ROYAUME DU MAROC		المملكة المغربية
Université Abdelmalek Essaâdi		جامعة عبد المالك السعدي
Faculté des Sciences de Tétouan		كلية العلوم بتطوان
Tétouan		تطوان

TD 1 : From Text to Data

Professeur : Harchli Fidae

Exercice 1. Comparaison de NLTK et spaCy

1. Installer et configurer NLTK et spaCy.
2. Télécharger les ressources nécessaires.
3. Charger le modèle spaCy et afficher les composants du pipeline.
4. Pourquoi est-il nécessaire de télécharger des ressources supplémentaires avec NLTK ?
5. Quels sont les composants d'un pipeline spaCy et à quoi servent-ils ?
6. Tokeniser une phrase avec NLTK.
7. Tokeniser la même phrase avec spaCy.
8. Quels types de différences remarques-tu entre les résultats de NLTK et spaCy ?
9. Pourquoi la tokenisation est-elle une étape cruciale dans le NLP ?
10. Comparer la liste des stopwords en français avec NLTK et spaCy.
11. Supprimer les stopwords d'un texte avec NLTK et spaCy.
12. Les listes de stopwords sont-elles les mêmes entre NLTK et spaCy ? Pourquoi ?
13. Quelles sont les limites de la suppression des stopwords ?
14. Utiliser le Porter Stemmer avec NLTK.
15. Lemmatiser avec spaCy.
16. Quelle est la différence fondamentale entre le stemming et la lemmatisation ?
17. Dans quel cas préférerais-tu l'une ou l'autre méthode ?
18. Peux-tu donner un exemple où la lemmatisation serait plus avantageuse que le stemming ?
19. Identifier la catégorie grammaticale des mots avec NLTK puis avec spaCy.
20. Quelles sont les différences entre les résultats obtenus avec NLTK et spaCy ?

21. Pourquoi l'étiquetage morphosyntaxique est-il important ?

22. Utiliser spaCy pour identifier les entités nommées dans un texte.

23. Quels types d'entités sont détectées ?

24. Comment pourrait-on améliorer la reconnaissance des entités nommées ?

Exercice 2. **Exécuter le petit code suivant :**

```
from sklearn.feature_extraction.text import TfidfVectorizer

avis_clients = [
"Le produit est excellent et la livraison rapide",
"Très déçu, mauvaise qualité",
"Service client exceptionnel",
"Produit de mauvaise qualité, je ne recommande pas"
]

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(avis_clients)

print(vectorizer.get_feature_names_out())
print(X.toarray())
```

1. Que peut-on conclure sur les mots les plus discriminants dans les avis négatifs ?

2. Essayez d'utiliser CountVectorizer à la place : que remarquez-vous ?

3. Quelle vectorisation serait plus efficace pour entraîner un modèle de classification sentimentale ?

Exercice 3. **Exécuter le code suivant :**

```
from sklearn.feature_extraction.text import CountVectorizer

# Corpus avec et sans stopwords
corpus = [
"Le chat mange une souris.",
"Le chien aboie fort.",
"Une souris grise court très vite."
]

# Suppression manuelle des stopwords
corpus_filtré = [
"chat mange souris",
"chien aboie fort",
"souris grise court vite"
]

# Vectorisation Bag of Words
vectorizer = CountVectorizer()
```

```
X = vectorizer.fit_transform(corpus_filtré)

print(vectorizer.get_feature_names_out())
print(X.toarray())
```

1. Que représentent les lignes et les colonnes de la matrice BoW ?
2. Pourquoi avons-nous filtré les stopwords avant la vectorisation ?
3. En quoi cette représentation est-elle sensible au vocabulaire exact ?

Exercice 4. **Exécuter le code suivant :**

```
from sklearn.feature_extraction.text import TfidfVectorizer

corpus_lemmatise = [
"chat manger souris",
"chien aboyer fort",
"souris gris courir vite"
]

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(corpus_lemmatise)

print(vectorizer.get_feature_names_out())
print(X.toarray())
```

1. Quelle est la différence principale entre TF-IDF et BoW ?
2. Que signifie une valeur TF-IDF élevée pour un mot ?
3. Pourquoi la lemmatisation améliore-t-elle la qualité du vecteur TF-IDF ?

Exercice 5. Prétraitement et Vectorisation avancée

1. Générer le texte "Pride and Prejudice" de Jane Austen via NLTK.
2. Appliquer les méthodes de prétraitement nécessaires.
3. Implémenter et entraîner les modèles Word2Vec avec CBOW et Skip-gram via Gensim.
4. Visualiser les vecteurs par ACP pour évaluer leur qualité.
5. Tester les modèles avec deux tâches pratiques :
   — Recherche de mots similaires
   — Analyse des analogies
6. Comment améliorer davantage les performances du modèle ?
7. Tester les vecteurs sur un texte jamais vu et analyser la gestion des mots inconnus.
8. Comparer les modèles CBOW et Skip-gram :

— Lequel est meilleur pour des textes courts ou longs ?
— Impact des paramètres `window size` et `vector size` ?

9. Discuter des applications de Word2Vec dans des projets NLP réels.

Exercice 6.

1. **Chargement d'Embeddings Pré-entraînés :**
   - Qu'est-ce qu'un embedding pré-entraîné et pourquoi est-il utile pour le traitement du langage naturel ?
   - Quels sont les formats courants pour stocker des embeddings pré-entraînés ?
   - Comment charger des embeddings pré-entraînés dans des bibliothèques comme Gensim ou spaCy ?
2. **Comparaison de Similarité avec des Embeddings Pré-entraînés :**
   - Comment mesurer la similarité entre deux mots à l'aide de leurs embeddings ? Expliquer les différentes métriques possibles
   - Comment utiliser des embeddings pour résoudre des problèmes de similarité de phrases ou de textes ?
   - Quels sont les avantages et limites de l'utilisation des embeddings pré-entraînés pour comparer des mots ou des phrases dans différents contextes ?
3. **Applications des Embeddings Pré-entraînés :**
   - Comment les embeddings peuvent-ils être utilisés dans des systèmes de recommandation ?
   - De quelle manière les embeddings pré-entraînés peuvent-ils améliorer les performances des modèles de classification de texte ?
   - Comment utiliser les embeddings pour analyser des sentiments dans des données textuelles ?
4. **Fine-tuning des Embeddings :**
   - Quelle est la différence entre l'utilisation d'embeddings pré-entraînés et l'entraînement d'un modèle d'embedding à partir de zéro ?
   - Comment ajuster les embeddings pré-entraînés (fine-tuning) pour un domaine spécifique ?
5. **Problèmes avec les Embeddings Pré-entraînés :**
   - Quels sont les biais possibles dans les embeddings pré-entraînés et comment les atténuer ?
   - Quelles sont les principales limitations des embeddings pré-entraînés (par exemple, gestion des mots rares, polysemy, etc.) ?