Prof. Amal Zouaq – INF8460 (Automne 2020)
Chargés de laboratoire : Dylan Farvacque et Antoine Lefebvre-Brossard

École Polytechnique de Montréal

Département Génie Informatique et Génie Logiciel

INF8460 – Traitement automatique de la langue naturelle

TP3

Objectifs d'apprentissage

- Utiliser des plongements lexicaux pré-entrainés pour de la classification
- Entrainer des plongements lexicaux de type word2vec
- Implanter des modèles de classification neuronaux

Ressources:

Entrainement avec gensim :

https://www.tensorflow.org/tutorials/keras/text_classification

https://radimrehurek.com/gensim/models/word2vec.html

Logiciels

- Gensims
- Scikit-learn
- Tensorflow
- ipython Notebook

Pour ce TP, il est **fortement recommandé** d'utiliser une machine avec un GPU. Vous pouvez par exemple utiliser les machines du lab L-4818 ou travailler sur Google Colab.

Modalités de remise du TP

Vous devez compléter le squelette `inf8460_tp3.ipynb` et le soumettre sous le nom `matricule1_matricule2_matricule3_TP3.ipynb` qui reprend les différentes questions, et implante les fonctionnalités requises.

Vous devez retourner le notebook complété. Assurez-vous que le notebook est roulé au complet avec toutes les réponses demandées affichées.

Critères d'évaluation

- L'exécution correcte du code et sa qualité
- Le professionnalisme du notebook

- La clarté des explications et commentaires qui l'accompagnent

Corpus et code

Le zip du TP contient :

- Le corpus de revues de films IMDB (http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz)
- Le squelette du notebook qui doit être complété

Travail à faire

Dans ce TP, on entraine des plongements lexicaux sur le corpus et on effectue une classification automatique avec ces plongements.

1. Entrainement de plongements lexicaux

Vous devez réaliser les étapes suivantes:

a) Utiliser Gensim pour entrainer un modèle word2vec sur le corpus.
b) Décrire les paramètres du ou des modèles entraînés, le nombre de mots encodés et le temps d'entraînement. Décrire le cas échéant et de manière précise tout problème que vous avez eu à obtenir votre modèle et les façons de résoudre ces problèmes.
c) Retrouver les 5 mots voisins des mots suivants : excellent, terrible

2. Classification avec des plongements lexicaux

On vous demande d'effectuer de la classification avec les plongements lexicaux obtenus.

a) En reprenant le code développé dans le TP1 avec Scikitlearn, on vous demande cette fois de tester un modèle Naïve Bayes et de régression logistique avec des n-grammes (n=1,2,3 ensemble). Essayez de voir si une réduction de dimension améliore la classification. Ne fournissez que votre meilleur modèle. Evaluez vos algorithmes selon les métriques d'accuracy générale et de F1 par classe sur l'ensemble de test.
b) En utilisant Tensorflow (ou Pytorch), on vous demande de développer un classificateur perceptron multicouches et un bi-LSTM avec les vecteurs d'un modèle word2vec pré-entrainé sur Wikipédia en Anglais (enwiki_upos_skipgram_300_3_2019) disponible à http://vectors.nlpl.eu/repository/11/3.zip.

On s'attend à ce que vous effectuiez une moyenne des vecteurs de mots de chaque document pour obtenir un plongement du document.

Evaluez vos algorithmes selon les métriques d'accuracy générale et de F1 par classe sur l'ensemble de test. Pour chacun des modèles, indiquez ses performances et ses spécifications (nombre d'époques, régularisation, optimiseur, nombre de couches, etc.). N'hésitez pas à expérimenter avec différents paramètres. Vous ne devez reporter que votre meilleure expérimentation.

c) Ré-entrainez les modèles en b) avec vos propres vecteurs obtenus en 1) a). Comparez maintenant la performance obtenue en 2) b) avec celle que vous obtenez en utilisant vos propres vecteurs de mots entrainés sur le corpus.

d) Générez une table ou un graphique qui regroupe les performances des modèles, leurs spécifications, la durée d'entraînement et commentez ces résultats.
Quelle est l'influence des *word embeddings* sur les performances?
Quel est votre meilleur modèle ?