Camembert

1. Introduction

Camembert est une version francophone de modèle BERT, pré-entraîné sur 138GB de texte français. Rendu publiquement par les équipes de facebook AI Research associés aux chercheurs de le L'INRIA.

Le fait qu'il est entraîné sur un grand corpus de texte, il se révèle le choix le plus adapté pour résoudre les problématiques liés à la classification de texte, traduction de texte. Cambebert est testé et évalué sur différentes tâches pour traitement de texte français, la reconnaissance d'entités nommées (NER) ,on trouve aussi l'inférence en langage naturel NLI.

2. Architecture Camembert

Camembert comme roberta et bert sont des transformateurs multicouche bidirectionnel. Camembert utilise l'architecture de base bert, similaire à roberta, la différence principale étant l'utilisation du masquage des mots entiers(whole-words masking) et l'utilisation de la tokenisation SentencePiece.

Masquage des mots entiers: consiste à masquer tous les sous-mots correspondant à un mot en une seule fois dans une séquence(token, phrase, paragraphe), Ceci est fait parce qu'ils veulent pré-entraîner un modèle bidirectionnel. La plupart du temps, le réseau verra une phrase avec un jeton [MASK], et il sera entraîné à prédire le mot qui est censé être là.

SentencePiece [2] est un analyse lexicale (tokenizer et detokenizer) de texte non supervisé principalement pour les systèmes de génération de texte basés sur un réseau neuronal (*ANN*) où la taille du vocabulaire est prédéterminée avant l'entraînement du modèle neuronal.

-Il entraîne des modèles à partir des modèles de tokénisation et detokénisation à partir de phrase.Il traite ces phrase comme des séquences de caractère unicode.

-Rapide et léger : la vitesse de segmentation est d'environ 50 000 phrases/secondes et l'empreinte mémoire d'environ 6 Mo

J'ai pas présenté l'architecture car il est pas présentée dans l'article de recherche Camembert il nous redirige vers l'architecture Roberta, déjà présenté dans l'état de l'art de Roberta. Les seules différences sont **Masquage des mots entiers, SentencePiece.**

3. Modèle de detection NLI pré-entrainé

Pour une version en français de notre API nous avons utilisé un modèle pré-entraîné issu de HuginFace[1] entraîné sur les données XNLI qui est un jeu de données composé de

premise et hypothesis écrites et parlées, ce corpus est composé de 14 langues parmi elle le français, d'ailleurs c'est le seul jeu de données NLI conséquent existant en langue française.

Pourquoi le choix de d'un modèle pré-entrainé:

à cause de manque ressource locale et de performance GPU, nous étions dans l'impossibilité de reproduire l'entraînement de modèle camembert sur le jeu de données XNLI. Toutefois pour s'assurer des performances de classification enregistré dans ce dernier, nous l'avons testé sur le jeu de données contradictory.

4. Test de modèle pré-entrainé:

a) Résultat de validation,test de modèle pré-entraîné:

Set	Accuracy
validation	81.4
test	81.7

Figure1:Accuracy de data test,validation de modèle pré-entrainé

b) Evaluation des résultat de données contradictory test:

Pour évaluer le modèle pré-entraîné, nous avons utilisé les données d'entraînement, de fait que nous avons pas suffisamment de données de test, et comme ces données étaient pas utilisée pour entraîner le modèle camberbet entrainer par XNLI français, la quantité des données de test sont les suivantes:

label	premise	hypothesis
0	133	133
1	129	129
2	128	128

figure2:Données test

Résultat:

set	accuracy
test	83.333333

figure3: accuracy de données test contadictory

Nous constatons que le modèle obtient une précision de classification de 83,33%, et un taux 16,77% de faux positifs et faux négatifs, qui sont de mauvaises classification.

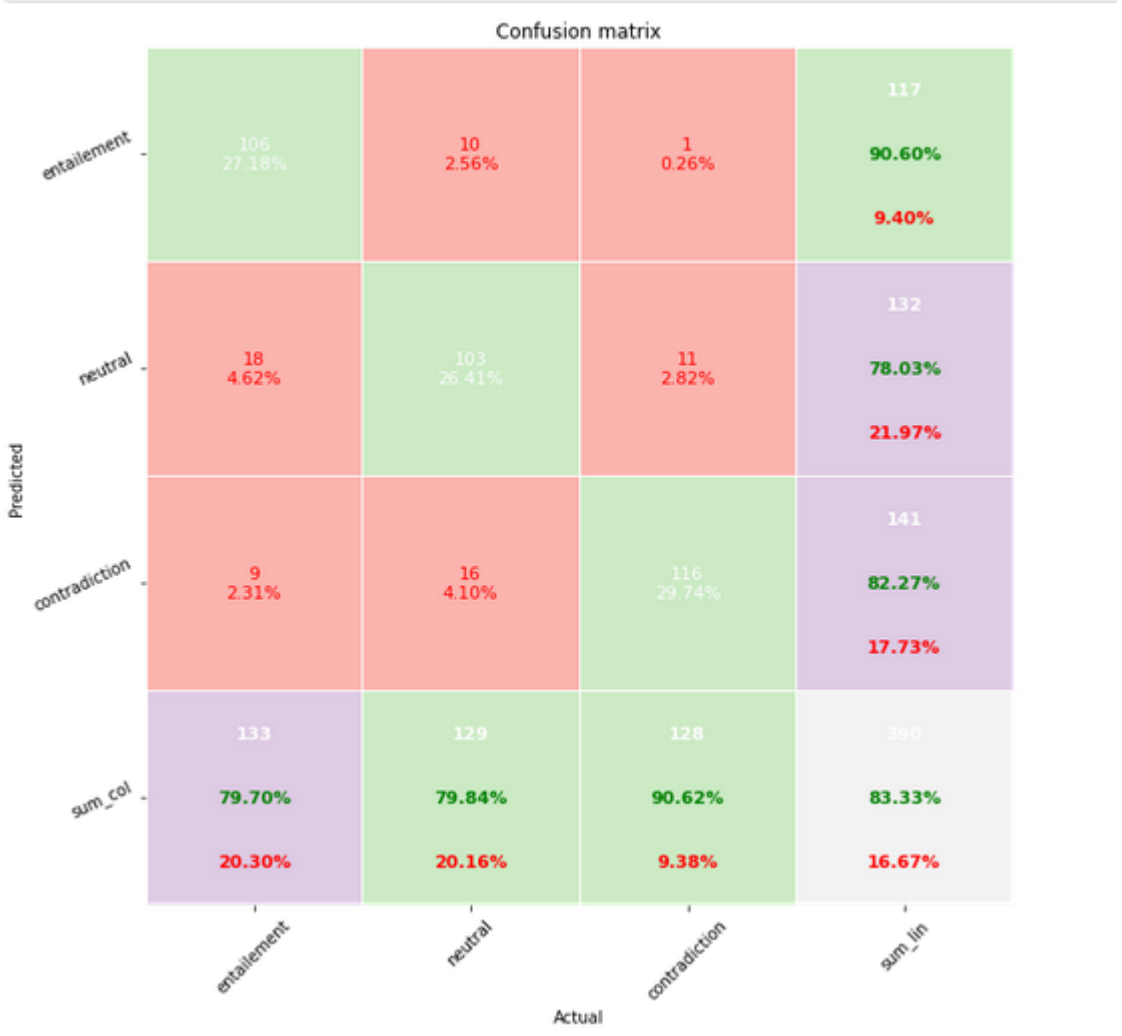

figure4: matrice de confusion des résultats sur data test

Nous remarquons que le predit bien: -106/133 entailement
-103/129 neutral
-116/128 contradiction

resources:
[1]: https://huggingface.co/BaptisteDoyen/camembert-base-xnli
[2]: https://www.gladir.com/CODER/SENTENCEPIECE/intro.htm