Indexeur Inversé Bilingue Fr-En

Contents

Module `indexer`

Classes

Class `BiInverIndex`

```
class BiInverIndex()
```

Indexe inversé bilingue français-anglais.

Attributes

`index : dict` L'index de la forme {'term':{'id_doc1':freq, 'id_doc2':freq, 'id_doc3':freq, ... }.
`plain_word_fr : Pattern` Le regex-pattern pour la détection des lemmes français.
`plain_word_en : Pattern` Le regex-pattern pour la détection des lemmes anglais.
`keep_path : str` Le nom du répertoire où seront stocké les fichiers indexés.
`index_name : str` le nom du fichier json pour sauvegarder l'index.

Methods

Method `add_doc`

```
def add_doc(self, file, id)
```

Ajoute un document à l'index.

Args

`file : str` Le chemin du document à ajouter.
`id` (int): l'identifiant du document.

Method `build_index`

```
def build_index(self, corpus_path, update=False)
```

Construit l'index inversé à partir d'un réperoire contenant des fichiers xml à indexer. Les fichiers doivent être sous la forme :

```
<article>
    <titre> </titre>
    <texte> </texte>
</article>
```

Args

corpus_path : str Le chemin du répertoire contenant les fichiers à indexer.
update : bool Indique si c'est une mise à jour de l'index. Dans ce cas l'état de l'index actuel sera récupéré. Sinon un nouvel index est crée.
Si un index existe déjà il sera supprimé avec accord de l'utilisateur False par défaut.

Method `check_state`

```
def check_state(self)
```

Récupère l'état actuel de l'index. Utilisé dans le cas d'un mise à jour de l'index.

Returns

currents_docs : list La liste des documents actuellement indexés.
id : int Le nouvel id, où va commencer l'indexation.

Method `clean_state`

```
def clean_state(self)
```

Tente de nettoyer l'environnement d'index Vérifie si le dossier "documentsIndex" et le fichier "index.json" existent déjà. Une demande de confirmation est demandée avant de les supprimer.

Returns

True si il n'y avait pas d'état ou que l'état a bien été réinitialisé. False si l'utilisateur a refusé le nettoyage.

Method `dump`

```
def dump(self)
```

Sauvegarde l'index dans un fichier json "index.json".

Method `get_freqs`

```
def get_freqs(self, text)
```

Récupère la fréquence des termes contenus dans un texte.

Args (str): Le textes à utiliser.

Returns (dict): Les fréquences des termes trouvés dans le texte.

Method `keep_doc`

```
def keep_doc(self, file)
```

Copie un document dans le dossier de sauvegarde : documentsIndex.

Args

file : str Le chemin du fichier à copier.

Method `parse_doc`

```
def parse_doc(self, xml_file)
```

Parse un document xml de la forme :

```
<article>
    <titre> </titre>
    <texte> </texte>
</article>
```

Args

`xml_file : str` Le chemin du fichier xml.

Returns

`text : str` Le contenu de la balise texte du fichier.
`title : str` Le contenu de la balise title du fichier.

Generated by *pdoc* 0.7.5 (https://pdoc3.github.io).