Cours TAL – Labo 4 : Le modèle word2vec et ses applications

Objectif

Comparer des modèles *word2vec* pré-entraînés avec des modèles que vous entraînerez sur deux corpus de tailles différentes. L'évaluation se fera sur des tâches de mesures de similarité et d'analogie entre mots, en anglais.

Consignes

- Veuillez suivre les étapes indiquées ci-après, en écrivant votre code, vos résultats et vos réponses aux questions dans un notebook Jupyter (à soumettre à la fin sur Cyberlearn).
- Consultez avec attemtion la documentation de Gensim sur word2vec, ainsi que celle sur les KeyedVectors (une classe plus générale) qui fournit des exemples utiles.
- Les différentes tâches se feront soit sur votre propre ordinateur (si possible avec 16 Go de RAM au moins), soit sur un notebook exécuté par le service en ligne Google Colab.

1. Tester et évaluer un modèle entraîné sur Google News

a. Installez `gensim`, une librairie Python qui fournit des outils pour travailler avec Word2Vec (avec `conda` ou avec `pip`). ***Prenez la version 3.8.3, et non pas la nouvelle version 4.0.X***. Obtenez depuis `gensim` le modèle `word2vec` pré-entraîné sur le corpus Google News en écrivant : `w2v_model = gensim.downloader.load("word2vec-google-news-300")`, ce qui téléchargera le fichier la première fois. Ne gardez en mémoire que les vecteurs des mots, en écrivant : « `w2v_vectors = w2v_model.wv` » puis « `del w2v_model` ».
    - Une fois que vous avez téléchargé le modèle, vous pouvez utiliser votre copie locale : `w2v_vectors = KeyedVectors.load_word2vec_format(path_to_file, binary=True)`.

b. Quelle place mémoire occupe le processus du notebook une fois les vecteurs de mots chargés ?

c. Quelle est la dimension de l'espace vectoriel dans lequel les mots sont représentés ?

d. Quelle est la taille du vocabulaire du modèle ? Affichez cinq mots (anglais) qui sont dans le vocabulaire et deux qui ne le sont pas.

e. Quelle est la distance entre les mots *rabbit* et *carrot* ? Veuillez aussi expliquer en une phrase comment on mesurer les distances entre deux mots dans cet espace.

f. Considérez au moins 5 paires de mots, certains proches par leurs sens, d'autres plus éloignés. Pour chaque paire, calculez la distance entre les deux mots. Veuillez indiquer si les distances obtenues correspondent à vos intuitions sur la proximité des sens des mots.

g. Pouvez-vous trouver des mots de sens opposés mais qui sont proches dans l'espace vectoriel ? Comment expliquez vous cela ? Est-ce une qualité ou un défaut du modèle word2vec ?

h. En vous aidant de la documentation de Gensim sur `KeyedVectors`, calculez le score du modèle word2vec sur les données **`WordSimilarity-353`**. (La doc vous permettra aussi de récupérer le fichier.) Expliquez en 1-2 phrases comment ce score est calculé et ce qu'il mesure.

i. En vous aidant de la documentation, calculez le score du modèle word2vec sur les données **`questions-words.txt`**. *Attention, cette évaluation prend une dizaine de minutes.* Expliquez en 1-2 phrases comment ce score est calculé et ce qu'il mesure.

2. Entraîner deux nouveaux modèles word2vec à partir de nouveaux corpus

a. En utilisant `gensim.downloader`, récupérez le corpus qui contient les 10^8 premiers caractères de Wikipédia (en anglais) avec la commande : `corpus = gensim.downloader.load('text8')`. Combien de phrases et de mots (*tokens*) possède ce corpus ?

b. Entraînez un nouveau modèle word2vec sur ce nouveau corpus. Si nécessaire, procédez progressivement, en commençant par 1% du corpus, puis 10%, pour contrôler le temps nécessaire.

- Indiquez la dimension choisie pour le *embedding* de ce nouveau modèle.
- Combien de temps prend l'entraînement sur le corpus total ?
- Quelle est la taille (en Mo) du modèle word2vec résultant ?

c. Mesurez la qualité de ce modèle comme dans la partie 1, points i et j. Ce modèle est-il meilleur que celui entraîné sur Google News ? Quelle serait la raison de la différence ?

d. Téléchargez maintenant le corpus quatre fois plus grand constitué de la concaténation du corpus *text8* et des dépêches économiques de Reuters (413 Mo) fourni en ligne par l'enseignant et appelé `wikipedia_augmented.dat`. Entraînez un nouveau modèle word2vec sur ce corpus, en précisant la dimension du plongement (*embedding*).

- Utilisez la classe `Text8Corpus()` pour charger le corpus et faire la tokenization et la segmentation en phrases.
- Combien de temps prend l'entraînement ?
- Quelle est la taille (en Mo) du modèle word2vec résultant ?

e. Testez ce modèle comme en 1.h et 1.i. Est-il meilleur que le précédent ? Pour quelle raison ?