Rapport mini-projet 3

Classification multilabel

Master 2 Data Science

Auteur :
Mohamed Dhmine

Encadrant :
M. Badih Ghattas

Année académique 2020–2021

Table des matières

1 Introduction

Le texte dans le text mining se réfère à la langue écrite qui a un certain contenu informationnel. Par exemple, les articles de journaux, les articles de magazines, les livres de fiction et de non-fiction, les courriels et les articles en ligne sont tous des textes. La quantité de texte qui existe aujourd'hui est vaste, et elle ne cesse de croître. Bien qu'il existe de nombreuses techniques et approches pour l'exploration de textes, l'objectif général est simple : découvrir des informations nouvelles et utiles contenues dans un ou plusieurs documents textuels. En pratique, l'exploration de textes est réalisée en exécutant des programmes informatiques qui lisent les documents et les traitent de différentes manières. Les résultats sont ensuite interprétés par humains. D'autre part, la classification multilabel est une tâche prédictive de data mining avec de multiples applications dans le monde réel, y compris l'étiquetage automatique de nombreuses ressources telles que des textes. L'apprentissage à partir de données multilabels peut être réalisé par différentes approches, telles que la transformation des données, l'adaptation des méthodes et l'utilisation d'ensembles de classificateurs. Le travail qui nous a été proposé l'a été par Monsieur Badih Ghattas. Le but étant d'essayer d'associer automatiquement des mots clès à des images à partir d'un texte et essayer enfin de faire un modèle qui prédit ces mots clès.

2 Base de données

Il nous a été donné un dossier **Images** contient 2327 images et un ficher **Fusion.csv** qui contient 5 colonnes :

- Une colonne **caption** qui contient une description chaque image ;
- Une colonne **image** qui contient le nom de l'image ;
- Une colonne **paragraph** qui contient une paragraphe générée ;
- Une colonne **page** qui donne le numéro de la page d'où chaque image est tirée
- Une colonne **structurel_check_DD** qui donne si la valeur est "1" indique que l'image en question est bien un schéma structural.

L'une des principales applications du traitement du langage naturel est d'extraire automatiquement les sujets dont les gens discutent à partir de gros volumes de texte. Certains exemples de texte volumineux peuvent être des flux de médias sociaux, des avis de clients sur des hôtels,des actualités, des e-mails de plaintes de clients, etc. Savoir de quoi les gens parlent et comprendre leurs problèmes et leurs opinions est très précieux pour les entreprises, les administrateurs et les campagnes politiques. Et il est vraiment difficile de lire manuellement des volumes aussi importants et de compiler les sujets. Il faut donc un algorithme automatisé capable de lire les documents texte et de sortir automatiquement les sujets abordés. On utilise dans ce projet LDA pour extraire les sujets.

3 Extraction des mots clès pertinents

3.1 Allocation de Dirichlet latente (LDA)

L'allocation de Dirichlet latente (latente Dirichlet allocation en anglais) est un modèle probabiliste qui permet d'identifier automatiquement des sujets présents dans un objet texte et d'extraire ainsi les motifs cachés. Il s'agit d'une approche non supervisée utilisée pour trouver et observer des groupes de mots (appelés «sujets ou concepts») dans un ensemble de documents de textes.

Le modèle LDA suppose que les documents sont produits à partir d'un mélange de sujets. Ces sujets génèrent ensuite des mots en fonction de leur distribution de probabilité. Étant donné un ensemble de données de documents, LDA tente de déterminer quels sujets créeraient ces documents.

Considérons une matrice (document-words DW) représentant le corpus (ensemble de document de textes) :

	W1	W2	⋯	Wn
D1	0	3	⋯	5
D2	1	4	⋯	2
D3	2	3	⋯	6
⋮	⋮	⋮	⋮	⋮
Dm	4	3	⋯	8

Cette matrice montre que le corpus est constitué de M documents répartis sur N vocabulaires (termes). La valeur de la cellule i,j donne le comptage de la fréquence du terme Wj dans le document Di.

La LDA consiste à convertir cette matrice en deux matrices binaires DT (document topic) et TW (topic words) respectivement :

	T1	T2	⋯	Wk
D1	0	1	⋯	1
D2	1	0	⋯	0
D3	1	1	⋯	1
⋮	⋮	⋮	⋮	⋮
Dm	1	0	⋯	0

	W1	W2	⋯	Wn
T1	0	0	⋯	1
T2	0	1	⋯	0
T3	1	0	⋯	1
⋮	⋮	⋮	⋮	⋮
Tk	1	1	⋯	0

La matrice DT représente la distribution des sujets ou concepts dans les documents et la matrice TW quand à elle représente la distribution des mots (vocabulaire) dans les sujets.

La LDA cherche à améliorer le modèle de sujet généré en calculant deux probabilités :

- $p_1 = \mathbb{P}(t/d)$ qui est la proportion de mots du document d affectéé au sujet t.
- $p_2 = \mathbb{P}(w/t)$ qui est la probabilité que le sujet t dans le corpus soit assigné au mot w.

On calcule alors le produit p_1 et p_2 qui correspond à la probabilité que le sujet t génère le mot w dans le document d.

Ce processus est répétés un grand nombre de fois jusqu'à ce que l'algorithme converge et on obtient le mélange de sujet présent dans chaque document en comptant chaque représentation d'un sujet (assigné aux mots du document). On obtient les mots associés à chaque sujet en comptant les mots qui y sont associés dans le corpus.

3.2 Prétraitement et nettoyage du texte

La modélisation de sujet textuels nécessite en grande partie une nettoyage en amont des données. C'est une étape cruciale pour obtenir des résultats satisfaisants et représentatifs. Ce prétraitement consiste entre autre à tokeniser les documents, c'est à dire découper chaque phrase du document en une liste de mots et supprimer les caractères inutiles tels que les ponctuations.

Un autre étape du prétraitement est aussi la normalisation des documents en passant par la lemmatisation, c'est à dire de ne considérer que les racines des mots. Par exemple on peut avoir les mots "drawing" et "drawings" en même temps dans le texte. Etant donné que ces mots sont proches, les

distinguer ne ferait qu'accroitre le surapprentissage et ne permettrait pas aux modèles d'exploiter pleinement les données d'apprentissage. Ensuite, il faut supprimer les mots vides, c'est à dire les mots qui apparaissent très fréquemment dans les documents notamment les articles définis et indéfinis.

Une fois ces étapes terminées, nous transformons les documents en une forme vectorisée en calculant la fréquence de chaque terme, constituant ainsi le corpus sur lequel nous allons entraîner le modèle LDA.

3.3 Mesure de cohérence des sujets

Afin de décider du nombre optimal de sujets à extraire à l'aide de LDA, on utilise le score de cohérence pour mesurer la qualité de l'extraction des sujets. Il est défini par :

$$CoherenceScore = \sum_{i<j} \text{score}(W_i, W_j)$$

La cohérence mesure la distance relative entre les mots dans un sujet. On distingue deux type de scores :

- Score UCI définit par :

$$\text{Score\_uci} = \log \frac{\mathbb{P}(w_i, w_j)}{\mathbb{P}(w_i)\mathbb{P}(w_j)}$$

où $\mathbb{P}(w_i)$ représente la probabilité de voir w_i dans un document aléatoire, et $\mathbb{P}(w_i, w_j)$ la probabilité de voir les deux w_i et w_j se produisant dans un document aléatoire.

- Score UMass définit par :

$$\text{Score\_UMass} = \log \frac{\mathbf{D}(w_i, w_j) + 1}{\mathbf{D}(w_i)}$$

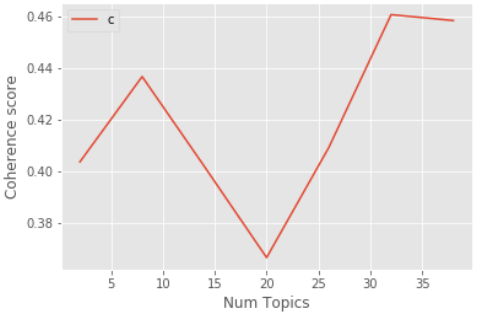
où le terme $\mathbf{D}(w_i, w_j)$ du numérateur est le nombre de documents dans lesquels les mots w_i et w_j apparaissent ensemble. On ajoute 1 à ce terme puisque nous prenons le logarithme et nous devons éviter de prendre un logarithme de 0 lorsque les deux mots n'apparaissent jamais ensemble. Le dénominateur $\mathbf{D}(w_i)$ est le nombre de documents $\mathbf{D}$ où apparaît le terme w_i.

Le score est donc plus élevé si w_i et w_j apparaissent beaucoup ensemble dans les documents par rapport à la fréquence que w_i apparaît seul dans les documents. Cela a du sens en tant que mesure de la cohérence du sujet, car si deux mots d'un sujet vont vraiment ensemble, on s'attend à ce qu'ils apparaissent beaucoup ensemble. Le dénominateur ajuste simplement la fréquence des mots dans les documents, de sorte que des mots comme les articles (le, la , $\cdots$) n'obtiennent pas un score artificiellement élevé.

3.4 Résultats

3.4.1 Nombre de sujets

Grace au score de cohérence nous pouvons définir le nombre de nécessiare pour construire le modèle.



Num Topics = 2 has Coherence Value of 0.4
Num Topics = 8 has Coherence Value of 0.44
Num Topics = 14 has Coherence Value of 0.4
Num Topics = 20 has Coherence Value of 0.37
Num Topics = 26 has Coherence Value of 0.41
Num Topics = 32 has Coherence Value of 0.46
Num Topics = 38 has Coherence Value of 0.46

FIGURE 1 – Score de cohérence pour obtenir le nombre de sujet nécessaire.

Ici, le modèle nous suggère de prendre 32 sujets puis qu'on obtient un score maximal lorsque le nombre de sujet est 32. Comme y'a pas grande différence entre le score en 8 et 32 on a choisit 8. Aprés avoir eu les sujets optimaux, on cherche maintenant les mots clés pour chaque sujets qui sont dans le tableau suivant.

p.png

FIGURE 2 – Une partie du tableau des mots clés.

On a maintenant les mots clés mais plusieurs mots peuvent représenter plusieurs sujets. Maintenant les mots clés, On commence la partie classification multilabel.

4 Classification multi-label

La classification multi-label est une extension de la classification traditionnelle dans laquelle les classes ne sont pas mutuellement exclusives, chaque individu pouvant appartenir à plusieurs classes simultanément. Ce type de classification est requis par un grand nombre d' applications actuelles telles que la classification d'images. L'ensemble de données que nous utiliserons dans cette partie de classification multi-label se compose de 2200 images. Chaque image a sa propre label. Les labels sont définis à partir des données textuelles, et sont extraits au dessus. Pour ce faire, on va utiliser un réseaux de neurones convolutionnels qui a pour objectif de prédire les mots clés des images. L'architecture du modèle qu'on a implémenté est la suivante :

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 196, 196, 16)	1216
batch_normalization_1 (Batch	(None, 196, 196, 16)	64
max_pooling2d_1 (MaxPooling2	(None, 98, 98, 16)	0
dropout_1 (Dropout)	(None, 98, 98, 16)	0
conv2d_2 (Conv2D)	(None, 94, 94, 32)	12832
max_pooling2d_2 (MaxPooling2	(None, 47, 47, 32)	0
batch_normalization_2 (Batch	(None, 47, 47, 32)	128
dropout_2 (Dropout)	(None, 47, 47, 32)	0
conv2d_3 (Conv2D)	(None, 43, 43, 64)	51264
max_pooling2d_3 (MaxPooling2	(None, 21, 21, 64)	0
batch_normalization_3 (Batch	(None, 21, 21, 64)	256
dropout_3 (Dropout)	(None, 21, 21, 64)	0
conv2d_4 (Conv2D)	(None, 17, 17, 64)	102464
max_pooling2d_4 (MaxPooling2	(None, 8, 8, 64)	0
batch_normalization_4 (Batch	(None, 8, 8, 64)	256
dropout_4 (Dropout)	(None, 8, 8, 64)	0
flatten_1 (Flatten)	(None, 4096)	0
dense_1 (Dense)	(None, 128)	524416
dropout_5 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 78)	10062

FIGURE 3 – Architecture CNN du modèle .

Le réseau a plusieurs couches de convolution comme on a des images. (3 couches). il a aussi plusieurs couches de batch-normalisation qui ont pour utilité de rendre le réseau de neurones plus rapides grâce à la normalisation des couches d'entrées. De plus, On a utilisé le max-pooling qui calcule le maximum ou le plus large valeur de chaque patch. Sa fonction est de reduire progressivement la taille spatiale de la représentation pou reduire la quantité et de calcul dans le réseau. On a utilisé la fonction entropie croisée comme fonction de perte.

4.1 L'entropie croisée et l'entropie croisée binaire

L'entropie d'une distribution de probabilité discrète est définie comme étant :

$$H(\xi) = -\sum_{k} p_k \log p_k$$

L'entropie croisée

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

l'entropie croisée binaire :

$$CE = -\sum_{i=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) - (1-t_1) log(1-f(s_1)), \qquad f(s_i) = \frac{1}{1+e^{-s_i}}$$

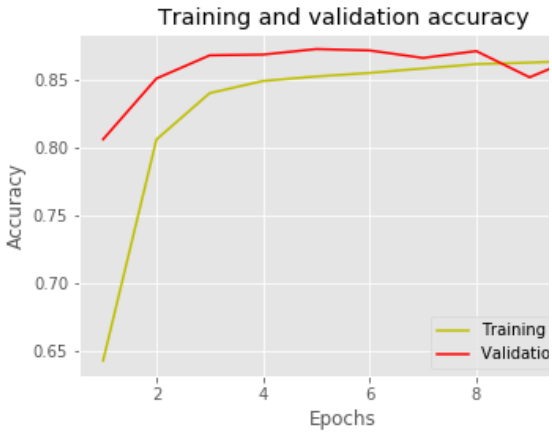
Il est utilisé pour la classification multi-labels, où l'intuition d'un élément appartenant à une certaine classe ne doit pas influencer la décision pour une autre classe car, contrairement à la perte Softmax, elle est indépendante pour chaque composante vectorielle (classe), ce qui signifie que la perte calculée pour chaque composante vectorielle de sortie CNN n'est pas affectée par les valeurs des autres composantes. En effet, Elle marche qu'avec la fonction d'activation Sigmoid.

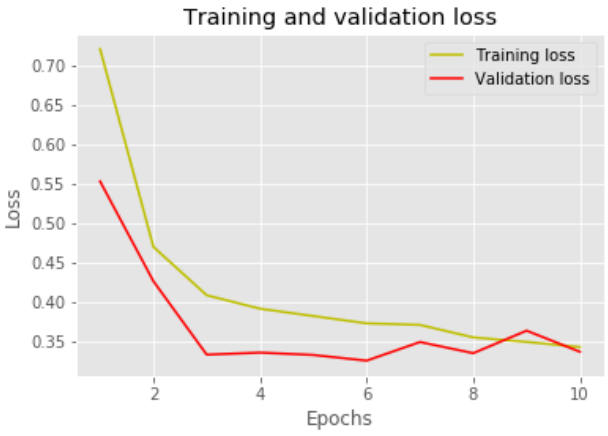
4.2 Résultats

Nous avons formé le réseau pendant 10 itérations, le nombre total d'époques pour lesquelles nous formerons notre réseau (c'est-à-dire combien de fois notre réseau «voit» chaque exemple de formation et en apprend des modèles), réalisant :

— Exactitude de la classification de 86,60% sur l'ensemble d'entraînement ;

— 86,87% de précision de classification sur l'ensemble de test.





5 Conclusion

L'extraction de mots-clés (également appelée analyse des mots-clés) est une méthode d'analyse d'un texte qui consiste à extraire automatiquement les mots et les termes les plus importants. Cela permet de résumer le contenu d'un texte et d'identifier les principaux sujets qui sont abordés. Aussi, contrairement aux tâches de classification normales où les labels de classe sont mutuellement exclusives, la classification multi-labels nécessite des algorithmes d'apprentissage automatique spécialisés qui prennent en charge la prédiction de plusieurs classes ou «étiquettes» mutuellement non exclusives. Finalement, nous avons effectué notre tâche de classification multilabels en la transformant en de multiples classifications binaires. Ce mini projet nous permis d'aborder plusieurs tâches comme le réseau de neurones ainsi que la LDA. On peut essayer avec d'autres fonctions de perte comme kullback leibler pour obtenir des meillueres résultats.

Références

[1] https ://machinelearningmastery.com/multi-label-classification-with-deep-learning/