HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Cours TAL – Labo 5 : Applications du modèle word2vec

Objectif

Comparer des modèles *word2vec* pré-entraînés sur l'anglais avec des modèles appris localement sur deux corpus, en les appliquant à des tâches de mesures de similarité et d'analogie entre mots.

Consignes

- Soumettre sur Cyberlearn un notebook Jupyter avec les expériences, les résultats obtenus et leur analyse. Bien présenter les étapes suivies, et répondre clairement aux questions posées dans l'énoncé. Bien préciser les données et les commandes utilisées.
- Le travail est à effectuer en binôme.
- Ne pas hésiter à consulter la documentation de Gensim sur word2vec, ainsi que celle sur les KeyedVectors, qui forment une classe plus générale avec plusieurs exemples intéressants.
- Les différentes tâches se feront soit sur votre propre ordinateur (si vous disposez d'au moins 16 Go de RAM), soit sur un notebook fourni par le service Google Colab.

1. Tester et évaluer un modèle entraîné sur Google News

a. Installez **gensim**, la bibliothèque implémentant les outils pour travailler avec Word2Vec (`pip install --upgrade gensim`). Puis chargez le modèle word2vec pré-entraîné sur le corpus Google News en écrivant : w2v_model = `gensim.downloader.load("word2vec-google-news-300")`, ce qui télécharge le fichier la première fois, et enfin en ne gardant que les vecteurs de mots, avec « `w2v_vectors = w2v_model.wv` » puis « `del w2v_model` »).
Si on dispose du fichier en local, on peut le charger en écrivant `w2v_vectors = KeyedVectors.load_word2vec_format(path_to_file, binary=True)`. Quelle place mémoire occupe le processus du notebook une fois les vecteurs de mots chargés ?

b. Quelle est la dimension de l'espace vectoriel dans lequel les mots sont représentés ? Et quelle est la taille du vocabulaire du modèle ? Identifiez cinq mots qui sont dans le vocabulaire et un qui ne l'est pas.

c. Comment peut-on mesurer la distance entre deux mots dans cet espace ? Calculez par exemple la distance entre les mots *rabbit* et *carrot*.

d. Testez le modèle de distance entre mots. Est-ce que des mots proches sémantiquement sont aussi proches dans l'espace vectoriel, et inversement ? Testez au moins cinq paires de mots.

e. Y a-t-il des cas ambigus, et pourquoi selon vous ? Par exemple, pouvez-vous trouver des mots opposés selon le sens qui sont proches dans l'espace réduit ?

f. Que dire des mots ayant plusieurs sens ? Pouvez-vous donner 3 exemples de ce problème ?

g. En vous aidant de la documentation de Gensim sur `KeyedVectors`, mesurez de manière quantitative la performance du modèle sur le corpus `WordSimilarity-353`. Expliquez ce que signifient vos résultats.

h. De même, en vous inspirant de la documentation, évaluez le modèle sur les données de test appelées `questions-words.txt`. Pouvez-vous expliquer ce que mesure ce test ? Les résultats du modèle sont-ils satisfaisants ? Commentez.

2. Entraîner et tester deux nouveaux modèles à partir de corpus

a. Afin de pouvoir télécharger un corpus textuel par Gensim, exécutez la commande : `import gensim.downloader`. Puis récupérez le corpus des 10^8 premiers caractères de Wikipédia (en anglais) avec la commande : `corpus = gensim.downloader.load('text8')`. Combien de phrases et de mots possède ce corpus ?

b. Entraînez un nouveau modèle word2vec sur ce nouveau corpus. (Procédez éventuellement progressivement, en commençant par 1%, puis 10% du corpus, pour contrôler le temps.) Combien de temps prend l'entraînement et quelle est la taille (en Mo) du modèle word2vec résultant ?

c. Quelle dimension avez-vous choisi pour le plongement (*embedding*) de ce nouveau modèle ?

d. Testez quantitativement la qualité de ce nouveau modèle comme dans la partie 1 (points h et i). Ce modèle est-il meilleur que celui entraîné sur Google News ? Quelle serait la raison de la différence ?

e. Considérez maintenant le corpus quatre fois plus grand constitué de la concaténation du corpus *text8* et des dépêches économiques de Reuters (corpus de 413 Mo que vous trouverez en ligne comme `wikipedia_augmented.dat`). Utilisez ce nouveau corpus afin d'entraîner un nouveau modèle word2vec. Vous devez pour cela utiliser la fonction `Text8Corpus()` afin de charger et pré-traiter le texte du corpus (tokenization et segmentation en phrases, qui étaient déjà faites par la fonction `load()` du *downloader*). Combien de temps prend l'entraînement et quelle est la taille (en Mo) du modèle word2vec résultant ? Quelle est la dimension de l'*embedding* ?

f. Testez aussi ce modèle. Est-il meilleur que le précédent ? Pour quelle raison ?

g. Créez un nouveau fichier de test en augmentant `questions-words.txt` avec une catégorie de mots de votre choix (environ 20 items de test). Par exemple, à partir de {(*eye*, *see*), (*ear*, *listen*), (*foot*, *walk*)} on peut construire 6 items de test. Testez les trois modèles sur ces données, présentez et commentez vos résultats.