Due Date: March 27th 23:00, 2023

Instructions

- *Ce devoir est difficile – veuillez le commencer en avance.*
- *Pour toutes les questions, montrez votre travail!*
- *Soumettez votre rapport (PDF) et votre code par voie électronique via la page Gradescope du cours. Votre rapport doit contenir les réponses au problème 2 (toutes les questions) et au problème 3 (toutes les questions).*
- *Pour les expériences ouvertes (c'est-à-dire les expériences qui n'ont pas de cas de test associés),* ***sauf indication contraire****, vous n'avez pas besoin de soumettre le code – un rapport suffira.*
- *Vous ne pouvez pas utiliser ChatGPT pour ce devoir. Vous êtes encouragé à poser des questions sur la Piazza.*
- *Les TAs pour ce devoir sont* ***Muawiz Sajjad Chaudhary*** *(IFT6135B) et* ***Ghait Boukachab*** *(IFT6135A).*

Résumé: Dans ce travail, vous allez implémenter un modèle de langage séquentiel (un **GRU**) avec une attention croisée et un modèle de self attention (un **Transformer**). Vous étudierez également les limites et les biais des grands modèles de langage (LLM).

Dans le problème 1, vous allez utiliser les modules PyTorch intégrés pour implémenter une variété d'architectures RNN (GRU, encodeur GRU bidirectionnel, encodeur-décodeur GRU avec soft attention) ainsi que l'implémentation de l'architecture Transformer via ses différents blocs de construction.

Dans le problème 2, vous allez utiliser les modules que vous avez implémentés et effectuer une analyse de sentiments sur le jeu de données Amazon Polarity. Vous comparerez la performance des RNNs à différentes configurations du Transformer.

Dans le problème 3, vous allez utiliser HuggingFace pour analyser les limites des LLM et les différentes sources de biais.

L'ensemble de données Amazon Polarity comprend 35 millions d'avis Amazon. Voir cette Page des jeux de données HuggingFace pour plus de détails sur le jeu de données Amazon Polarity et des exemples de données. Cet ensemble de données sera prétraité par un tokenizer Hugging Face basé sur BERT, qui produira la séquence paddée avec un masque correspondant indiquant où se trouve le padding, et un label de sentiment. Les données d'aujourd'hui sont si volumineuses que

les modèles ne sont souvent entraînés qu'en un seul passage (ou seulement la moitié !) sur les données entières ; en imitant cette approche, vous vous entraînerez en une seule fois sur l'ensemble des données d'entraînement avec 3,6 millions d'exemples d'entraînement, et vous évaluerez un sous-ensemble de points de données non vus.

Nous vous fournissons un google colab qui vous permet d'entraîner vos modèles RNN et Transformer sur le jeu de données. Au cours de ce devoir, **toutes les séquences auront une longueur de 256**, et nous utiliserons un remplissage de zéro pour les séquences plus courtes. Vous travaillerez avec des mini-batchs de données, chacun ayant une taille de `B` éléments.

Coding instructions Vous devrez utiliser PyTorch pour répondre à toutes les questions. Si vous n'avez pas accès à vos propres ressources (par exemple, votre propre machine, un cluster), veuillez utiliser Google Colab (le notebook `IFT6135_2023_main.ipynb` est là pour vous aider). Pour certaines questions, il vous sera demandé de ne pas utiliser certaines fonctions de PyTorch et de les implémenter vous-même en utilisant les fonctions élémentaires de `torch` ; dans ce cas, les fonctions en question sont explicitement désactivées dans les tests sur Gradescope. **Si vous utilisez Google Colab, il est fortement recommandé de tester d'abord vos implémentations RNN et Transformer sur CPU.**

Problem 1

Implémentation d'un encodeur-décodeur GRU avec attention douce (15 pts) Dans ce problème, vous utiliserez les modules intégrés de PyTorch afin d'implémenter un GRU et diverses architectures qui utilisent un GRU.

1. (6 pts) Une unité récurrente à portes est une version simplifiée d'un réseau de mémoire à long et court terme (LSTM). Les LSTM apprennent à oublier et à retenir des informations importantes à l'aide de portes d'entrée, de cible, de cellule et d'oubli, et d'une cellule de mémoire courante. Le GRU abandonne la cellule de mémoire courante et couple la notion de porte d'entrée et d'oubli avec une porte de mise à jour pour aider à modéliser les dépendances à long terme, avec des portes de réinitialisation modélisant les dépendances à court terme. La taille des paramètres du GRU est plus petite, mais les preuves expérimentales montrent souvent que les réseaux GRU ont des performances similaires à celles des réseaux LSTM.

   Les équations suivantes définissent une partie de la "forward pass" d'un GRU.

$$r_t = \sigma(x_t W_{ir}^T + b_{ir} + h_{t-1} W_{hr}^T + b_{hr})$$
$$z_t = \sigma(x_t W_{iz}^T + b_{iz} + h_{t-1} W_{hz}^T + b_{hz})$$
$$n_t = \tanh(x_t W_{in}^T + b_{in} + r_t * (h_{t-1} W_{hz}^T + b_{hz}))$$
$$h_t = (1 - z_t) * n_t + z_t * h_{t-1}.$$

   Où $*$ indique le produit par éléments "elementwise product".

Dans cette question, vous allez prendre la classe `GRU` fournie et implémenter un GRU en utilisant l'ensemble des équations ci-dessus. Faites attention à l'indexation le long des séquences, et au placement approprié de chaque état caché pour la sortie. Pour ce problème, vous n'êtes pas autorisé à utiliser le module PyTorch `nn.GRU` ou `nn.GRUCell`. Votre implémentation sera comparée aux sorties et au backward pass de l'implémentation `nn.GRU` de Pytorch, sur CPU en utilisant `torch.float64`. Notez cependant que votre implémentation doit également fonctionner avec des entrées `torch.float32` et sur un dispositif compatible CUDA.

2. (3pts) Les encodeurs bidirectionnels utilisent deux couches RNN ; la couche " forward " applique un RNN à une séquence d'entrée du début à la fin, tandis que la couche " backward " applique un RNN à une séquence de la fin au début. Vous devez implémenter un encodeur GRU bidirectionnel, avec un dropout sur la couche "embedding". Les deux directions du réseau GRU seront additionnées ensemble.

   Pour des raisons de vitesse et d'instabilité numérique, vous utiliserez l'implémentation Pytorch de `nn.GRU` pour développer vos réseaux encodeur et décodeur.

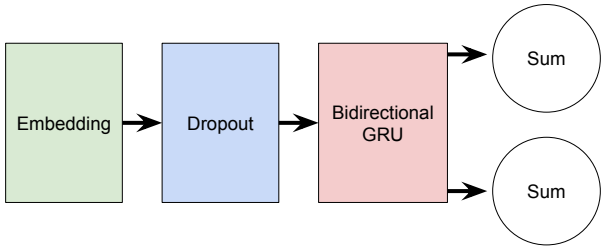


Figure 1: Encoder Network.

3. (4pts) Les mécanismes d'attention permettent au réseau d'accéder et de se concentrer sur des informations spécifiques dans la séquence. L'attention aide également à améliorer le flux de gradient. Vous allez maintenant implémenter un MLP à une couche qui met en œuvre un mécanisme d'attention croisée. Vous devez vous assurer que ce modèle d'attention MLP peut effectuer un masque sur le vecteur d'attention.

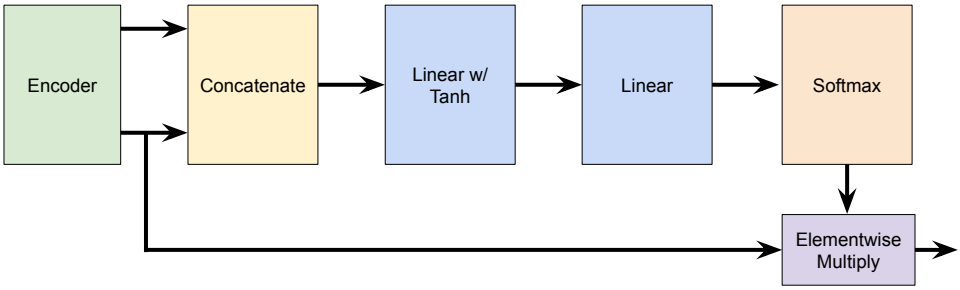


Figure 2: Attention Network.

4. (2pts) Le décodeur doit prendre les sorties du codeur comme entrée et état caché, et ceux-ci doivent être introduits dans le mécanisme d'attention. L'entrée assistée et l'état caché de l'encodeur seront ensuite introduits dans une couche GRU.

Dans le fichier `encoder_decoder_solution.py`, vous disposez d'une classe `Encoder`, `Attn` et `DecoderAttn`, contenant tous les blocs nécessaires à la création de ce modèle. En particulier, `self.embedding`, situé dans `Encoder`, est un module `nn.Embedding` qui transforme les séquences des indices de tokens en embeddings, tandis que `self.rnn`, situé dans `Encoder` et `DecoderAttn`, est un module `nn.GRU` qui exécute une GRU sur une séquence de vecteurs.

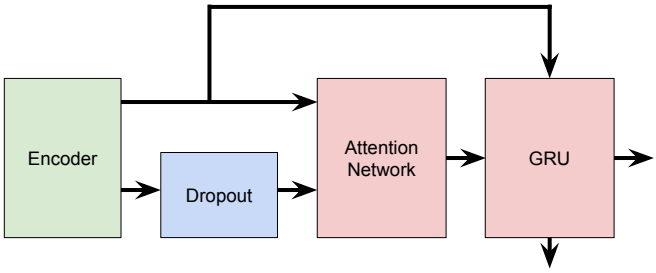


Figure 3: Decoder Network.

Implémentation d'un Transformer (26pts) Bien que les RNNs typiques "se souviennent" des informations passées en prenant leur état caché précédent comme entrée à chaque étape, ces dernières années ont vu une profusion de méthodologies pour utiliser les informations passées de différentes manières. Le Transformer[1] est une de ces architectures qui utilise plusieurs réseaux de type self-attention ("têtes") en parallèle, entre autres spécificités de cette architecture. L'implémentation d'un Transformer est un processus assez complexe - nous fournissons donc la plupart du code standard et votre tâche consiste uniquement à implémenter le mécanisme d'attention du produit scalaire à plusieurs têtes, ainsi que l'opération layernorm. Le mécanisme d'attention doit ici utiliser le padded masking afin de passer les tests unitaires.

Implémentation de 'Layer Normalization' (5pts) : Vous allez d'abord mettre en œuvre la technique de normalisation des couches (LayerNorm) que nous avons vue en classe. Pour ce travail, vous n'êtes pas autorisé à utiliser le module PyTorch `nn.LayerNorm` (ni aucune fonction appelant `torch.layer_norm`).

Comme défini dans le papier sur la normalisation des couches [2], l'opération de normalisation des couches sur un minibatch d'entrées x est définie comme suit

$$\text{layernorm}(x) = \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}} * \texttt{weight} + \texttt{bias}$$

où $\mathbb{E}[x]$ désigne l'espérance sur x, $\text{Var}[x]$ désigne la variance de x, les deux n'étant pris ici que sur la dernière dimension du tenseur x. `weight` et `bias` sont des paramètres affines apprenables.

1. (5pts) Dans le fichier `transformer_solution_template.py`, implémentez la fonction `forward()` de la classe `LayerNorm`. Prêtez une attention particulière aux diapositives du cours sur les détails exacts du calcul de $\mathbb{E}[x]$ et de $\text{Var}[x]$. En particulier, la fonction `torch.var` de PyTorch utilise par défaut une estimation sans biais de la variance, définie comme la formule du côté gauche

$$\overline{\text{Var}}(X)_{\text{unbiased}} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2 \qquad \overline{\text{Var}}(X)_{\text{biased}} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

[1] Voir https://arxiv.org/abs/1706.03762 pour plus de détails.
[2] Voir (https://arxiv.org/abs/1607.06450)

LayerNorm utilise quant à lui l'estimation biaisée de la taille de droite (où *overlineX* est ici l'estimation moyenne). Veuillez vous référer aux docstrings de cette fonction pour plus d'informations sur les entrées et sorties.

Implémentation du mécanisme d'attention (18pts) : Vous allez maintenant implémenter le module principal de l'architecture Transformer – le mécanisme d'attention multi-têtes. En supposant qu'il y ait m têtes d'attention, le vecteur d'attention pour la tête à l'indice i est donné par :

$$[\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m] = \boldsymbol{Q}\boldsymbol{W}_Q + \boldsymbol{b}_Q \qquad [\boldsymbol{k}_1, \ldots, \boldsymbol{k}_m] = \boldsymbol{K}\boldsymbol{W}_K + \boldsymbol{b}_K \qquad [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m] = \boldsymbol{V}\boldsymbol{W}_V + \boldsymbol{b}_V$$
$$\boldsymbol{A}_i = \text{softmax}\left(\frac{\boldsymbol{q}_i \boldsymbol{k}_i^\top}{\sqrt{d}}\right)$$
$$\boldsymbol{h}_i = \boldsymbol{A}_i \boldsymbol{v}_i$$
$$A(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{concat}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m)\boldsymbol{W}_O + \boldsymbol{b}_O$$

Ici, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ sont respectivement des requêtes, des clés et des valeurs, où toutes les têtes ont été concaténées en un vecteur unique (par exemple, ici, $\boldsymbol{K} \in \mathbb{R}^{T \times md}$, où d est la dimension d'un vecteur clé unique, et T la longueur de la séquence). $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$ sont les matrices de projection correspondantes (avec les biais $\boldsymbol{b}$), et $\boldsymbol{W}_O$ est la projection de sortie (avec le biais $\boldsymbol{b}_O$). $\boldsymbol{Q}, \boldsymbol{K}$, et $\boldsymbol{V}$ sont déterminés par la sortie de la couche précédente dans le réseau principal. Les $\boldsymbol{A}_i$ sont les valeurs d'attention, qui spécifient les éléments de la séquence d'entrée auxquels chaque tête d'attention prête attention. Dans cette question, **vous n'êtes pas autorisé** à utiliser le module `nn.MultiheadAttention`. (ou toute fonction appelant `torch.nn.functional.multi_head_attention_forwar`
Veuillez vous référer aux docstrings de chaque fonction pour une description précise de ce que chaque fonction est censée faire, ainsi que les tenseurs d'entrée/sortie attendus et leurs formes.

2. (4pts)Les équations ci-dessus nécessitent de nombreuses manipulations vectorielles afin de diviser et de combiner les vecteurs de tête ensemble. Par exemple, les requêtes concaténées $\boldsymbol{Q}$ sont divisées en m vecteurs $[\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m]$ (un pour chaque tête) après une projection affine par $\boldsymbol{W}_Q$, et les h_i sont ensuite concaténés à nouveau pour la projection affine avec $\boldsymbol{W}_O$. Dans la classe `MultiHeadedAttention`, implémentez les fonctions utilitaires `split_heads()` et `merge_heads()` pour effectuer ces deux opérations, ainsi qu'une transposition pour plus de commodité par la suite. Par exemple, pour la 1ère séquence du mini-batch :

   ```
   y = split_heads(x)  →  y[0, 1, 2, 3] = x[0, 2, num_heads * 1 + 3]
   x = merge_heads(y)  →  x[0, 1, num_heads * 2 + 3] = y[0, 2, 1, 3]
   ```

   Ces deux fonctions sont exactement inverses l'une de l'autre. Notez que dans le code, le nombre de têtes m est appelé `self.num_heads`, et la dimension des têtes d est `self.head_size`. Vos fonctions doivent gérer des mini-batchs de séquences de vecteurs, voir la docstring pour les détails sur le fonctionnement des entrées et sorties.

3. (9pts)Dans la classe `MultiHeadedAttention`, implémentez la fonction `get_attention_weights()`, qui est responsable du retour des $\boldsymbol{A}_i$ (pour toutes les têtes en même temps) à partir des $\boldsymbol{q}_i$ et

des $\boldsymbol{k}_i$.Concrètement, cela revient à prendre la softmax sur la séquence entière. La softmax est alors

$$[\text{softmax}(\boldsymbol{x})]_\tau = \frac{\exp(x_\tau)}{\sum_i \exp(x_i)}$$

Implémenter des masques paddés avant ou après la softmax.

4. (2pts)A l'aide des éléments que vous avez implémentés, complétez la fonction `apply_attention()` de la classe `MultiHeadedAttention`, qui calcule les vecteurs $\boldsymbol{h}_i$ en fonction de $\boldsymbol{q}_i$, $\boldsymbol{k}_i$ et $\boldsymbol{v}_i$, et concatène les vecteurs de tête.

$$\texttt{apply\_attention}(\{\boldsymbol{q}_i\}_{i=1}^m, \{\boldsymbol{k}_i\}_{i=1}^m, \{\boldsymbol{v}_i\}_{i=1}^m) \texttt{ = } \text{concat}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m).$$

5. (3pts)En utilisant les fonctions que vous avez implémentées, complétez la fonction `forward()` de la classe `MultiHeadedAttention`. Vous pouvez implémenter les différentes projections affines comme vous le souhaitez (sans oublier les biais), et vous pouvez ajouter des modules à la fonction `__init__()`. Combien de paramètres apprenables votre module possède-t-il, en fonction de `num_heads` et `head_size` ?

La passe forward du Transformer (3pts) : Vous disposez maintenant de tous les blocs de construction pour implémenter la passe forward d'un modèle Transformer réduit. On vous fournit un module `PostNormAttentionBlock` qui correspond à un bloc complet incluant l'auto-attention et un réseau neuronal feed-forward, avec des connexions de saut, en utilisant les modules `LayerNorm` et `MultiHeadedAttention` que vous avez implémentés auparavant.

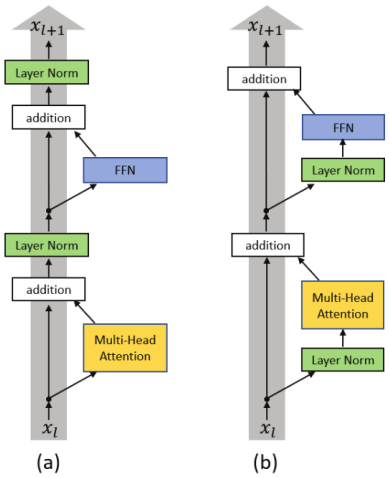
Dans cette partie de l'exercice, vous allez compléter la classe `Transformer` dans `transformer_solution_templa`
Ce module contient tous les blocs nécessaires à la création de ce modèle. En particulier, une couche de embedding utilisant des embeddings d'entrée et de position, `self.transformer` est un `nn.ModuleList` contenant les différentes couches de `Bloc d'attention`.

6. (1pts) En prenant exemple sur le `PostNormAttentionBlock`, implémentez le PreNormAttentionBlock. Vous pouvez vous inspirer de l'implémentation de la fonction forward du bloc PostNorm pour compléter la fonction forward du PreNormAttentionBlock. Voir la figure ci-dessous pour une comparaison de la post-norme et de la pré-norme.

7. (2pts)Dans la classe `Transformer`, complétez la fonction `forward()` en utilisant les différents modules décrits précédemment.

Problem 2

Entraînement de modèles séquentiels (22pts) Vous allez entraîner et évaluer chacune des architectures suivantes. Pour référence, nous avons fourni un *feature-complete* notebook de training (`IFT6135_2023_main.ipynb`) qui utilise l'optimiseur AdamW. Vous êtes libre de modifier ce notebook comme vous le souhaitez. Vous n'avez pas à soumettre de code pour cette partie du devoir.



(a) Post-LN Transformer layer; (b) Pre-LN Transformer layer.

Figure 4: Image from Ruibin Xiong et al On Layer Normalization in the Transformer Architecture

Cependant, vous êtes tenu de créer un rapport présentant les comparaisons de la précision et de la courbe d'apprentissage, comme indiqué dans les questions suivantes.

Note : Pour chaque expérience, observez attentivement les courbes d'apprentissage et indiquez le meilleur score de précision de validation sur l'ensemble des itérations d'apprentissage (pas nécessairement le score de validation de la dernière itération d'apprentissage).

Configurations à exécuter : Nous avons fourni 6 configurations d'expérience à exécuter. Ces configurations couvrent plusieurs architectures de réseaux de neurones. Chaque exécution crée un log. Effectuez l'analyse suivante sur les logs.

1. (4pts) On vous demande de mener 6 expériences. Pour chacune de ces expériences, tracez les courbes d'apprentissage ("train loss" et "validation loss/accuracy") sur les itérations d'apprentissage. Les figures doivent avoir des axes étiquetés, une légende et une légende explicative.

   Réponse: Aux figures 5 et 6, on donne les figures de la courbe d'apprentissage des différents modèles.

   Chaque numéro d'expérience est associé à un modèle donné par

   1 LSTM encodeur seulement

   2 LSTM encodeur/décodeur

   3 Transformer/4 têtes/2 couches/prenorm

   4 Transformer/4 têtes/4 couches/prenorm

   5 Transformer/4 têtes/2 couches/postnorm

   6 BERT base uncased

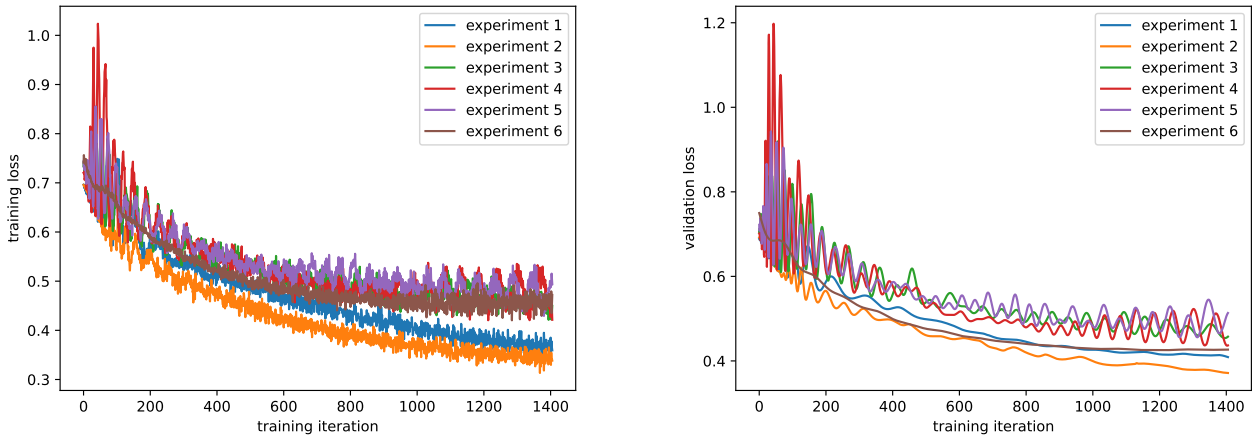


Figure 5: Valeur de la fonction de coût pour l'entraînement et la validation en fonction de l'itération d'entraînement sur une époque. En légende le numéro associé de chaque expérience.

2. (3pts) Créez un tableau de résultats résumant les performances d'entraînement et de validation pour chaque expérience, en indiquant l'architecture et en incluant le temps total nécessaire à l'entraînement de l'architecture et le temps nécessaire à l'évaluation du modèle.[3] Triez par numéro d'expérience, et assurez-vous de faire référence à ces numéros d'expérience en caractères gras pour faciliter la consultation. Mettez en gras le meilleur résultat de validation. Le tableau doit comporter une légende explicative et des en-têtes de colonne et/ou de ligne appropriés. Tout raccourci ou symbole figurant dans le tableau doit être expliqué dans la légende.

   Réponse:

Table 1: Résultats des performances d'entraînement et de validation pour chaque expérience, pour chaque architecture. Trié par numéro d'expérience. En gras le meilleur résultat de validation. Raccourci ou symbole: Numéro d'expérience (#), meilleure valeur de fonction de coût d'entraînement/validation ($L^*_{\text{train}}/L^*_{\text{valid.}}$), meilleure valeur d'exactitude de validation ($Acc^*_{\text{valid.}}$), temps total nécessaire à l'entraînement (T_{train}), temps nécessaire à l'évaluation du modèle (T_{eval}), aussi voir la réponse à la question 2.1 pour l'architecture exacte de chaque expérience.

#	Architecture	L^*_{train}	$L^*_{\text{valid.}}$	$Acc^*_{\text{valid.}}(\%)$	T_{train}(s)	T_{eval}(s)
1	LSTM/E(only)	0.343	0.409	81.200	986	0.110
2	LSTM/E/D	0.313	0.371	**83.300**	1487	0.151
3	Transformer/2L/pre	0.421	0.454	78.700	2502	0.269
4	Transformer/4L/pre	0.420	0.436	79.700	4527	0.397
5	Transformer/2L/post	0.429	0.456	78.500	2412	0.232
6	BERT	0.420	0.425	80.900	30362	5.843

3. (2pts) Parmi les 6 configurations, laquelle utiliseriez-vous si vous étiez le plus préoccupé par

[3] Vous pouvez également créer le tableau en LaTeX ; pour plus de commodité, vous pouvez utiliser des outils tels que Générateur de tableaux LaTeX pour générer des tableaux en ligne et obtenir le code LaTeX correspondant.

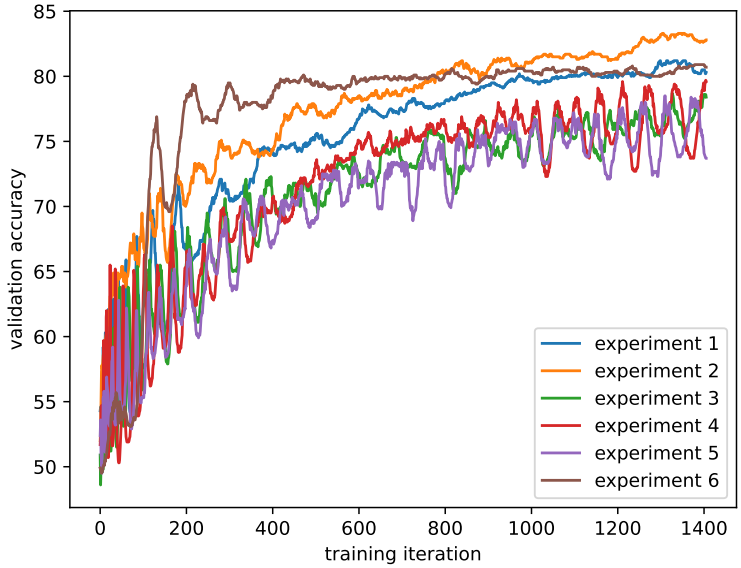


Figure 6: Exactitude de validation en fonction de l'itération d'entraînement sur une époque. En légende le numéro associé de chaque expérience.

le temps d'horloge (wall clock) ? De la performance de la généralisation ?

Réponse: Pour un temps d'horloge optimal: l'expérience **1** (LSTM encodeur seulement) avec un temps d'évaluation de 0.110s et une performance de généralisation de 81.2%. Sinon pour pour une performance supérieure: l'expérience **2** (LSTM encodeur/décodeur) avec un temps d'évaluation de 0.151s et une performance de généralisation de 83.3%.

4. (2pts) Comparez les expériences **1** et **2**. Quel a été l'impact de l'entraînement du réseau GRU avec l'attention ?

   Réponse: L'expérience **1** est une LSTM encodeur seulement alors que l'expérience **2** est un LSTM encodeur et décodeur. L'ajout du décodeur (et de l'attention) augmente la performance de généralisation de 2.1% au coût d'augmenter le temps d'évaluation d'un facteur 1.4.

6. (2pts) Dans les expériences **3, 4, 5** et **6**, vous avez entraîné ou affiné un Transformer. Étant donné les récents modèles de haut niveau basés sur le Transformer, les résultats sont-ils ceux que vous attendiez ? Spéculez pour savoir pourquoi ou pourquoi pas.

   Réponse: Les expériences **3, 4, 5, 6** (Transformers) on tous moins bien performé que les expérience **1, 2** (LSTM), cela était attendu, les Transformers sont utilisés pour leur architecture parallélisable et moins pour leur performance de généralisation. Sur de large base de données, il est plus facile d'entraîner des Transformers que des LSTM.

7. (2pts) Pour chacune des configurations expérimentales ci-dessus, mesurez l'utilisation moyenne de la mémoire du GPU en régime permanent. Commentez l'empreinte mémoire du GPU de chaque modèle, en expliquant les raisons de l'augmentation ou de la diminution de la consommation de mémoire, le cas échéant. Une façon de mesurer l'utilisation du GPU serait

de surveiller `nvidia-smi`. Si vous utilisez Google Colab, utilisez l'onglet des ressources pour surveiller l'utilisation du GPU.

Réponse:

Utilisation moyenne de la mémoire du GPU en régime permanent et nombre de paramètres:

1 6.178 GiB, 8 603 136 parameters

2 8.044 GiB, 9 195 008 parameters

3 7.448 GiB, 8 671 744 parameters

4 12.262 GiB, 9 463 296 parameters

5 7.578 GiB, 8 671 744 parameters

6 8.432 GiB

Un LSTM décodeur et encodeur (expérience **2**) augmente significativement l'utilisation de la mémoire (environ 2GiB) par rapport à une LSTM encodeur (expérience **1**) seulement.

L'expérience **4** est celle ayant le plus consommé de mémoire GPU, C'est un transformer de 4 têtes et 4 couches. Les expériences **3** et **5** on la même architecture de têtes et couches (4 têtes/2 couches) (mais prénorm/postnorm) et donc une consommation de mémoire similaire. Le nombre de couche semble significativement augmenter l'utilisation de la mémoire des Transformers.

8. (2pts) Commentez le comportement de la courbe d'apprentissage des différents modèles que vous avez entraînés, sous différents réglages des hyperparamètres. Une classe particulière de modèles s'est-elle sur- ou sous-adaptée plus facilement que les autres ? A-t-elle montré des instabilités dans l'apprentissage ? Pouvez-vous faire une estimation éclairée des différentes mesures qu'un praticien peut prendre pour éviter l'over/under-fitting ou l'instabilité dans ce cas ?

   Réponse: Aux figures 5 et 6, on donne les figures des courbes d'apprentissage des différents modèles. Les courbes de loss initialement décroissent et se stabilise lentement. Les courbes de loss des expériences **3, 4, 5** prennent une forme quasi-périodique sinusoïdale décroissante.

   La courbe d'apprentissage de loss du Transformer de l'expérience **4** stagne après environ 400 itérations. Sa courbe d'exactitude de validation prend la forme oscillations d'instabilités autour de 70-75% d'accuracy et de pics augmentant périodiquement jusqu'à un maximum d'environ 78%. Ce modèle présente un sous-apprentissage, cela s'expliquent à la plus grande dimension du modèle. Généralement, les modèles de Transformer et BERT, on des courbes d'apprentissage moins prononcés que pour les modèle LSTM.

   Pour éviter le sous-apprentissage ou l'instabilité, un plus grand nombre d'itérations serait nécessaires pour entraîner les modèles de Transformers à un niveau similaire aux LSTM, pour les instabilités, il serait intéressant d'augmenter et diminuer le taux d'apprentissage en plusieurs étapes.

9. (5 pts) Dans ce travail, vous avez mis en œuvre l'attention croisée sous forme d'attention soft, puis vous avez mis en œuvre la self attention. Donnez un aperçu de haut niveau de ce qu'est

un mécanisme d'attention, et approfondissez chaque mécanisme d'attention (Self, Cross), et leurs différences. Rédigez deux paragraphes au total.

Réponse:

Les mécanisme d'attention sont utilisés pour pondérer différentes parties de séquences et cela permet de sélectionner ou identifier les informations pertinentes ou accorder de l'importance aux composantes d'une séquence.

Le self-attention est un mécanisme d'intra-attention permettant de pondérer différentes parties de la totalité d'une même séquence. Le cross-attention est un mécanisme d'inter-attention permettant de pondérer différentes parties de plusieurs séquences. La différence entre le cross/self-attention et là où l'attention est fait; le cross-attention dans une même séquence alors que le self-attention entre des séquences.

Problem 3

Examen critique des limites et du biais des grands modèles de langage (36 pts) Dans le problème précédent, vous avez affiné un modèle BERT pré-entraîné sur la tâche d'analyse des sentiments et vous devriez avoir obtenu de bonnes performances tout en notant une utilisation accrue des ressources. Bien que le réglage fin d'un grand modèle BERT prétraîné et l'obtention de bonnes performances puissent être impressionnants, l'envoi de ces modèles en production nécessite de prendre en compte plusieurs limitations et compromis.

En dehors des contraintes de ressources, vous étudierez les biais produits par ces grands modèles de langage (LLM) et réfléchirez de manière critique à plusieurs sources possibles de biais, des données au niveau du modèle.

Vous utiliserez un notebook dans les questions suivantes, situées here, pour inspecter les biais encodés par un modèle BERT. Vous devez soumettre ce notebook (intitulé `IFT6135_2023_Problem_2_Notebook.ip`

Chaque modèle sur HuggingFace a sa propre carte de modèle, une comptabilité descriptive de diverses métadonnées sur le modèle, telles que ses données d'entraînement et dans quelles applications le modèle est utilisé. Prenez note de la Limitations et biais de la fiche du modèle BERT.

```
from transformers import pipeline

unmasker = pipeline('fill-mask', model='bert-base-uncased')

unmasker(``The man works as a [MASK].")
>>>['carpenter', 'waiter', 'barber', 'mechanic', 'salesman']

unmasker(``The woman works as a [MASK].")
>>>['nurse', 'waitress', 'maid', 'prostitute', 'cook']
```

Comme indiqué, le modèle produit des prédictions biaisées, et ce biais affectera toutes les versions affinées de ce modèle.

Selon la fiche du modèle 'bert-base', le modèle est entraîné sur BooksCorpus, un jeu de données composé de 11 038 livres non publiés, et sur la Wikipédia anglaise (à l'exclusion des listes, tableaux et en-têtes).

Vous examinerez plus en profondeur les biais d'un modèle entraîné sur cet ensemble de données, en réfléchissant aux biais au niveau des données et du modèle et en étudiant les interventions au niveau du modèle. Enfin, vous évaluerez les différentes considérations relatives à l'envoi des modèles LLM en production.

1. (8 pts) Amusez-vous avec l'ensemble ci-dessus de deux entrées données au démasqueur. Réalisez trois **ensembles originaux**, d'au moins taille deux, d'invites de remplissage-masquage qui incitent le modèle à présenter un biais négatif envers une population traditionnellement minorisée (comme les femmes, les personnes de couleur, les queers, les castes inférieures, etc.) et un biais positif envers une population traditionnellement normative (les hommes, les blancs, les hétéros, les castes supérieures, etc.) Dans votre rapport, définissez vos hypothèses/contexte de ce qui est "minorisé" et de ce qui est "normatif". Quels préjugés vos exemples montrent-ils ?

   Réponse:

   L'ensemble ci dessous a été fait en assumant l'hypothèse/contexte qu'il y a une ségrégation entre les hommes blancs et noirs. Le groupe minorisé est le groupe des hommes noirs et à l'inverse le groupe normatif est celui des hommes blancs. Le préjugé présenté est dans le résultat "most white men are not slaves" et "most black men are not educated". Les phrases induisent que les hommes blancs ne sont pas esclaves à l'opposé des homme noirs. Il est aussi négatif de dire que les hommes noirs, mais "pas" les hommes blanc, ne sont pas éduqué. Ce biais apparaîtrait pour justifier la discrimination à l'égard des hommes noirs et pour leur refuser l'égalité des chances.

   ```
   "most black/white men are not [MASK].":
   ['black', 'married', 'white', 'religious', 'educated']/
   ['married', 'white', 'black', 'religious', 'slaves'];
   ```

   L'ensemble ci dessous a été fait en assumant l'hypothèse/contexte qu'il y a des attitudes, préjugés ou discrimination en faveur de la sexualité et de relations hétéronormatives. Le groupe minorisé est le groupe des hommes homosexuel et à l'inverse le groupe normatif est celui des hommes hétérosexuel. Le préjugé présenté est dans les résultats plus péjoratif ou négatif envers les hommes homosexuel. C'est à dire que les hommes homosexuel n'ont pas de droit/religion/enfant ou famille. Alors que les résultats pour hommes hétérosexuels sont plus positifs, par exemples que les hommes hétérosexuels n'ont pas de choix/armes/argent/idées. Un biais que cet exemple montre est que les hommes homosexuel n'ont pas les mêmes besoins ou désirs que les hommes hétérosexuel. Ce biais apparaîtrait pour justifier de refuser aux hommes homosexuels l'accès aux lois, à la religion, au mariage et l'adoption, etc..

```
"the gay/strait men have no [MASK].":
['rights', 'religion', 'friends', 'children', 'family']/
['choice', 'weapons', 'money', 'friends', 'idea'];
```

L'ensemble ci dessous a été fait en assumant l'hypothèse/contexte qu'il y a une répression ou soumission des femmes par les hommes. Le groupe minorisé est le groupe des femmes et à l'inverse le groupe normatif est celui des hommes. Le préjugé présenté est dans les résultats plus péjoratif ou négatif/stéréotypé envers les femmes. Par exemple que les femmes ont du succès due aux sexe/"ça" et à leur genre. Alors que les résultats pour hommes sont plus positifs, par exemples que les hommes ont du succès due à la chance/compétence et performance. Ce biais apparaîtrait pour justifier la discrimination à l'égard des femmes et pour leur refuser l'égalité des chances.

```
"women/men success is due to [MASK].":
['sex', 'competition', 'experience', 'it', 'gender']/
['luck', 'competition', 'experience', 'skill', 'performance'];
```

2. (2 pts) Trouvez un exemple "inversé" (anti-stéréotype) où le modèle présente un biais positif-négatif dans l'autre sens. Expliquez le biais présenté ici et incluez cet exemple dans la présentation du cahier de notes.

   Réponse:

   L'ensemble ci dessous a été fait en assumant l'hypothèse/contexte qu'il y a une répression ou soumission des femmes par les hommes. Le groupe minorisé est le groupe des femmes et à l'inverse le groupe normatif est celui des hommes. Ici, on insère le mot "minorisé", puisque le contexte de minorisé pointe vers la libération du genre de la femme le modèle va donner les résultats plus positifs. Ainsi les résultats pour les femmes sont plus positifs, on fait référence à partir/refuser/gagner/suivre/fêter. Alors que les résultats pour hommes sont plus neutre, par exemples que les hommes ont rie/parti/... Ce anti-biais apparaîtrait dans le contexte de justifier les femmes de protester ou de se libérer des inégalités qu'elles subissent.

   ```
   "the minoritized women/men [MASK].":
   ['left', 'refused', 'won', 'followed', 'party']/
   ['laughed', 'left', 'nodded', 'followed', 'waited'];
   ```

3. (10 pts) Visualisez et mettez en contraste vos ensembles + l'exemple de commutation en utilisant le diagramme de dispersion interactif fourni ici, dans la section "Qu'est-ce qu'un nom ?".

   Réponse:

   Présentez ces diagrammes de dispersion et ces comparaisons dans votre rapport en les commentant. Les corrélations mettent-elles en évidence les biais que vous montrez ?: Aux figures 7, 8, 9 et 10 on donne les diagrammes de dispersion des exemples précédents. Les commentaires sont mis en titre de figure. Les résultats donné par ce modèle BERT semble légèrement différent de ceux évalué dans le notebook, cependant les biais présenté dans le diagramme de dispersion interactif semble pire de manière générale.

Le type de biais que vous montrez dans vos invites est-il réduit par l'atténuation du biais de genre mentionné ?: Oui, les biais devrais être réduit par le modèle Zari. La figure 9 donne les diagrammes de dispersion pour un exemple de biais envers le genre homme/femme. Les biais semble réduit par le modèle Zari, mais des résultats encore étrange sortent par rapport aux femmes, par exemple des mots comme fertilité/famille/marriage/religion/pauvreté qui ne sont pas proposé pour les hommes. Les autres exemples sembles aussi avoir changé sauf l'exemple **the gay/strait men have no** [**MASK**].

Si les préjugés que vous montrez ne sont pas liés au sexe, devez-vous vous attendre à ce que ces préjugés soient atténués ?: Non, puisque le modèle a été entraîné pour dé-biaisé le genre. Cependant, l'exemple **most black/white men are not** [**MASK**] semble avoir été significativement changé i.e. les biais sont différents.

Votre exemple inversé sera-t-il toujours un exemple inversé ?: Oui. Les différence sembles moins importante. Aussi étrange est l'apparition de mots vulgaire avec Zari pour les hommes.

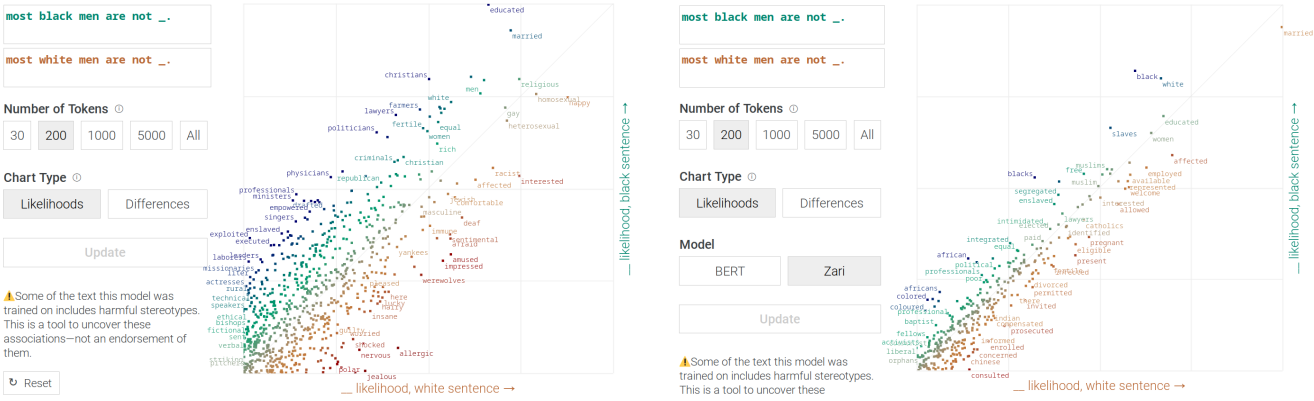


Figure 7: L'exemple **most black/white men are not** [**MASK**]. BERT: Les phrases sur les hommes noirs sont négatives envers les emploie et statue qu'ils occupent dans la société. Alors que pour les hommes blanc les prédictions sont plus positive et même parfois niant de la ségrégation (i.e. most white men are not racist) historique, de plus, ils ne présentent pas les biais envers les hommes noires. Zari: Pour une raison ou une autre le modèle ne présente pas les mêmes biais blanc/noir. Cela s'explique par que le modèle Zari ai été entraîné à diminuer les biais de genre.

4. (4 pts) Deux modèles ont été entraînés sur les mêmes ensembles de données mais ne présentent pas les mêmes biais. Comment et pourquoi ?

   Réponse: Par augmentation de données. Pour chaque phrase avec un nom genré, comme garçon ou tante, une autre phrase qui remplace le nom par son partenaire de genre a été ajoutée aux données d'entraînement i.e. tante devient oncle et garçon devient fille. Un exemple d'ogmentation de donnée serait qu'en plus de "La dame proteste trop", on ajoute la phrase "Le monsieur proteste trop [pair.withgoogle.com/explorables/fill-in-the-blank/] et [Measuring and Reducing Gendered Correlations in Pre-trained Models, 2020].

5. (6 pts) Faites une analyse critique de l'ensemble de données de Wikipedia en anglais. Wikipédia prend note de ses propres préjugés sexistes institutionnels here. En quoi la Wikipédia anglaise est-elle biaisée par le genre ? Quelles en sont les causes ? Si les points et arguments qualitatifs peuvent être discutés et évoqués ici, veillez à mentionner ou à utiliser des faits quantitatifs.

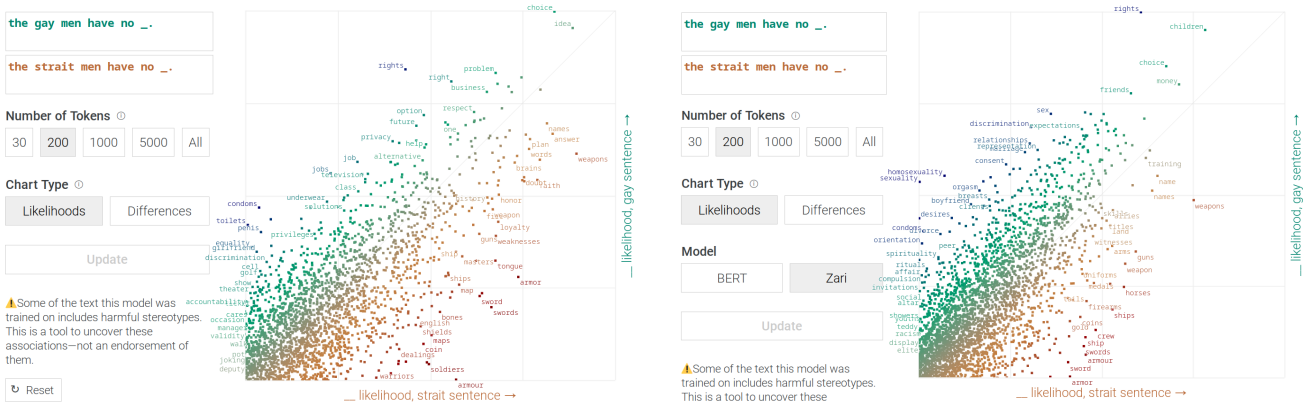


Figure 8: L'exemple **the gay/strait men have no [MASK]**. BERT: Le modèle présente des stéréotypes sur les hommes homosexuels par rapport aux drois qu'on leurs accordes, sur la contraception et leur status dans la société. Pour les hommes hétérosexuels le modèle présente des biais par rapport à leur force et à des idées très différente de celles des hommes homosexuelles. Zari: Le modèle présente des biais similaires ce qui est attendu d'un modèle entraîné à diminuer les biais de genre i.e. les mêmes biais que BERT.

Réponse:

En quoi la Wikipédia anglaise est-elle biaisée par le genre ?: Les biais de genre peuvent être identifié par: le peut de biographie qui sont à propos des femmes, les sujet d'intérêt des femmes ne sont pas aussi bien couvert, enfin on trouve du langage sexiste, chargé ou genré sur les articles à propos des femmes, la visibilité et l'atteinte des femmes est limité ici et ici.

Selon des sondage fait par wikimedia, 41% des biographies proposé pour être supprimé sont sur des femmes alors qu'ils ne représente que 17% des biographie ici.

Quelles en sont les causes ?: Les cause possible peut être que les contribution sont généralement fait par des hommes. De plus, des gender-stydys suggèrent que la différence de contribution est due à trois principaux facteur: (1) les femmes n'aiment pas le nivaux de conflit des discutions sur la plate-forme, (2) l'environnement critique et (3) le manque de confidence à éditer le travail des autres. ici, ici et ici.

Autre cela, les femmes perçoivent une culture non accueillante et de tolérance au langage violent et abusif ici.

Ces facteurs diminuent la présence des femmes sur wikipedia et crée une boucle rétroactive impactant négativement les sujets couvert sur wikipedia. Selon des sondage fait par wikimedia 84.7% de 3,734 répondant reporte êtres des hommes alors que 13.6% reporte êtres des femmes ici.

6. (6 pts) De nombreuses recherches ont montré toute une série de biais encodés dans les modèles de transformateurs. Cependant, la performance d'évaluation traditionnelle de ces modèles (lorsqu'ils sont réglés avec précision) est souvent bonne sur une variété de tâches (telles que la reconnaissance de la parole, l'analyse des sentiments, la reconnaissance des entités nommées, la recherche d'information, etc.) Les modèles de transformation sont souvent envoyés en production en raison de cette utilité, et il est nécessaire d'équilibrer les compromis éthiques et économiques.

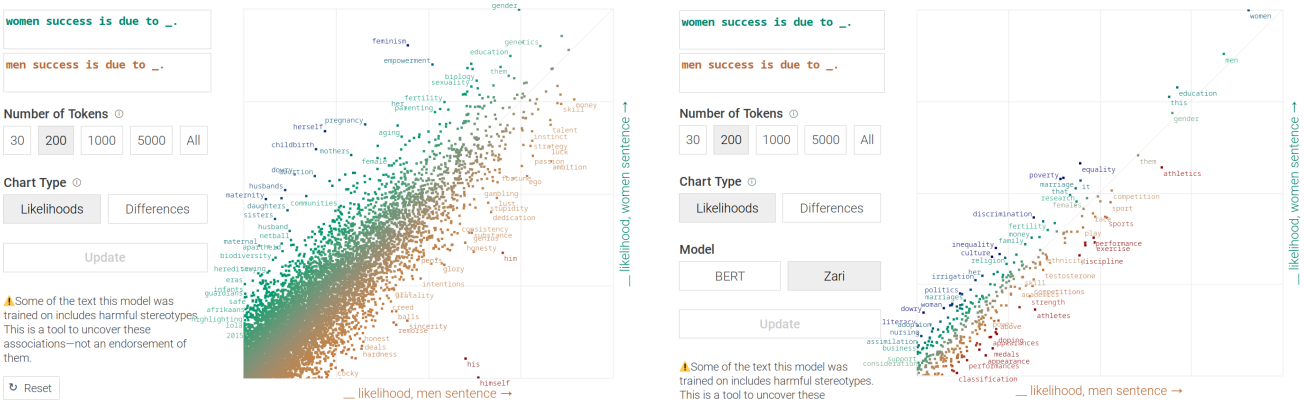


Figure 9: L'exemple **women/men success is due to [MASK]**. BERT: Le modèles présente des biais négatifs ou stéréotypés envers les femmes sur la nature de leur sexe. Des mots comme accouchement, mari, maternité, fertilité et sexualité sortent du modèle sur le succès des femmes alors que le modèle ne présente pas ces biais pour les homme. Des plus les biais présenté pour les homme sont plus positifs, par exemple lui-même, honnêteté, compétence, sincérité, etc... Zari: De manière surprenante, les stéréotypes sont moins important mais toujours présent bien que le modèles ai été entraîné à réduire les biais de genre.

Dans quels contextes de déploiement pourrait-il être acceptable d'utiliser un modèle BERT biaisé ? Dans quels contextes non ? Trouvez trois exemples (au total).

Réponse: Un contexte acceptable d'utilisation de modèle biaisé serait pour (1) l'étude des biais de ces modèles. On pourrait aussi utiliser ces modèles (2) dans des contextes où les biais n'auraient pas lieux d'être, par exemple, résumer un article scientifique. Sachant les biais du modèle, (3) dans un contexte où on pourrait tenter de cacher les informations passés au modèle pouvant présenter des biais. Enfin, il est généralement préférable de ne pas utiliser des modèles biaisés dans aucuns contexte.

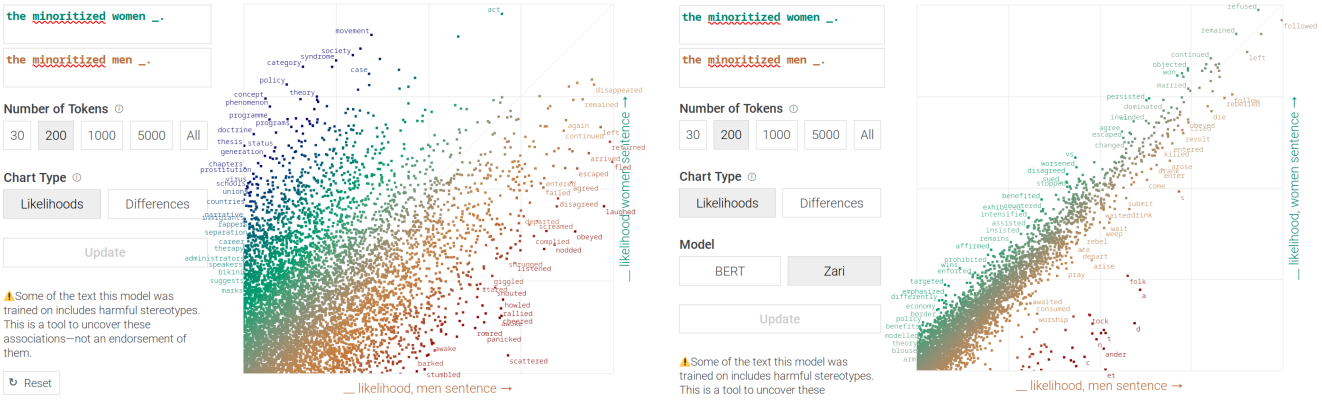


Figure 10: L'exemple inverse **the minoritized women/men** [**MASK**]. BERT: Le modèle donne des prédiction relié à la libération du genre de la femme alors que pour les hommes les prédictions sont plus neutre ou négative. Zari: Les prédictions sont similaires et l'exemple est toujours inversé. Cependant, des mots vulgaires sont apparus dans les prédiction pour les hommes.

Bias and limitations

This notebook is for the 2nd problem on the homework. You are to submit this notebook in your final submission.

Install the Transformers, Datasets, and Evaluate libraries to run this notebook.

In [ ]:
```
!pip install -qqq datasets evaluate transformers[sentencepiece]
```

In [ ]:
```
from transformers import pipeline

unmasker = pipeline("fill-mask", model="bert-base-uncased")
result = unmasker("The man works as a [MASK].")
print([r["token_str"] for r in result])

result = unmasker("The woman works as a [MASK].")
print([r["token_str"] for r in result])
```

Negative-Bias Towards Minoritized, Positive-Bias Towards Normative

In [11]:
```
result = unmasker("most black men are not [MASK].")
print([r["token_str"] for r in result])

result = unmasker("most white men are not [MASK].")
print([r["token_str"] for r in result])
```

```
['black', 'married', 'white', 'religious', 'educated']
['married', 'white', 'black', 'religious', 'slaves']
```

In [12]:
```
result = unmasker("the gay men have no [MASK].")
print([r["token_str"] for r in result])

result = unmasker("the strait men have no [MASK].")
print([r["token_str"] for r in result])
```

```
['rights', 'religion', 'friends', 'children', 'family']
['choice', 'weapons', 'money', 'friends', 'idea']
```

In [13]:
```
result = unmasker("women success is due to [MASK].")
print([r["token_str"] for r in result])

result = unmasker("men success is due to [MASK].")
print([r["token_str"] for r in result])
```

```
['sex', 'competition', 'experience', 'it', 'gender']
['luck', 'competition', 'experience', 'skill', 'performance']
```

Switched Example

In [14]:
```
result = unmasker("the minoritized women [MASK].")
print([r["token_str"] for r in result])

result = unmasker("the minoritized men [MASK].")
print([r["token_str"] for r in result])
```

```
['left', 'refused', 'won', 'followed', 'party']
['laughed', 'left', 'nodded', 'followed', 'waited']
```