

Reunion 10/10/2024

# To do list

- Ecrire abstrait pour LIFT2024
- Implementer classification binaire

# Classification binaire

- A cause des resultats pas satisfaisants avec la classification de pauses en plusieurs categories, on avait discuté de la possibilité de faire une classification binaire:
  - 0 - pas de pause significative
  - 1 - pause significative
- J'ai gardé les textes reconstruits avec les pauses en categories:
- *"L'intention de l **cat\_5** 'aéroport de biard **cat\_4** diminuer la poussé des gaz **cat\_1** sur le décollage de ses avions."*
- Mais j'ai remplacé les pauses avec de 1 et le manque de pause avec 0:
- *"L **0** 'intention **0** de **0** l **1** 'aéroport **0** de **0** biard **1** diminuer **0** la **0** poussé **0** des **0** gaz **1** sur **0** le **0** décollage **0** de **0** ses **0** avions."*

# A premiere vue, des bons resultats

Evaluation Accuracy:

{'accuracy':

0.8310657596371882}

Tous les predictions sont “no pause”...

```
Token: '_Cette', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_initiative', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_de', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_retour', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_en', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_arriere', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_dans', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_la', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_medecin', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_e', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_est', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_appuyé', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_par', True Label: 'pause', Predicted Label: 'no pause'
Token: '_des', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_expérience', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_démon', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_tra', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_nt', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_la', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_puissance', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_de', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_l', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_t', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_effet', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_place', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_bo', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_sur', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_notre', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_corps', True Label: 'pause', Predicted Label: 'no pause'
Token: '._', True Label: 'pause', Predicted Label: 'no pause'
Token: '_L', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_t', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_effet', True Label: 'no pause', Predicted Label: 'no pause'
Token: '_place', True Label: 'no pause', Predicted Label: 'no pause'
```

Donc j'ai calculé l'accuracy par categorie...

```
Detailed Evaluation Predictions:  
Accuracy for 'pause': 0.00  
Accuracy for 'no pause': 1.00
```

Possible cause: les classes ne sont pas équilibrées - beaucoup plus de “no pause” que de pause, donc le modele va forcément predire plutot no pause.

Pause count: 451, No pause count: 2748

# Une possible solution

Ajouter des poids a la fonction loss pour que les predictions de pause soient penalisees plus strictement, donc la classe “pause” est plus importante

```
Detailed Evaluation Predictions:  
Accuracy for 'pause': 0.57  
Accuracy for 'no pause': 0.68
```

# Une autre possible solution: oversample pause / undersample no pause

On modifie nos donnees pour avoir une distribution plus équilibrée des classes avant d'entraîner le modèle.

Oversampling = duplication des exemples de la classe minoritaire - donc les tokens suivis par pause

Undersampling = supprimer des exemples de la classe majoritaire

Est-ce une bonne stratégie dans notre cas?

# Essai...

J'ai essayé d'écrire une fonction pour équilibrer le dataset (Pause count: 921, No pause count: 5554), qui a résulté dans une petite amélioration de l'accuracy

```
Detailed Evaluation Predictions:  
Accuracy for 'pause': 0.65  
Accuracy for 'no pause': 0.68
```



# Ce qui nous reste a faire

1. Hyperparameter tuning
  - a. Changer le nombre d'epoch
  - b. Ajuster la learning rate
2. Calibrer la fonction de weighted loss
3. Utiliser un model different - bert, roberta
4. ? Ensemble methods - on entraine plusieurs modeles et on combine leurs predictions (majority voting, averaging probabilities)
5. Analyser les echecs -> voir s'il y a des patterns la ou il y a des erreurs dans la prediction

# 1. Hyperparameter tuning

10 epochs, learning\_rate =  $2e-5$

Detailed Evaluation Predictions:

Accuracy for 'pause': 0.63

Accuracy for 'no pause': 0.78

# 1. Hyperparameter tuning

15 epochs, learning\_rate = 2e-5:

Detailed Evaluation Predictions:

Accuracy for 'pause': 0.69

Accuracy for 'no pause': 0.83

# 1. Hyperparameter tuning

15 epochs, learning\_rate = 1e-5:

Detailed Evaluation Predictions:

Accuracy for 'pause': 0.70

Accuracy for 'no pause': 0.70

# 1. Hyperparameter tuning

20 epochs, learning\_rate = 1e-5:

Detailed Evaluation Predictions:

Accuracy for 'pause': 0.72

Accuracy for 'no pause': 0.73

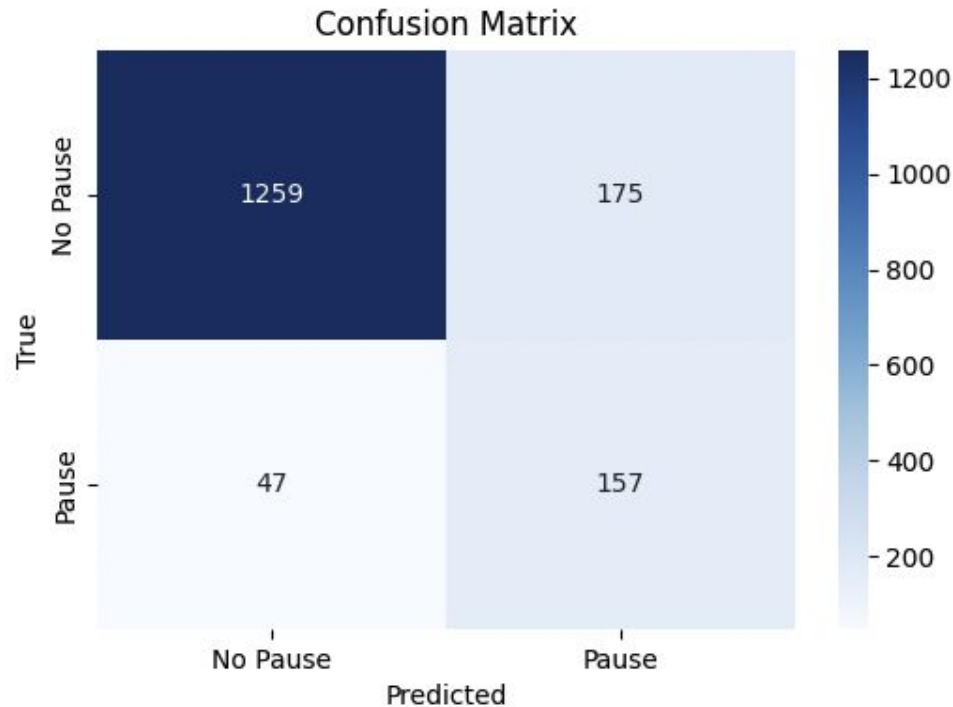
# 1. Hyperparameter tuning

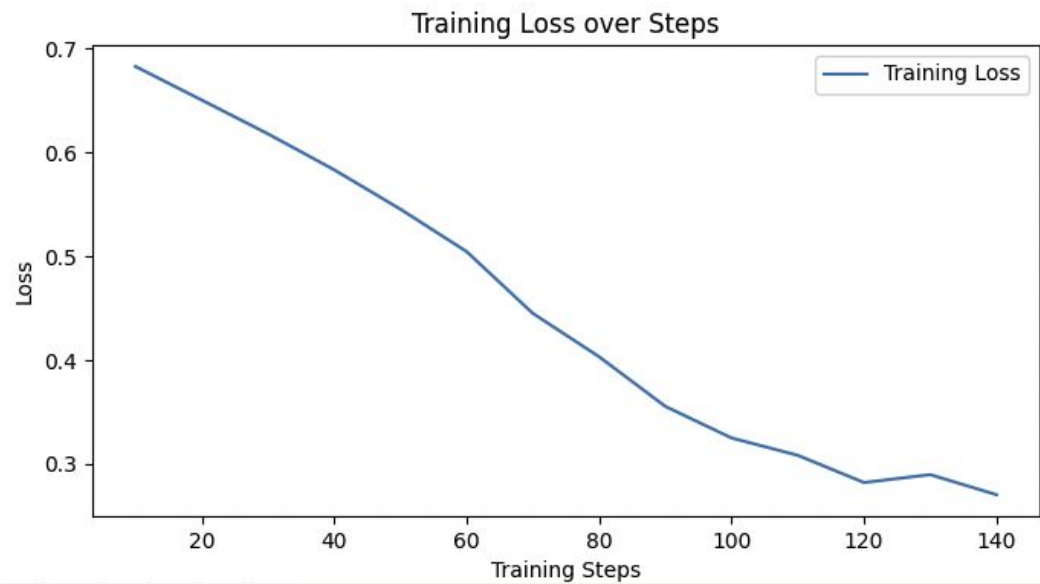
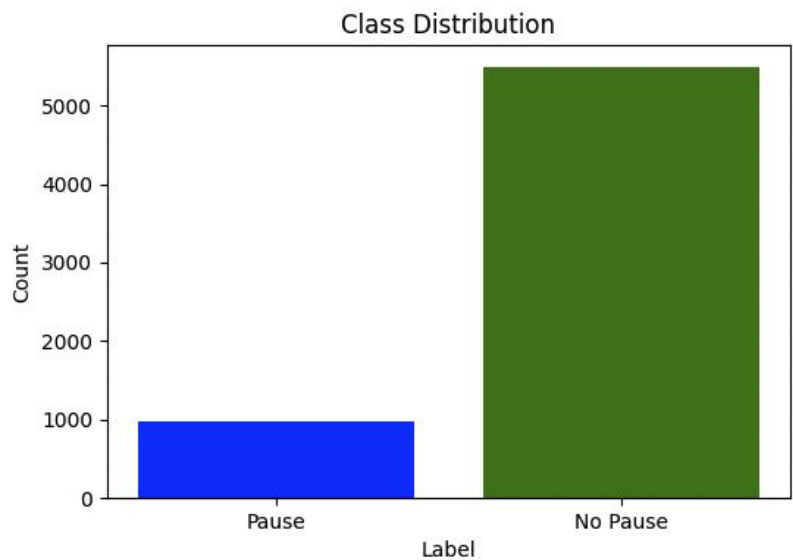
20 epochs, learning\_rate = 2e-5:

Detailed Evaluation Predictions:

Accuracy for 'pause': 0.82

Accuracy for 'no pause': 0.82





\* Memes hyperparametres, mais sans augmenter les donnees

Detailed Evaluation Predictions:

Accuracy for 'pause': 0.53

Accuracy for 'no pause': 0.69