Nom : TIEMTORE

Prénom : Wendyam Ariel Clauvis Eddy

Réponses aux questions

Exercice 1 : Questions de cours

1. Description des quatre étapes clés du prétraitement du texte :

- Tokenisation : elle consiste à segmenter des données textuelles brutes en tokens c'est-à-dire diviser le texte en unités lexicales comme les mots, les phrases.
- Normalisation ou suppression de la ponctuation : mise en forme uniforme des tokens : minuscules, suppression de la ponctuation, etc.
- Suppression des stop-words: Elimination des mots très fréquents (articles, prépositions) qui apportent peu d'information.
- Lemmatisation ou stemming: Réduction des mots à leur racine lexicale (lemme ou stem) c'est-à-dire le processus qui consiste à découper la fin des mots afin de ne conserver que la racine du mot.

2. La différence entre la lemmatisation et le stemming est que la lemmatisation est la réduction d'un mot à sa forme canonique (lemme) en utilisant un lexique et des règles grammaticales : le lemme est toujours un mot valide dans la langue ; tandis que le stemming est la réduction d'un mot à sa racine en supprimant les préfixes, suffixes : le stem n'est pas toujours un mot valide. En définitif, la différence entre les deux réside au produit final de la réduction, le lemme est un mot valide dans la langue tandis que le stem ne l'est pas toujours.

   On peut préférer l'une à l'autre selon ses objectifs. En effet, la lemmatisation produit des résultats plus précis et permet de mieux préserver le sens du texte tandis que le stemming est plus rapide mais peut conduire à des mots sans signification.

3. L'hypothèse distributionnelle a pour idée de modéliser le contexte d'un mot pour construire sa représentation vectorielle. Elle part du postulat que des mots apparaissant dans des contextes similaires ont des sens similaires.

Word2Vec s'appuie sur cette hypothèse pour apprendre des représentations vectorielles de mots c'est-à-dire, en paramétrant les mots comme des vecteurs denses et en apprenant à prédire le contexte en utilisant cette paramétrisation. En entraînant un modèle Word2Vec sur un grand corpus de texte, les mots ayant des contextes similaires se retrouveront proches dans l'espace vectoriel.

4. L'algorithme TF-IDF est une technique utilisée pour évaluer l'importance d'un mot dans un document par rapport à un corpus.TF est la fréquence d'apparition d'un mot dans un document et IDF l'importance inverse de la fréquence d'un mot dans le corpus.

Plus un mot est rare, plus son IDF est élevé. Ainsi, TF-IDF pénalise les mots très fréquents et met en avant les mots rares mais pertinents dans un corpus de document.

5. La différence entre les modèles Skip-gram et CBOW est que Skip-gram est plus adapté aux situations où on veut capturer les relations fines et plus rares entre les mots, tandis que CBOW est plus rapide et efficace pour les corpus avec une grande quantité de données.

Skip-gram prédit les mots contextuels à partir d'un mot cible tandis que CBOW prédit un mot cible à partir de ses mots contextuels.

Les word embeddings issus de ces modèles capturent des relations sémantiques et syntaxiques entre les mots. Il est donc possible d'effectuer des opérations linéaires simples sur ces vecteurs.

6. Les problèmes posés par les fréquences nulles de certains mots dans les classifieurs Naïve Bayes appliqués au texte résident dans le fait que les classifieurs Naïve Bayes supposent que les caractéristiques (les mots) sont indépendantes. Or, de nombreux mots peuvent ne pas apparaître dans certains documents, ce qui conduit à des probabilités nulles et pose des problèmes lors du calcul.

Pour les résoudre, nous pouvons, ajouter une petite valeur à toutes les fréquences pour éviter les zéros, ou utiliser un modèle de langage de plus haut niveau (bigrammes) pour estimer les probabilités des mots rares.

7. Les méthodes que nous pouvons utiliser pour générer des données supplémentaires lorsque les données disponibles sont limitées sont :

- Augmentation de données: remplacer les mots par des synonymes ; traduire le texte dans une autre langue puis le retraduire ; ajouter du bruit au texte c'est-à-dire supprimer des mots aléatoirement.
- Over-sampling**:** dupliquer les exemples de la classe minoritaire.
- Under-sampling**:** réduire le nombre d'exemples de la classe majoritaire.
- Génération de données artificielles: modèles génératifs (GANs, VAEs) pour créer de nouveaux exemples ; appliquer des règles simples pour générer de nouvelles phrases.

Exercice 2 : Recherche et Extraction d'Information dans les Moteurs de Recherche

1. Pour créer une représentation vectorielle (embeddings) pour ces données j'utiliserai des modèles de langage comme BERT basées sur des transformateurs qui permettent de générer des représentations contextuelles de mots, et cela va me permettre d'obtenir une meilleure représentation sémantique des requêtes entières.

Pour construire cette base de données vectorielle j'utiliserai le package python nommé « **Transformers** » qui me permet d'utiliser des modèles comme BERT pour extraire des embeddings à partir de requêtes en utilisant des modèles pré-entraînés.

2. Pour identifier des requêtes similaires, une fois que nous avons créé des embeddings pour chaque requête, nous pouvons utiliser des métriques de distance pour calculer la similarité entre la requête entrée et celles de la base de données.

Les métriques qui pourraient être utilisées pour évaluer la pertinence de la requête par rapport aux documents sont : Top-K Accuracy ; Mean Reciprocal Rank ; Mean Average Precision.

3. Décrivons comment un modèle de langue peut être utilisé pour proposer des complétions de requêtes à partir d'une requête et de la liste des documents similaires récupérés.

Tout d'abord nous allons encoder la requête et les documents similaires en utilisant le même modèle de langage (par exemple, BERT). Ensuite, nous allons générer les mots suivants de la requête, en se basant sur la représentation vectorielle de la requête et des documents similaires. Enfin, nous allons sélectionner les meilleures complétions car les complétions sont classées en fonction de leur probabilité et de leur pertinence par rapport à la requête initiale.

Exercice 3 : Questions sur les travaux pratiques

Partie 1 :

1. La fonction np.argpartition de NumPy est utilisée pour trouver les indices des éléments qui, s'ils étaient triés, seraient aux n premières ou dernières positions. En d'autres termes, elle partitionne un tableau en deux parties : les n éléments les plus grands (ou les plus petits) et les autres. Dans le code fourni, np.argpartition(-similarities, n) trouve les indices des n plus grandes valeurs dans le tableau similarities. Le signe moins devant similarities sert à inverser l'ordre, car np.argpartition trouve par défaut les plus petits éléments.

2. La fonction function_x semble implémenter une recherche de voisins les plus proches (Nearest Neighbors) dans un espace vectoriel.

En NLP elle peut être utilisée dans les problèmes suivants : recherche sémantique, recommandation (proposer des éléments similaires à ceux déjà consultés par un utilisateur), regrouper des éléments similaires en clusters et classification des documents en fonction de sa similarité avec les documents de cette classe.

Partie 2 :

1. Explication des paramètres du vectorizer :

- ngram_range=(1, 2): Indique que le vectoriseur va considérer des unigrammes (mots seuls) et des bigrammes (paires de mots consécutifs).
- max_df=0.5: Ignore les termes qui apparaissent dans plus de 50% des documents. Cela permet d'éliminer les mots très fréquents qui apportent peu d'information discriminante (comme les stop-words).
- min_df=5: Ignore les termes qui apparaissent dans moins de 5 documents. Cela permet d'éliminer les termes trop rares qui peuvent être spécifiques à un document et ne pas être généralisables.

2. Opération réalisée par la fonction fit

La fonction fit du TfidfVectorizer apprend le vocabulaire à partir du corpus de textes fourni. Elle construit un dictionnaire de tous les termes (unigrammes et bigrammes) qui respectent les critères max_df et min_df. Pour chaque document, elle calcule la fréquence de chaque terme et la pondère par l'inverse de sa fréquence documentaire (IDF). Cela permet d'obtenir une représentation vectorielle de chaque document où chaque dimension correspond à un terme du vocabulaire et la valeur de cette dimension représente l'importance du terme dans le document.