Rapport TP Word2vec avec gensim

— Qian DU

1. Introduction

Dans la première partie du cours *Fouille de données textuelles*, nous avons étudié les représentations vectorielles des mots ainsi que les principes fondamentaux du traitement automatique du langage naturel (TAL).

L'objectif de ce projet est d'entraîner un modèle Word2Vec sur deux corpus distincts — les descriptions de films (Movies Overview) et les avis clients sur des téléphones portables (Cellphone Reviews) — afin d'analyser la qualité des représentations vectorielles apprises et d'évaluer la cohérence sémantique des embeddings générés.

2. Description du modèle Word2Vec

2.1 Principes théoriques

Word2Vec est un modèle d'apprentissage non supervisé développé en 2013. Il transforme chaque mot en un vecteur numérique dense dans un espace vectoriel. Les mots apparaissant dans des contextes similaires obtiennent des vecteurs proches. La similarité entre deux mots est généralement mesurée à l'aide de la distance cosinus.

Il existe deux architectures principales:

- CBOW (Continuous Bag of Words) : prédire un mot cible à partir de son contexte
- Skip-gram : prédire les mots du contexte à partir d'un mot cible

Dans ce projet, nous utilisons le modèle Skip-gram. Le modèle repose sur un réseau de neurones simple avec une couche cachée. L'optimisation se fait par descente de gradient stochastique.

L'entraînement consiste à maximiser la probabilité des mots contextuels étant donné un mot cible, ce qui correspond à une estimation par maximum de vraisemblance (MLE). Minimiser la divergence de Kullback-Leibler entre la distribution réelle et la distribution prédite revient à maximiser la log-vraisemblance.

2.2 Implémentation et paramètres choisis

Le modèle a été implémenté en Python avec la bibliothèque Gensim.

Les principaux paramètres utilisés sont :

- **vector_size** = 100 : dimension des vecteurs
- **window** = 5 : taille de la fenêtre de contexte
- **min_count** = 5 : suppression des mots rares

- **sg = 1** : utilisation du modèle Skip-gram
- **epochs** = 20 : nombre de passages sur les données

Le choix de ces paramètres représente un compromis entre qualité sémantique et coût computationnel.

2.3 Résultats attendus

Nous attendons que :

- Les mots apparaissant dans des contextes similaires soient proches dans l'espace vectoriel ;
- Les relations sémantiques soient partiellement capturées ;
- Les résultats varient selon le corpus utilisé.

Par exemple :

- Dans le corpus Movies, le mot "war" devrait être associé à des termes historiques
- Dans le corpus Cellphone, le mot "phone" devrait être associé à des composants techniques.

3. Mise en pratique et analyse des performances

3.1 Description des données

Deux datasel ont été utilisés :

Dataset 1 : Movies

Fichier : movies_metadata.csv

Texte utilisé : overview (description des films)

Corpus narratif contenant des thèmes variés (guerre, amour, famille, crime, etc.)

Dataset 2 : Cellphone Reviews

Fichier : Cell_Phones_and_Accessories_5.json

Texte utilisé : reviewText

Corpus composé d'avis clients contenant des vocabulaire lié aux produits et à l'évaluation

3.2 Architecture du projet

Le projet a été structuré en plusieurs fichiers Python afin de séparer clairement les différentes étapes du traitement.
Deux fichiers principaux ont été créés :
movies_word2vec.py
cellphone_word2vec.py
Dans chacun de ces fichiers, les étapes suivantes ont été implémentées :

- Lecture des données **(read_data)**
- Prétraitement et tokenisation **(prepare_data)**
- Entraînement du modèle Word2Vec **(train_model)**
- Sauvegarde du modèle dans le dossier TP1/models

Les modèles entraînés sont enregistrés sous forme de fichiers :

- **movies_w2v.model**
- **cellphone_w2v.model**

Le fichier **main.py** ne contient pas l'entraînement direct du modèle. Il appelle les deux scripts précédents. Étant donné que l'entraînement peut être long, un mécanisme de vérification a été ajouté : si le modèle existe déjà dans le dossier models, il est directement chargé sans relancer l'entraînement. Cela permet de passer immédiatement à la phase de test.

Pour évaluer les performances du modèle, une analyse qualitative a été réalisée à l'aide de la fonction :
test_keywords(model, words, model_name="Model")
Deux listes de mots ont été utilisées :

- Movies
  movie_test_words = ["love", "war", "family", "death", "king"]
- Cellphone
  cellphone_test_words = ["phone", "battery", "good", "screen", "price"]

3.3 Analyse des performances

Pour le dataset **Movies**, les résultats montrent une cohérence sémantique claire.
Le mot "love" est principalement associé à "madly" (63%), "romance" (62%) et "falls" (58%), ce qui correspond au champ lexical des relations affectives.
Le mot "war" est associé à "ii" (83%), "civil" (74%) et "vietnam" (72%), indiquant une forte cohérence avec les conflits historiques.
Le mot "king" est rapproché de "queen" (69%), "xiv" (68%) et "duke" (66%), ce qui reflète des relations hiérarchiques liées à la royauté.
Enfin, le mot "death" est associé à "deaths" (63%), "carnage" (58%) et "grieving" (58%), montrant une proximité sémantique autour du thème de la mortalité et des événements tragiques.

Ces résultats montrent une cohérence sémantique claire, puisque les mots retournés appartiennent au même champ lexical ou au même thème narratif. Par exemple, l'association entre "king" et "queen" indique que le modèle capture des relations hiérarchiques et thématiques liées à la royauté.

Pour le dataset **CellPhone**, les résultats reflètent clairement le domaine technique et évaluatif du corpus.
Le mot "phone" est associé à "cellphone" (66%), "iphone" (64%) et "device" (62%), ce qui montre une cohérence autour des appareils mobiles.
Le mot "battery" est associé à "batter" (89%), "mah" (74%) et "capacity" (73%), indiquant une proximité avec les spécifications techniques.
Le mot "good" est rapproché de "decent" (83%), "great" (77%) et "nice" (70%), ce qui confirme la cohérence des termes évaluatifs.
Le mot "screen" est associé à "glass" (76%), "protector" (75%) et "film" (74%), reflétant le vocabulaire lié aux accessoires.
Enfin, le mot "price" est associé à "value" (78%), "theprice" (73%) et "bargain" (72%), montrant que le modèle capture correctement les notions liées au coût et à la valeur.

Les résultats sont cohérents avec le domaine du corpus. Le modèle regroupe correctement les composants techniques tels que batterie et screen, ainsi que des synonymes évaluatifs comme good, great et excellent. Il associe également les notions liées au prix, notamment price, cost et value. Ces regroupements montrent que les représentations vectorielles reflètent la structure sémantique propre au dataset Cellphone.

4. Conclusion et limites

Les résultats obtenus confirment que le modèle Word2Vec apprend des représentations dépendantes du contexte, dans lesquelles les mots apparaissant dans des environnements similaires sont associés à des vecteurs proches. La structure sémantique observée reflète clairement le domaine du corpus utilisé. Le corpus Movies produit ainsi un espace sémantique principalement narratif et thématique, tandis que le corpus Cellphone génère un espace centré sur les produits et l'évaluation des consommateurs. Ces observations sont cohérentes avec les principes théoriques étudiés en cours.

Cependant, certaines limites doivent être soulignées. Le prétraitement appliqué reste relativement simple et ne comprend pas de lemmatisation, ce qui entraîne la séparation des formes morphologiques telles que "death" et "deaths". De plus, la tokenisation peut fragmenter certaines expressions composées, comme "World War II", ce qui explique la présence du token "ii" parmi les mots similaires à "war". Enfin, le modèle Word2Vec repose sur un contexte local et ne prend pas en compte la structure globale des phrases, ce qui limite sa capacité à distinguer les différents sens d'un même mot selon le contexte.