كلية الهندسة المعلوماتية

السنة الخامسة

نظم استرجاع المعلومات

إعداد الطلاب :

سارة أيمن الدعاس

تيماء سلمان أبو أحمد

نغم ناصر الظاهر عزام

ربى ناصر السيد

اختيار الـ datasets للبناء عليها:

* antique/train:

**تحتوي على** 404k docs و 27K qrels,2.4K queries

**وتحوي على مستنداتها على واصفات بسيطة هي:**

* **id**
* **text**
* bier/marco:

تحتوي على 376k docs و510 queries و36K qrels

وتحوي مستنداتها على الصفات التالية

* **text**
* **id**

معالجة البياناتPreprocessing Data:

1. إزالة stop words والأقواس الزائدة.
2. إزالة الإضافات stemming.
3. إرجاع الكلمات لأصلها lemmatization.
4. Spell checking لتصحيح الكلمات.

تمثيل البيانات Data representation:

مثلنا البيانات على شكل ملفات XML حيث يتم توزيع هذه الملفات في مجلدات ديناميكية حسب أسماء ومحتويات الداتا.

بالبداية قمنا بنقل البيانات كافة إلى ملف XML وهذا ما تقوم به مجموعة التوابع التالية:

* create\_directory
* create\_xml\_files
* read\_Data\_To\_XML
* creat\_XML
* read\_XML
* start\_work\_dataset
* readData\_antique\_doc\_in\_level\_0
* process\_chunk

قمنا باختيار العدد 200 ألف كعدد العناصر في كل ملف لكي يبقى كل جزء محافظاً على الشروط.

ونلاحظ أن عملية النقل شملت كل من (Documents, Queries, Qrels) .

تمثيل البيانات بعد معالجتها:

تعد عملية التمثيل عملية تكاملية مع معالجة البيانات لأن البيانات المعالجة تحتاج إلى تمثيل مناسب لها

عملية المعالجة تتم على عدة مراحل وذلك لتوثيق المعالجة والفرق بين كل مرحلة والمرحلة التي تسبقها.

توابع المرحلة الأولى:

* clean\_text

توابع المرحلة الثانية:

* lower\_text

توابع المرحلة الثالثة:

* Tokenize
* removeStopWords
* lemmatize
* get\_wordnet\_pos
* editing
* correct\_sentence\_spelling

ويتم تنفيذ المراحل ضمن التوابع التالية:

* transport\_xml\_to\_next\_level
* transport\_xml\_complix

بناء الفهرس العكسي inverted index:

الفهرس العكسي حيث ننطلق من الكلمة إلى المستند أي أننا نهتم بالكلمة وID المستندي الذي يحتويها فيمكن الوصول للمستند انطلاقاً من الكلمة.

تتم عن طريق ملفات xml أيضاً وبنفس المنهجية المتبعة في تمثيل البيانات إلا أن المجلد الذي يعبر عن نوع البيانات (Documents, Queries, Qrels) لم يعد يحوي ملفات مرقمة عشوائياً بل أصبحت تحوي مجلدات بواصفتها وكل واصفة (تتبع لمجموعة بيانات) تحوي على مجلدات مسماة حسب الأحرف الإنكليزية وبكل مجلد (حرف) يحوي ملفات ْXML بأحرف أيضاً، وبهذا نستطيع تخزين ID المستند الذي يحوي الكلمة في ملفها المناسب حسب الحرفين الأول والثاني منها.

وتتم العملية عن طريق التوابع:

* create\_clustersd
* indexing
* invert\_index
* transport\_xml\_files\_to\_clusterd\_pro
* add\_word\_to\_clusterd
* xml\_to\_json

تم اختيار منهجية تمثيل الفهرس تبعاً لمنهجية القاموس والتوزيع على مستوى المجلدات والملفات.

عمليات حساب ووزن الكلمات ضمن مجموعة البيانات:

تتم هذه العملية خلال نقل الكلمات إلى الفهارس حيث اخترنا وجود فهرسين اثنين:

حيث تعتمد فكرة الفرق بينهما على وزن الكلمة (الكلمة ذات الوزن العالي تعتبر خاصة)

وسيتم تخرين الكلمات الخاصة في الفهرس الخاص والهدف رفع أولوية هذه الكلمات وجعلها في مقدمة النتائج.

أسس حساب أوزان الكلمات بالنسبة للمستند وبالنسبة لكافة البيانات هي العمليات المتبعة في مقرر العملي:

* حساب tf (term frequency):

نقوم بحساب عدد مرور كل term ضمن ملف واحد one document عن طريق التابع التالي:

* calculate\_tf
* حساب idf :

نقوم بحسابه من أجل جميع documents عن طريق التابع التالي:

* calculate\_idf

حساب tf-idf :

نقوم بحسابه عن طريق القانون " tf \* idf "

* تخزين نتائج tf-idf :

قمنا بتخزين النتائج المصفوفة ضمن ملف من صيغة pickle

معالجة الاستعلامات query Preprocessing :

1. إزالة stop words ووالأقواس الزائدة
2. إزالة الإضافات stemming
3. إرجاع الكلمات لأصلها lemmatization

تمثيل البيانات query representation :

نقوم بتمثيل query بنفس طريقة تمثيل البيانات ولكن بداخل مجلد خاص لها

المصادر:

<https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>

<https://realpython.com/k-means-clustering-python/>

<https://www.javatpoint.com/what-is-sklearn-in-python>

<https://poe.com/ChatGPT>