TD2 - Introduction au TAL
Modèle Word2vec et Visualisation d'embeddings

1. Introduction

Ce TP porte sur la mise en place d'un modèle de représentation des mots plus avancé que l'approche TF-IDF : nous utiliserons ici l'approche par réseaux de neurones word2vec. Nous chercherons dans un premier temps à comparer les vecteur et ensuite à créer notre propre modèle de représentation des mots par word embeddings au moyen de la librairie Gensim. Vous aurez besoin de la librairie *word2vec* (https://radimrehurek.com/gensim/models/word2vec.html) et également de *simple_process* pour réaliser le retraitement de chaque ligne (https://tedboy.github.io/nlps/generated/generated/gensim.utils.simple_preprocess.html).

Nous chercherons ensuite à visualiser les embeddings sur deux dimensions (réduction de dimension pour pouvoir les « interpréter »). Nous utiliserons l'algorithme t-SNE (https://fr.wikipedia.org/wiki/Algorithme_t-SNE). Il faudra alors utiliser la librairie TSNE de sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html). Il faudra également utiliser la librairie pat de pyplot pour visualiser les graphiques (https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html).

2. Recherche des mots proches et opérations arithmétiques

Modèle. Afin de nous intéresser aux embeddings et à leur comparaison, nous allons dans un premier temps récupérer un modèle pré-entrainé en français. Nous allons par exemple récupérer ce modèle : https://huggingface.co/Word2vec/wikipedia2vec_frwiki_20180420_300d

N'hésitez pas à aller voir la librairie KeyedVectors de Gensim : https://radimrehurek.com/gensim_3.8.3/models/keyedvectors.html

1. Vous avez normalement des informations sur comment l'utiliser. Récupérez et chargez ensuite ce modèle. Réalisez ensuite plusieurs tests avec différents mots pour voir les mots similaires. Essayez notamment d'utiliser des majuscules/minuscules, des noms courants, des noms plus rares…

2. Vous pouvez ensuite découvrir la puissance des embeddings au travers des opérateurs arithmétiques. Essayez par exemple : model.most_similar(model.get_vector("roi")-model.get_vector("homme")+model.get_vector("femme"))

3. Création d'un modèle word2vec

Données. Nous allons maintenant créer notre propre modèle word2vec. Pour cela, nous avons besoin de données textuelles. Voici un exemple de jeu de données que vous pouvez utiliser : http://redac.univ-tlse2.fr/corpus/wikipedia.html

Vous pouvez notamment ici utiliser les librairies Word2Vec et simple_preprocess de Gensim (voir introduction). Vous allez devoir :

1. Ouvrir le fichier de données textuelles.

2. Prétraiter les données avec simple_preprocess.

3. Entrainer le modèle word2vec avec la fonction Word2Vec().

4. Sauvegarder le modèle.

5. Regarder que tout a bien fonctionné en réutilisant la visualisation des vecteurs comme dans la partie 2 Recherche des mots proches.

3. Visualisation des vecteurs avec tSNE et matplotlib

Modèles. Vous pouvez utiliser ici le premier modèle pré-entrainé récupéré ou celui que vous avez entrainé de votre côté.

Concrètement, deux étapes vont être réalisées : une réduction de dimension (ici, on obtient deux dimensions pour une visualisation des points en 2D) puis un affichage de ces points.

1. Afin de pouvoir avoir quelque chose de visualisable, nous n'allons pas mettre tous les mots disponibles sur le graphique (à vous de choisir le nombre). Vous pouvez utiliser le code suivant ou refaire de votre côté une liste de mots à afficher. Ici, nous cherchons les n mots les plus fréquents :

```
mots_outils = ["le", "la", "les", "de", "du", "des", "en", "et", "à", "au", "aux", "un", "une", "ce", "cet", "cette", "ces", "qui", "que", "quoi", "qu", "dont", "où", "si", "s", "ne", "pas", "plus", "moins", "mais", "ou", "et", "donc", "or", "ni", "car", "aussi", "après", "avant", "pendant", "depuis", "alors", "lorsque", "lors", "après", "avant", "enfin", "puis", "ensuite", "aujourd'hui", "demain", "hier", "maintenant", "parfois", "quelquefois", "souvent", "jamais", "toujours", "peu", "beaucoup", "trop", "assez", "plusieurs", "tout", "rien", "personne", "autre", "même", "chacun", "chacune", "ensemble", "seul", "seule", "autre", "autres", "chaque", "plusieurs", "quelques", "tant", "si", "tellement", "tel", "telle", "telles", "tels", "toute", "tous", "tout", "rien", "personne", "quelqu'un", "quelqu'une"]

import numpy as np

from sklearn.manifold import TSNE

import matplotlib.pyplot as plt

from gensim.models import Word2Vec

from collections import Counter

# Chargement du modèle

model = Word2Vec.load("modele_word2vec.model")

# Fréquence des mots

word_freq = Counter(model.wv.index_to_key)

top_100_words = [word for word, freq in word_freq.most_common(500) if word not in mots_outils]

# Extraire les vecteurs d'embedding des 100 mots les plus fréquents
```

vectors = [model.wv[word] for word in top_100_words]

2. Vous allez maintenant réaliser la réduction de dimension de cette liste. Je vous invite à regarder la librairie TSNE.

3. Positionnez ensuite les points avec matplotlib.