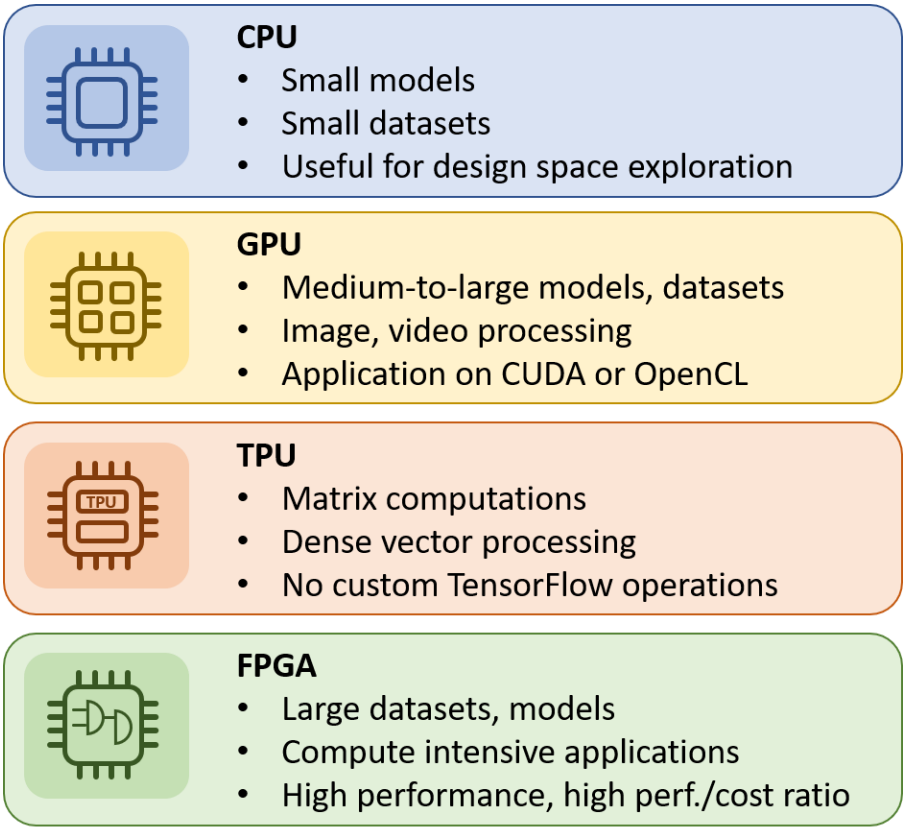


La méthode nommée "tutti" / modulo 17 présente les meilleurs résultats, après l'application de tous les filtres créés au sein de la fonction "TraitementTexte".

4) Méthodologie / Environnement

Bien que l'environnement colab pro ait des TPUs moins performants que ceux de Kaggle, leurs notebooks peuvent être exécutés bien plus longtemps. Une stratégie de sauvegarde des résultats, des logs et des poids a été mise en place afin de pouvoir "résumer" l'entraînement de ce modèle autant de fois que nous le souhaitons.

Comparatif des types de processeurs disponibles pour le machine learning, l'accès au FPGA (Field Programmable Gate Arrays : Ensemble de portes logiques programmables afin d'être le replica physique d'un modèle) disponible sur microsoft Azure pour ResNet50, VGG16 par exemple .. Tout comme certains processeurs (ambarella h2) sont prévus pour l'encodage vidéo.



https://inaccel.com/cpu-gpu-fpga-or-tpu-which-one-to-choose-for-my-machine-learning-training/

De plus une stratégie de monitoring a été appliquée en lançant en thread de fond, au début du notebook, une requête récurrente visant à envoyer des paquets "heartbeat" vers un serveur cible, afin d'obtenir des notifications à l'arrêt de l'exécution / fin d'un traitement, ou déconnexion de l'environnement. Ces notifications sont récupérées sur mon environnement local, et passées dans une synthèse vocale afin d'obtenir ces notifications sans avoir à consulter ses mails toutes les 5 minutes.

5) Problèmes rencontrés

Les environements Kaggle se retrouvaient fréquemment "out of memory" lors de l'entraînement des modèles. Ce problème a été résolu en observant la version employée sur notebooks colab qui ne présentaient pas ce défaut : tensorflow 2.2.

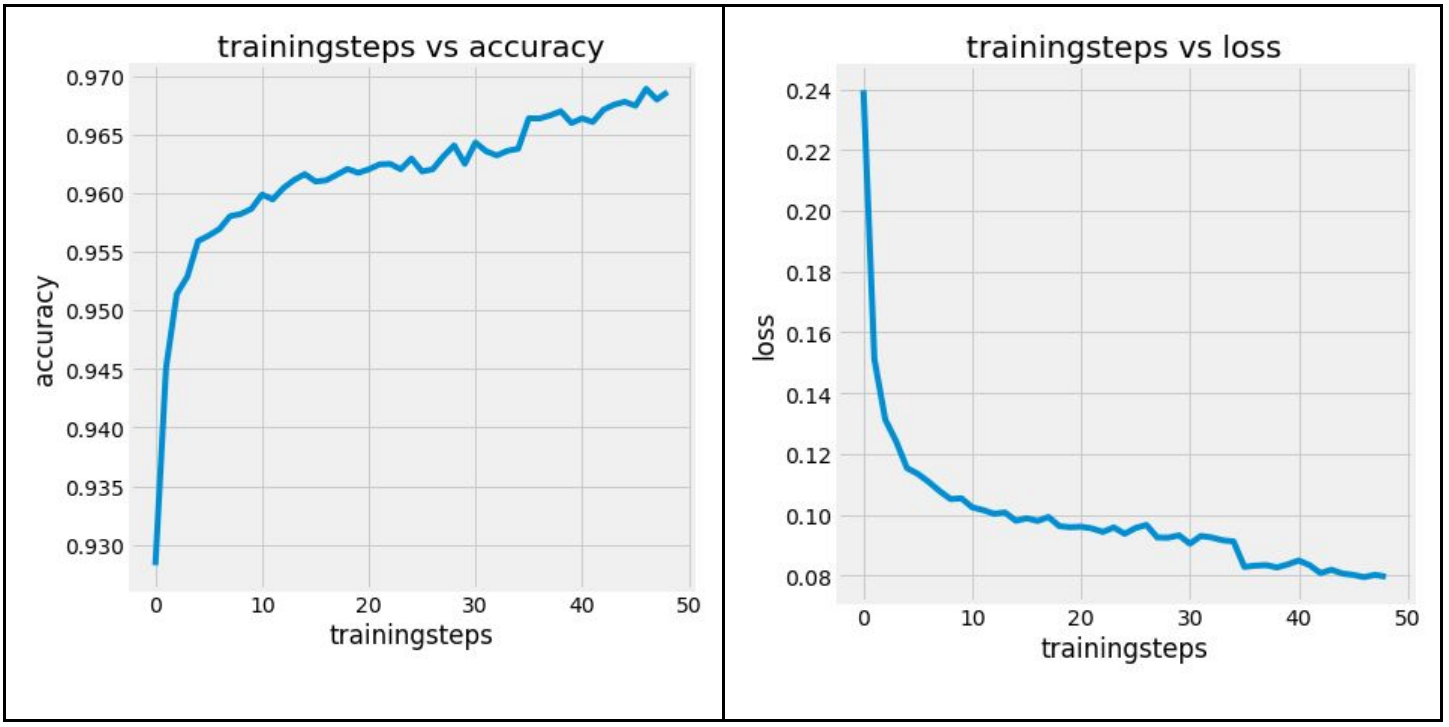
Les environnements Kaggle ont tendance à avoir le switch internet basculé à offline ..
Cela peut anéantir des heures d'exécution de code au moment d'envoyer une notification de monitoring avant d'exporter les résultats par exemple ..

A l'instar d'un serveur web, lors du test des variation de preprocessing des variables, ainsi que le test de différents modèle j'ai pris conscience qu'il fallait interrompre parfois les entraînements à la hâte afin de pouvoir tester d'autres idées. Je me suis heurté au problème que la mémoire de la carte graphique ou du tpu n'était pas vidée et saturait au prochain essai .. J'ai donc mis en place une simple interrogation d'une url via requests qui retourne si je souhaite interrompre l'entraînement du modèle en cours, sauvegardant son historique ses poids etc .. ce qui correspond à peu près à un "graceful restart"

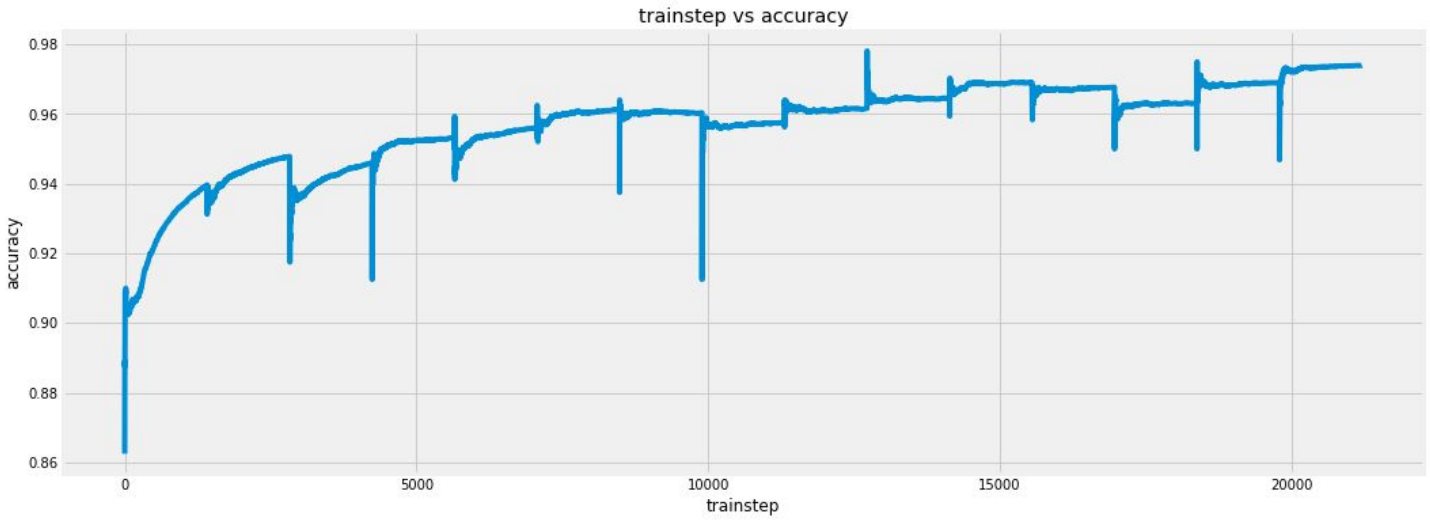
La mise en place de l'entraînement des modèles (distillbert, bert, xlmroberta) sur TPU aboutissait souvent à des erreurs de débordement de l'utilisation mémoire de ce dernier, afin d'y pallier, il suffit de ré-initialiser l'environnement (relancer un container docker) afin de s'assurer que le TPU soit déconnecté et vidange son usage HBM, et de réduire le batch size de manière proportionnelle à cet excès (notamment concernant xlmroberta_large)

6) Entraînement des modèles

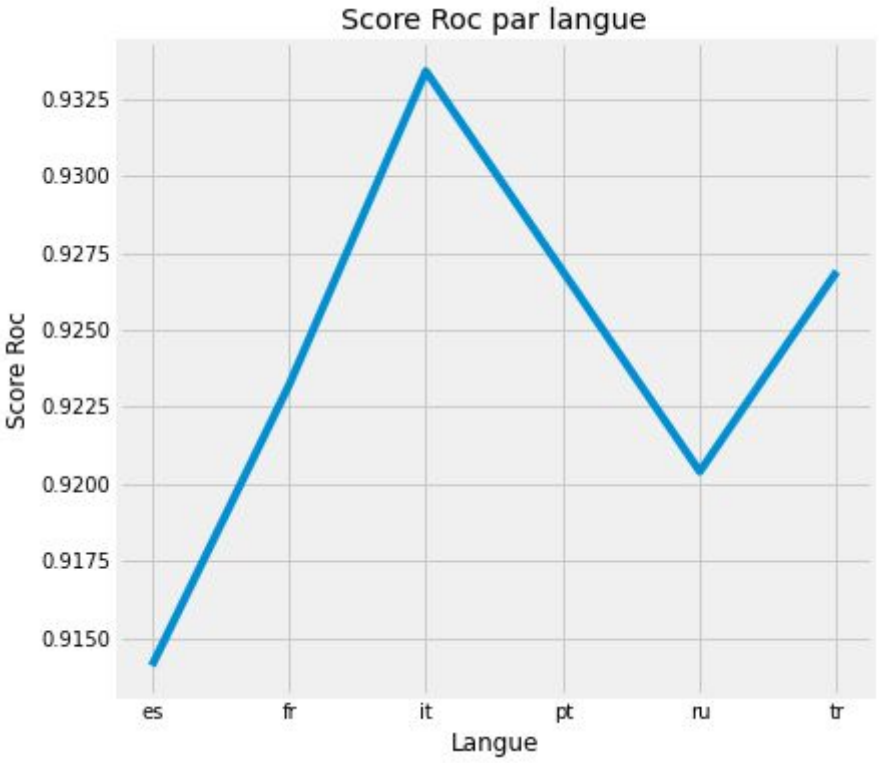
On constate rapidement que la précision de ce dernier atteint un palier de 0.92 dès la première séquence d'entraînement (sur 106720 enregistrements avec un batch size de 80, l'ensemble des données étant réparties sur 30 splits afin de ne pas saturer la mémoire de l'environnement / TPU sur colab).



On constate que le modèle parvient à une précision de 0.91 très rapidement, puis l'obtention de meilleures performances devient de plus en plus coûteux en terme d'entraînement de ce dernier.



Les scores par langue semble corroborer le fait que l'italien est une langue simple (le florentin, notamment concernant l'orthographe vis à vis d'autres langues latines)



Le notebook kaggle (adapté pour la limitation d'éxecution de 3 heures) dispose des logs d'une execution complète sur sa version 20 (une fois les problèmes relatifs à la mémoire et la version de tensorflow réglés) :
https://www.kaggle.com/benjaminfontaine/xlm-roberta-toxicity-predictions?scriptVersionId=34369485

7) Leaderboard

Search

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions Submit Predictions

673	Ivan Z		0.9199	4	3d
674	Anthony Chan		0.9198	2	2d
675	duthchao		0.9197	2	19d
676	data warriors		0.9196	4	2mo
677	Rajnish Singh		0.9192	23	17h
678	Gary		0.9190	11	2mo
679	Benjamin Fontaine	</> Averaged submissi...	0.9183	46	21h

Your Best Entry

Your submission scored 0.9183, which is an improvement of your previous score of 0.9163. Great job!

Ces travaux m'ont permis de me "hisser" à la 679 place ce classement, le meilleur score roc_auc obtenu au 20 mai 2020 est de 0.9546

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	Lingua Franca			0.9546	283	10h
2	vecxoz			0.9497	261	1h
3	qin & zhang & ma			0.9491	211	9h

8) Application

Une modeste application a été conçue dans le cadre de ce projet : elle permet de constater que le jeu d'entrainement manque d'exemple positifs pour les mots fréquemment utilisés dans des insultes :

```
git clone https://github.com/ben74/nlp_toxic_predictor.git && cd nlp_toxic_predictor && py toxic.py;# Cloner et executer l'application

curl -k 127.0.0.1:8082 -F "text=Je promène mon chien au bord de la seine";#0
curl -k 127.0.0.1:8082 -F "text=Je promène ma chienne au bord de la seine";#1
curl -k 127.0.0.1:8082 -F "text=It's okay to be gay";#1
```

L'utilisation de féminins de nombreux mots conduisent à une classification "toxic", il est cependant mieux d'avoir de Faux Positifs que de vrais Négatifs

9) Axes d'amélioration

- Tester d'autres méthodes de nettoyage de texte : conversion de acronymes et contractions en texte complet (asap => as soon as possible, wouldn't => would not), ceci séquentiellement sur les données de validation afin d'étudier plus en détail leur influence
- Utiliser sentencepiece au lieu du tokenizer par défaut
- Utiliser des modèles plus spécialisés (Camembert, GermanBert, Umberto) vis à vis de chaque langue du dataset afin d'obtenir de meilleures prédictions
- Parvenir à implémenter torchxla afin de lancer des modèles transformers sur TPU via pytorch
- Geler certains layers des modèles afin d'obtenir une amélioration des performances

- Rajouter une couche dense (32 unités) avant la couche finale de classification afin d'affiner les prédictions (oom)
- Entrainer le modèle uniquement sur les textes les plus similaires à l'ensemble du jeu de test / soumission pourrait parvenir à un modèle plus spécialisé atteignant un score de classification supérieur par tâche donnée.