

Année Universitaire : 2024/2025 Master 2 : SII Module : Recherche d'Information	Université des Sciences et de la Technologie Houari Boumediene Faculté d'Informatique Département d'Intelligence Artificielle et Sciences des Données	<b>TP N°4</b> Recherche de l'Information : Appariement Partie 1
---	---	---

## Support :

### 1. Extraction de termes (Tokens) à l'aide de l'expression régulière suivante :

```
nltk.RegexpTokenizer('(?:[A-Za-z]\. )+|[A-Za-z]+[\-@]\d+(?:\.\d+)?|\d+[A-Za-z]+|\d+(?:[\.\,\-]\d+)?%?|\w+(?:[\-/\]\w+)*')
```

## 2. Appariement :

### 2.1. Modèle vectoriel (Vector space model)

#### 2.1.1. Modèle basé sur le Produit Scalaire (Scalar Product)

##### Entrée (requête) :

Un ensemble de termes normalisés

##### Sortie :

Une liste de documents ordonnés selon leurs degrés de pertinences. Le degré de pertinence *RSV* d'un document *d* par rapport à une requête *Q* est calculé à l'aide de **Scalar Product** comme suit :

$$RSV(Q, d) = \sum_{i=1}^n v_i * w_i$$

$$Q = \langle v_1, v_2, v_3, \dots, v_n \rangle$$

$$d = \langle w_1, w_2, w_3, \dots, w_n \rangle$$

*n* : la taille du vocabulaire

*v<sub>i</sub>* : poids du terme *t<sub>i</sub>* dans la requête *Q* (par défaut *v<sub>i</sub>* = 1 si la requête *Q* contient le terme *t<sub>i</sub>*, 0 sinon)

*w<sub>i</sub>* : poids du terme *t<sub>i</sub>* dans le document *d*

### 2.1.2. Modèle basé sur la Similarité Cosinus (Cosine Measure)

**Entrée (requête) :**

Un ensemble de termes normalisés

**Sortie :**

Une liste de documents ordonnés selon leurs degrés de pertinences. Le degré de pertinence *RSV* d'un document *d* par rapport à une requête *Q* est calculé à l'aide de **Cosine Measure** comme suit :

$$RSV(Q, d) = \frac{\sum_{i=1}^n v_i * w_i}{\sqrt{\sum_{i=1}^n v_i^2} * \sqrt{\sum_{i=1}^n w_i^2}}$$

### 2.1.3. Modèle basé sur l'Indice de Jaccard (Jaccard Measure)

**Entrée (requête) :**

Un ensemble de termes normalisés

**Sortie :**

Une liste de documents ordonnés selon leurs degrés de pertinences. Le degré de pertinence *RSV* d'un document *d* par rapport à une requête *Q* est calculé à l'aide de **Jaccard Measure** comme suit :

$$RSV(Q, d) = \frac{\sum_{i=1}^n v_i * w_i}{\sum_{i=1}^n v_i^2 + \sum_{i=1}^n w_i^2 - \sum_{i=1}^n v_i * w_i}$$

## **Exercice :**

### **I. Implémenter les trois méthodes de recherche du modèle vectoriel:**

- . Produit Scalaire
- . Similarité Cosinus
- . Indice de Jaccard

### **II. Visualiser les résultats retournés par chaque méthode de recherche.**

RI Project 2024

Query

Large language models (LLM)

Search

☐ Queries Dataset

Processing

Token

RegEx

Stemmer

Porter

Index

☐

☒ DOCS per TERM

☐ TERMS per DOC

Matching

☒ Vector Space Model

Scalar Product

Content

bridge the user/item IDs and the template words and to unleash the power of LLM for recommendation. To address the problems, we propose to distill the discrete prompt for a specific task to a set of continuous prompt vectors so as to bridge IDs and words and to reduce the inference time. We also design a training strategy with an attempt to improve the efficiency of training these models. Experimental results on three real-world datasets demonstrate the effectiveness of our PrOmpt Distillation (POD) approach on both sequential recommendation and top-N recommendation tasks. Although the training efficiency can be significantly improved, the improvement of inference efficiency

Find

Results

N°doc	Relevance
5	0.6434
2	0.4792
3	0.4618
6	0.3217
4	0.2478
1	0.1806

Introduire la requête

Résultats retournés par le  
modèle vectoriel basé sur le  
Produit Scalaire

RI Project 2024

Query

Large language models (LLM)

Search

☐ Queries Dataset

Processing

Token

RegEx

Stemmer

Porter

Index

☐

☒ DOCS per TERM

☐ TERMS per DOC

Matching

☒ Vector Space Model

Cosine Measure

Content

bridge the user/item IDs and the template words and to unleash the power of LLM for recommendation. To address the problems, we propose to distill the discrete prompt for a specific task to a set of continuous prompt vectors so as to bridge IDs and words and to reduce the inference time. We also design a training strategy with an attempt to improve the efficiency of training these models. Experimental results on three real-world datasets demonstrate the effectiveness of our PrOmpt Distillation (POD) approach on both sequential recommendation and top-N recommendation tasks. Although the training efficiency can be significantly improved, the improvement of inference efficiency

Find

Results

N°doc	Relevance
5	0.1765
3	0.1709
2	0.1341
6	0.1132
4	0.1094
1	0.0520

Introduire la requête

Résultats retournés par le  
modèle vectoriel basé sur la  
Similarité Cosinus

RI Project 2024

Query

Large language models (LLM)

Search

☐ Queries Dataset

Processing

Token

RegEx

Stemmer

Porter

Index

☐

☒ DOCS per TERM

☐ TERMS per DOC

Matching

☒ Vector Space Model

Jaccard Measure

Content

bridge the user/item IDs and the template words and to unleash the power of LLM for recommendation. To address the problems, we propose to distill the discrete prompt for a specific task to a set of continuous prompt vectors so as to bridge IDs and words and to reduce the inference time. We also design a training strategy with an attempt to improve the efficiency of training these models. Experimental results on three real-world datasets demonstrate the effectiveness of our PrOmpt Distillation (POD) approach on both sequential recommendation and top-N recommendation tasks. Although the training efficiency can be significantly improved, the improvement of inference efficiency

Find

Results

N°doc	Relevance
5	0.0963
3	0.0861
2	0.0714
6	0.0565
4	0.0492
1	0.0264

Introduire la requête

Résultats retournés par le  
modèle vectoriel basé sur  
Indice de Jaccard