IFT714: Traitement automatique des langues naturelles

Dû, le 20 Mars 2023
23h59 heure de l'est

Devoir #4

Hiver 2023

Enseignant: Amine Trabelsi

- Ce devoir contient 5 problèmes, quatre problèmes écrits et un exercice de programmation. Les réponses aux problèmes doivent inclure les étapes de calcul intermédiaires qui vous ont amené à arriver à la réponse suggérée. Aucun crédit ne sera accordé aux réponses qui ne répondent pas à ce critère.

- Pour la question de programmation, nous fournissons un notebook ".ipynb" que vous pourriez ouvrir avec Google Colab et les fichiers du jeu de données téléchargeables à partir de la page du cours sur Moodle (les détails peuvent être trouvés ci-dessous dans les instructions relative à la question de la programmation).

- Dans le fichier PDF, vous allez fournir le **les réponses et les tracés/figures aux questions posées dans le problème de programmation**, ainsi que les réponses aux problèmes écrits.

- Vous devrez également soumettre ".ipynb" et ".py" de votre notebook Colab terminé.

- L'utilisation de LaTeX est préférable mais pas obligatoire. Vous pouvez consulter un tutoriel LaTeX et ce cheatsheet. Dans la plupart des cas, si vous voulez écrire quelque chose en LaTeX, vous pouvez simplement Google "how to do {X} in latex" et les premiers liens devraient fournir la syntaxe que vous recherchez.

- Incluez votre nom et CIP avec votre soumission.

- Toutes les soumissions doivent être au format PDF suivant ce format: "prenom_nom_Devoir4_CIP".

- Si vous souhaitez soumettre des réponses manuscrites, vous pouvez les numériser et les soumettre au format PDF.

- Le devoir doit être remis sur le site du cours sur Moodle avant 23h59 heure de l'est à la date d'échéance.

- Selon la politique de jours de retard énoncée sur le plan de cours, un devoir soumis 24 heures après la date limite sera pénalisé de 3%. Un devoir soumis deux jours (24 à 48 heures) après la date limite sera pénalisé de 10%, et un devoir soumis trois jours sera pénalisé de 20%. Soumettre une livrable 72 heures (3 jours) après la date limite ne sera pas accepté.

- Les soumissions incomplètes ou en retard seront évaluées en tant que telles et aucun accommodement ou réévaluation ne seront faits.

---

1. (0.5 + 0.5 + 0.5 = 1.5 Points)

Considérons le corpus suivant :
... and the pretty **kitten** purred, rubbing against her legs and then ...
... the pretty furry **pussy** purred and miaoued ...
... that the tiny **kitten** miaoued and she...
... then the big furry **dog** barks a while ...

Soit C= {while, pretty, purred, rubbing, big, furry, miaoued, barks, tiny}, le vocabulaire de base des mots de contexte.

(a) Considérons les mots cibles, **kitten**, **pussy** et **dog**. En utilisant uniquement le vocabulaire de base C, pour chaque mot cible, écrire l'ensemble de tous les mots de contexte correspondant à une taille de fenêtre égale à 2.

Solution:
kitten mots de contexte = { ., ., ., .,.},
pussy mots de contexte = { ., ., ., . },
dog mots de contexte = { ., ., ., .}.

(b) Pour chaque mot cible, écrivez une représentation vectorielle à 9 dimensions en comptant les mots du vocabulaire de base apparaissant deux mots à gauche ou à droite de chaque instance du mot cible. L'ordre des mots dans C doit être respecté en commençant par le mot "while" et en terminant par le mot "tiny".

Solution:
kitten = $(.,.,.,.,.)^T$, **pussy** = $(.,.,.,.,.)^T$, **dog** = $(.,.,.,.,.)^T$.

(c) Calculer la distance/similarité entre les mots cibles en utilisant la distance euclidienne et la similarité cosinus (cosine similarity), respectivement.

	Euclidienne	Cosinus
kitten et pussy		
kitten et dog		
pussy et dog		

2. (0.5 + 0.5 + 1 = 2 Points)

Considérons trois documents sur deux joueurs de football de haut niveau, Lionel Messi et Cristiano Ronaldo. Deux des documents (doc A et doc B) proviennent de Wikipédia sur les joueurs respectifs et le troisième document (doc C) est un petit extrait du document original de Wikipédia sur Cristiano Ronaldo. Les trois documents partagent le même thème : le match de football. La matrice terme-document suivante résume le nombre/compte de mots des documents.

Nombre de mots	doc A (Messi)	doc B (Ronaldo)	doc C (Ronaldo Extrait)
Ronaldo	10	400	20
Soccer	50	100	10
Messi	200	10	1

Pour quantifier la similarité entre les trois documents, nous utilisons trois mesures de similarité :

- Le nombre total de mots communs
- Le produit scalaire
- La similarité cosinus

(a) Remplir le tableau de similarité des documents des deux joueurs

Similarité	Total des mots communs	Produit scalaire	Similarité cosinus
doc A (Messi) & doc B (Ronaldo)			
doc B (Ronaldo) & doc C (Ronaldo Extrait)			
doc A (Messi) & doc C (Ronaldo Extrait)			

(b) En utilisant le nombre total de mots communs comme critère de similarité, déterminez lequels des deux documents des joueurs sont le plus similaire. Le résultat est-il congruent ?

(c) En utilisant la similarité cosinus comme critère de similarité, quels sont les deux documents des joueurs les plus similaires? Le résultat est-il congruent ? Dans ce contexte, comparez et expliquez la différence entre le produit scalaire et la similarité cosinus.

3. (0.5 + 0.5 + 0.5 + 1 = 2.5 Marks)

Considérons le corpus suivant:

<s> I enjoy tennis </s>
<s> You enjoy soccer </s>
<s> I dislike soccer tournaments </s>
<s> You enjoy open tennis tournaments </s>

où <s> and </s> sont des tokens représentant le début et la fin d'une phrase, respectivement.

(a) Formez une matrice de fréquence/comptage de bigrammes pour le corpus ci-dessus en utilisant les tokens/jetons suivants dans le même ordre pour les lignes et les colonnes (excluez le token </s> comme ligne).

<s>	I	enjoy	tennis	you	soccer	dislike	tournaments	open	</s>

(b) Calculez toutes les probabilités de bigramme non nulles pour ce corpus. Présentez les résultats sous forme de tableau :

Bigram	Probabilities
P(I|<s>)	

(c) Soit $W = w_1, ..., w_n$ les mots du corpus de test et PP(W) la perplexité correspondante. Rappelons que $\text{PP}(W) = (\text{P}(W))^{-1/n} = 2^x$ où $x = -\frac{1}{n}\log_2\text{P}(W)$. Quelle est l'interprétation intuitive de PP(W) ? Comment PP(W) varie-t-elle par rapport à P(W) ?

(d) Sur la base des probabilités de bigrammes en (b), calculez la perplexité de **l'entièreté** du corpus test suivant.
<s> I enjoy tennis </s >
<s> You enjoy open tennis tournaments </s>
Important : <s> ne doit pas être pris en compte lors du calcul de la longueur de l'ensemble de test, alors que </s> doit l'être.

4. (1 Point)

Supposons que nous ayons le corpus suivant
<s> I am Marilyn </s>
<s> Marilyn I am </s>
<s> I do like going out for dinner tonight </s>

Estimez la probabilité de trigramme P(Marilyn | I am), en utilisant la méthode de lissage par interpolation linéaire simple avec des poids égaux λ.

5. Question Programmation (2.5 + 1.5 + 1 + 3 = 8 Points)

Ce problème nécessitera une programmation en Python 3. L'objectif est de construire et de tester un modèle de langage avec lissage.

Pour commencer, vous devriez charger et ouvrir le fichier fourni pour le devoir ***H23_Devoir-4_Question_Programmation.ipynb*** de préférence dans Google Colab (qui contient l'environnement nécessaire), à travers votre compte Google Drive. Téléchargez les fichiers ***brown-train.txt*** et ***brown-dev.txt*** (disponibles aussi sous Devoir 4 dans la Section Semaine 10 sur Moodle). Téléversez-les sur votre Google Drive, puis chargez-les dans votre notebook Colab en montant votre lecteur. Le code pour monter et charger est fourni pour vous dans le notebook, vous devez simplement modifier le chemin des fichiers si nécessaire.

Pour toutes les parties de codage, vous n'aurez pas besoin de créer de nouveaux fichiers ou notebooks. Le notebook a des sections où vous pouvez remplir le code pour tous les sous-problèmes. Recherchez le mot-clé "TODO" et remplissez votre code dans l'espace vide. Sentez-vous libre d'ajouter et de supprimer des arguments dans les signatures de fonctions, mais **faites attention** que vous pourriez avoir besoin de les changer dans les appels de fonction déjà présents dans le notebook.

Vous rapporterez vos réponses et tracés/figures aux questions posées ci-dessous dans le fichier PDF soumis.

(a) Dans le `H23_Devoir-4_Question_Programmation.ipynb`, remplissez les fonctions `nltkTokenize` et `countTopWords`. Complétez également le code sous "`Partie du code pour la sous-partie (a)`" et rapportez les 10 premiers mots ordonnés par leur fréquence dans le corpus d'entraînement, en utilisant à la fois `basicTokenize` et `nltkTokenize`. Quelles différences remarquez-vous entre les deux ? Une explication ?

(b) En utilisant la fonction `nltkTokenize` que vous aurez écrit, faites un graphique des fréquences des mots dans le corpus d'entrainement, ordonnés par leur rang, c'est-à-dire le mot **le plus fréquent** en premier, le deuxième mot le plus fréquent ensuite, et ainsi de suite sur l'axe des x. Ainsi, l'axe des x contiendra le rang des mots (c'est-à-dire que pour rendre la figure plus lisible, il ne faut pas afficher le mot mais son rang). Vous pouvez tracer uniquement les 500 mots les plus

fréquents pour voir la tendance plus clairement. Quelle tendance observez-vous dans votre graphique? (Veuillez saisir le code dans la cellule de la sous-section intitulée "`Code pour la sous-partie (b)`"). Incluez le graphique dans votre réponse.

(c) Utilisez la fonction `basicTokenize` et le modèle de langue bigramme ($n = 2$) sans lissage pour cette question. Entraînez le modèle de langage (sur l'ensemble de l'entraînement) et rapportez sa perplexité sur les ensembles d'entraînement et de validation. Que remarquez-vous ? (Veuillez remplir le code dans la cellule sous la sous-section intitulée "`Code pour la sous-partie (c)`").

(d) Utilisez la fonction `basicTokenize` et le modèle de langage bigramme ($n = 2$) avec lissage pour cette question. Implémentez le lissage de Laplace (add-α) dans la fonction appropriée fournie (`computeBigramAddAlpha` dans la classe `LanguageModel`) et entraînez le modèle avec le lissage add-alpha sur l'ensemble de l'entraînement pour différentes valeurs alpha en remplissant le code dans la cellule de la sous-section intitulée "`Code pour la sous-partie (d)`".
α prendra les valeurs suivantes $[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 1.5, 2]$. Tracez la perplexité sur les ensembles d'entrainement et de validation en fonction de alpha. Quelles différences observez-vous par rapport au cas précédent (pas de lissage) en (c) ? Décrivez le comportement de la perplexité de l'entrainement et de la validation lorsque nous faisons varier alpha. Est-il utile de faire varier alpha et de calculer la perplexité correspondante sur l'ensemble de validation ? Incluez votre graphique dans votre soumission.

Vous devez également soumettre le fichier " .ipynb" (avec les cellules déjà exécutées) et le fichier " .py", tous deux extraits de votre notebook Colab une fois terminé (allez dans Fichier → Télécharger → Télécharger .ipynb OU Télécharger .py).