Ayoub Echchahed
Marouane Maâou
Oussama Ameknassi

Natural Language Processing
IFT-7022

Assignment 1
Regular Expressions & N-gram Models & Text Classification

Work presented to
Mr. Luc Lamontagne

Faculty of Science and Engineering
Laval University
Fall 2022

TÂCHE #1 – EXPRESSIONS RÉGULIÈRES

Après avoir analyser les structures récurrentes qui vont nous aider à repérer les aliments et les quantités de chaque item, nous avons convergé vers l'utilisation de cette expression régulière:

```
r"\d*,?\d* (t\S*|c\S*? à (s\S*|c\S*)|ml |g |lb )?(.*\(.*\))?")"
```

Voici un résumé de l'utilité des sous-expressions présentes ci-dessus:

- **(\d*,?\d*)** : Chercher des nombres (Avec ou sans virgules)

- **(t\S*|c\S*? à (s\S*|c\S*)|ml |g |lb):** Chercher des mots commençant par la lettre t (pour tasse) ou c (pour cuillère). Pour le cas cuillère, les mots café et soupes ainsi que leurs abréviations (c. ,à s.) sont cherchés. D'autres cas sont aussi cherchés comme ml, g, lb tous suivis par des espaces afin d'enlever les cas des ingrédients commençant par chacune des expressions précédentes.

- **(.*\(.*\))?")**: Cette expression est utile dans les cas où les quantités sont décrites et expliquées par des unités contenues dans des parenthèses. Dans l'exemple "30 ml (2 c. à soupe) d'huile d'olive", cette expression va aller chercher la partie (2 c. à soupe)

A) Analyse des performances obtenues avec nos expressions régulières

La performance fût évaluée par la capacité de notre expression régulière à discerner les quantités/ingrédients de façon exacte sur les 128 cas présent dans le jeu de test.

Le score obtenu fût de 58 sur 128, mais cela en considérant une tolérance 0 sur des déviations possibles ne changeant pas le sens de la réponse. Par exemple, dans l'exemple suivant, les quantités obtenues sont exactes mais quant aux ingrédients, les chaines de caractères ne sont pas exactes mais vont dans le même sens, montrant ainsi que le score ne représente pas réellement la performance réelle de cette expression régulière.

QUANTITE: 3 c. à s	INGREDIENT: huile de sésame et arachide **(désiré)**
QUANTITE: 3 c. à s.	INGREDIENT: d'huile sésame et arachide **(obtenu)**

En ignorant ce problème, nous avons évalué qu'il est possible d'obtenir un score de 87 sur 128.

B) Discutez dans votre rapport des principales erreurs commises par vos expressions régulières.

En analysant notre expression, nous avons déduits que tous les mots commençant par t peuvent être problématique car ceux-ci peuvent être interprétés comme un qualificatif d'une quantité (représenté statistiquement par le mot tasse).

Également, une seconde erreur obtenue peut être observé dans l'exemple ci-dessous, où des mots additionnels sont obtenus dans certains cas lors de la recherche d'ingrédient.

QUANTITE: 16	INGREDIENT: petites palourdes **(désiré)**
QUANTITE: 16	INGREDIENT: petites palourdes dans leur coquille, rincées **(obtenu)**

Mais une expression régulière trop complexe risque d'être trop "collée" sur des données spécifiques, rendant la capacité à généraliser plutôt faible dans des cas divers.

TÂCHE 2 – MODÈLES DE LANGUE N-GRAMMES - COMME LE DISAIT LE PROVERBE…

A) Résultats obtenus

- Voici les résultats exprimés sous forme de proportions des tests réussis sur les 46 tests totaux:

Type du modèle	Logprob	Perplexité
Unigram	0.33	0.22
Bigram	0.61	0.54
Trigram	0.98	0.96

B) Les différents modèles capturent-ils bien le langage utilisé dans les proverbes ?

Il est possible d'affirmer que en effet, puisque nos modèles les plus performants qui sont les trigrams ont obtenus des perplexités faibles, ceux-ci ont bien capturé la structure statistique du langage utilisé dans les proverbes.

C) Quel est l'impact de la longueur de l'historique?

En augmentant la longueur du contexte prise en compte lors des prédictions, c'est à dire en permettant un conditionnement statistique sur plus de variables, la performance du modèle augmente rapidement. En théorie, cette performance atteindra un plateau éventuel avant de redescendre dû à la difficulté d'obtenir des distributions de probabilités conjointes tenant en compte un nombre de variables très haut.

D) Quelle est la différence entre les résultats obtenus avec le log-prob et la perplexité ? Expliquez.

Dans ce cas-ci, le log-prob nous offre de meilleurs résultats pour tous les types de modèles considérés, quoique chaque métrique possède son utilité. Un meilleur modèle de n-grammes est celui qui attribue une probabilité plus élevée aux données de test, et la perplexité est une version normalisée de la probabilité de l'ensemble de tests s'appliquant à l'entièreté des séquences.

TÂCHE 3 – CLASSIFICATION DE TEXTES – ANALYSE DE SENTIMENT

A) Quelles sont les performances obtenues avec chacun des classificateurs ? Notez-vous des différences significatives au niveau de la performance et de l'exécution du code?

	Naïve Bayes		
	Entrainement	**Test**	**Matrice de confusion**
Sans normalisation	0.9493	0.8115	$\begin{pmatrix} 769 & 217 \\ 154 & 828 \end{pmatrix}$
Normalisés avec stemming	0.9388	0.8069	$\begin{pmatrix} 758 & 228 \\ 152 & 830 \end{pmatrix}$
Normalisés avec lemmatisation	0.9428	0.8206	$\begin{pmatrix} 777 & 209 \\ 144 & 838 \end{pmatrix}$

	Régression Logistique		
	Entrainement	**Test**	**Matrice de confusion**
Sans normalisation	0.9992	0.8415	$\begin{pmatrix} 822 & 164 \\ 148 & 834 \end{pmatrix}$
Normalisés avec stemming	0.9977	0.8354	$\begin{pmatrix} 819 & 167 \\ 157 & 825 \end{pmatrix}$
Normalisés avec lemmatisation	0.9975	0.8440	$\begin{pmatrix} 828 & 158 \\ 149 & 833 \end{pmatrix}$

En termes d'exactitude, il est possible de constater que les modèles de régression logistique ont beaucoup plus de capacités à approximer les données d'entrainement, leurs permettant de ce fait d'obtenir d'excellents scores d'entrainement et des scores de test au-dessus des modèles Naïve Bayes, qui possèdent quant à eux moins de capacité mais une bonne capacité de généralisation aux tests. En analysant la différence entre les scores d'entrainement et de test pour toutes les configurations des modèles, il est possible d'affirmer qu'il existe un *overfitting*.

Finalement, les hyperparamètres utilisés dans toutes les configurations des modèles sont ceux par défauts de la librairie sklearn, sauf pour le nombre d'itérations qui fût fixé à 500 pour le modèle de régression logistique. Quant aux différences au niveau de la performance et de l'exécution du code, les modèles de régression logistique ainsi que les configurations utilisant la lemmatisation sont beaucoup plus exigeants lorsqu'il est question de ressources utilisées.

B) Est-ce que certains mots semblent jouer un rôle plus important ?

Il est évident que des mots à connotation positive ou négative sont ceux exprimant le plus d'information lorsqu'il est question de classification de sentiment binaire. Ainsi, ceux-ci posséderont des poids plus importants lors des prédictions faites par des modèles d'analyse de sentiments. Par exemple, pour des mots de la classe sentiment=1 (positif), des mots tels que *"great, love, good, ..."* posséderont certainement des poids plus importants lorsqu'il est question de faire une inférence.

C) Recommandez la configuration qui vous semble la plus intéressante.

En analysant les tableaux du dessus, quoique la simplicité de Naïve Bayes semble intéressante dans certains cas précis, nous pensons que la configuration la plus performante est celle de la **régression logistique normalisée via la lemmatisation.**

TÂCHE 4 – CLASSIFICATION DE TEXTES - IDENTIFICATION DE LANGUE

A) Évaluez la performance de chacun des modèles à l'entraînement & test tout en faisant varier la composition des N-grammes. Puis analyser les résultats

	Naïve Bayes	
	Entrainement	**Test**
Unigram	0.55 (+/- 0.12)	0.16
Bigram	0.69 (+/- 0.07)	0.4
Trigram	0.70 (+/- 0.05)	0.38
Multigram	0.71 (+/- 0.02)	0.36

	Régression logistique	
	Entrainement	**Test**
Unigram	0.62 (+/- 0.08)	0.26
Bigram	0.73 (+/- 0.05)	0.53
Trigram	0.73 (+/- 0.02)	0.52
Multigram	0.75 (+/- 0.04)	0.66

En analysant les résultats, il est donc évident que conditionner sur plus de caractères augmente la performance d'entrainement des deux modèles. Et comme mentionné au numéro 3, les modèles de régression logistiques performent de façon supérieure dû à leur meilleure capacité à approximer la distribution du jeu de données.

En raison de la capacité plus faible des modèles Naïve-Bayes, ceux-ci possèdent des performances qui atteignent un sommet dans la configuration des bigram tandis que les modèles de régression logistique possèdent des performances augmentant de façon constante à mesure que le nombre de caractères prise en compte lors du conditionnement statistique augmente. Bien-sûr, comme mentionner dans la question 2c, cette augmentation atteindra un plateau éventuel.

Cependant, à mesure que nous passons du bigram à des modèles de n-grammes plus élevés, la probabilité logarithmique moyenne diminue considérablement en grande partie dû au nombre élevé de n-grammes inconnus qui apparaissent dans les données du test mais pas dans les données d'entrainement.