

Rapport : Entraînement et Évaluation d'un Modèle Word2Vec

Réalisé par : SKOURI Youssef

1 Introduction

Ce rapport présente la création, l'entraînement et l'évaluation d'un modèle de plongement lexical **Word2Vec** appliqué aux données textuelles issues du *Movies Dataset*. L'objectif principal est de construire des vecteurs de représentation des mots capables de capturer leurs relations sémantiques à partir du corpus prétraité.

Le travail s'appuie sur la bibliothèque **Gensim**, et toutes les étapes décrites ci-dessous correspondent fidèlement au contenu du notebook fourni.

2 Préparation des Données

2.1 Chargement du *Movies Dataset*

Les données sont importées depuis le fichier `movies_metadata.csv`. Seuls les champs textuels nécessaires à l'entraînement du modèle sont extraits. Le notebook lit les données via :

```
pd.read_csv("movies_metadata.csv")
```

2.2 Nettoyage et Prétraitement

Le texte est soumis à plusieurs transformations successives, telles qu'elles apparaissent dans le notebook :

- **Conversion en minuscules** pour homogénéiser le vocabulaire.
- **Suppression des caractères spéciaux** et ponctuation.
- **Tokenisation** à l'aide de NLTK.
- **Suppression des stopwords** (*the, and, in, etc.*).
- **Lemmatisation** avec `WordNetLemmatizer` afin de réduire les mots à leur forme canonique.

À l'issue de ces étapes, chaque description de film est convertie en une liste propre de tokens utiles pour Word2Vec.

3 Entraînement du Modèle Word2Vec

L'entraînement utilise l'implémentation `gensim.models.Word2Vec`.

3.1 Corpus

Le corpus d'entraînement est constitué de l'ensemble des textes nettoyés, représentés sous forme de listes de tokens. Chaque phrase ou document traité contribue à enrichir le vocabulaire appris.

3.2 Paramètres du Modèle

Les paramètres définis dans le notebook incluent :

- **vector_size** : dimension des vecteurs de mots.
- **window** : taille de la fenêtre de contexte.
- **min_count** : fréquence minimale pour qu'un mot soit conservé.
- **sg** : choix du modèle \rightarrow **sg=0** : **CBOW**, \rightarrow **sg=1** : **Skip-Gram**.
- **epochs** : nombre de passes sur le corpus.

Une fois ces paramètres fixés, le modèle est entraîné par la commande :

```
model = Word2Vec(...)
model.train(...)
```

4 Évaluation et Performances

L'évaluation du modèle dans le notebook repose sur les outils de similarité offerts par Gensim.

4.1 Similarités Entre Mots

Le notebook teste la cohérence du modèle via :

```
model.wv.similarity("mot1", "mot2")
```

Ce type de test mesure à quel point les vecteurs de deux mots sont proches en termes cosinus. Les similitudes retournées permettent de vérifier que des mots liés par le sens apparaissent effectivement proches dans l'espace vectoriel.

4.2 Recherche des Mots les Plus Similaires

Le modèle est également évalué via :

```
model.wv.most_similar("mot")
```

Cette fonction renvoie les mots dont les vecteurs sont les plus proches du mot donné. Les résultats obtenus dans le notebook montrent des associations pertinentes, confirmant que le modèle a bien capturé des relations sémantiques présentes dans le corpus.

4.3 Cohérence du Vocabulaire

La cohérence observée provient du fait que le corpus nettoyé contient de nombreux termes liés aux films : thèmes, genres, rôles, descriptions. Les tests de similarité montrent par exemple que :

- les genres se rapprochent d'autres genres,
- certains adjectifs reviennent près d'autres adjectifs similaires,
- les noms liés aux descriptions de films partagent des vecteurs proches.

Ces observations indiquent que l'entraînement Word2Vec a produit des plongements lexicalement cohérents.

5 Conclusion

Le modèle Word2Vec entraîné sur les données du *Movies Dataset* fournit des représentations vectorielles pertinentes. Les tests de similarité et de voisinage montrent que le modèle capture efficacement les relations sémantiques présentes dans les descriptions textuelles une fois celles-ci correctement prétraitées.

Ce travail démontre la capacité de Word2Vec à transformer un corpus textuel non structuré en un espace vectoriel riche et exploitable pour de futures tâches d'analyse.