# Cours TAL – Labo 5 : Le modèle word2vec et ses applications

## Objectif

Le but de ce travail est de comparer un modèle *word2vec* pré-entraîné avec deux modèles que vous entraînerez vous-mêmes sur des jeux de tailles différentes. La comparaison se fera sur une tâche de similarité de mots et sur une tâche de raisonnement par analogie, les deux en anglais. Vous utiliserez la librairie Python Gensim qui offre des fonctions pour manipuler des vecteurs de mots.

## Consignes

- Veuillez suivre les étapes indiquées ci-après, en écrivant votre code, vos résultats et vos réponses aux questions dans un notebook Jupyter, que vous soumettrez à la fin sur Cyberlearn.
- Il sera utile de bien étudier la documentation de Gensim sur les KeyedVectors, notamment *What can I do with word vectors?*, le tutoriel sur Word2Vec, ainsi que les pages sur word2vec et sur la classe KeyedVectors qui représente des vecteurs de mots génériques.
- Les différentes tâches se feront soit sur votre propre ordinateur (si possible avec au moins 8 Go de RAM), soit sur le service en ligne Google Colab.

## Installation de Gensim

Veuillez installer `gensim` avec `conda` ou avec `pip`. Le travail a été testé avec la version 4.3.3 et la librairie `scipy` version 1.15. (Note : la version 4.3.2 demandait la version 1.12 de `scipy`. En cas de problème, vérifier que la variable système `Path` contient `C:\ProgramData\Miniconda3\Library\` et `C:\ProgramData\Miniconda3\Library\bin\`.)

## 1. Tester et évaluer un modèle déjà entraîné sur Google News

Veuillez télécharger le modèle `word2vec` pré-entraîné sur le corpus Google News en écrivant :

```
from gensim import downloader as api
w2v_vectors = api.load("word2vec-google-news-300")
```

ce qui téléchargera le fichier la première fois.

Après avoir téléchargé le modèle, vous pourrez l'utiliser ainsi (dans le dossier `gensim-data`) :

```
from gensim.models import KeyedVectors
w2v_vectors = KeyedVectors.load_word2vec_format(path_to_file, binary=True).
```

a. Quelle place en mémoire occupe le processus du notebook avec les vecteurs de mots ?

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

c. Quelle est la taille du vocabulaire connu du modèle ? Veuillez afficher cinq mots anglais qui sont dans le vocabulaire et deux qui ne le sont pas.

d. Quelle est la similarité entre les mots *rabbit* et *carrot* ? Veuillez rappeler comment on mesure les similarités entre deux mots grâce à leurs vecteurs.

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

f. Pouvez-vous trouver des mots de sens opposés mais qui sont proches selon le modèle ? Comment expliquez-vous cela ? Est-ce une qualité ou un défaut du modèle word2vec ?

g. En vous aidant de la documentation de Gensim sur `KeyedVectors`, obtenez les scores du modèle word2vec sur les données de test **`WordSimilarity-353`**. Veuillez rappeler en 1-2 phrases comment les différents scores sont calculés.

h. 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, donc il vaut mieux commencer par tester avec un fragment de ce fichier (copier/coller les 100 premières analogies).* Expliquez en 1-2 phrases comment ce score est calculé et ce qu'il mesure.

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

a. En utilisant `gensim.downloader` (voir question 1) récupérez le corpus qui contient les $10^8$ premiers caractères de Wikipédia (en anglais) avec la commande : `corpus = api.load('text8')`. Combien de phrases et de mots (*tokens*) possède ce corpus ?

b. Entraînez un nouveau modèle word2vec sur ce nouveau corpus (voir la documentation de Word2vec). Si nécessaire, procédez progressivement, en commençant par utiliser 1% du corpus, puis 10%, etc., pour contrôler le temps nécessaire.

- Veuillez indiquer 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 en (1g) et (1h). Ce modèle est-il meilleur que celui entraîné sur Google News ? Quelle est selon vous 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 fourni par l'enseignant et appelé `wikipedia_augmented.zip` (à décompresser en un fichier '.dat' de 413 Mo). Entraînez un nouveau modèle word2vec sur ce corpus, en précisant la dimension choisie pour les *embeddings*.

- Utilisez la classe `Text8Corpus()` pour charger ce corpus, ce qui fera automatiquement la tokenisation 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 (1g) et (1h). Est-il meilleur que le précédent ?