مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (35) العدد (8) 2013
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (35) No. (8) 2013

استخدام المصادر المفتوحة لتحسين أداء نظم استعادة المعلومات

الدكتور أحمد صقر أحمد*
الدكتور قاسم قبلان**
عبد الحميد قريعة***

(تاريخ الإيداع 7 / 4 / 2013. قُبِل للنشر في 10/ 11 / 2013)

∇ ملخّص ∇

من أجل مواكبة التقدم الهائل لثورة المعلومات و توفرها على الوب تم وضع طرق واقتراحات من أجل تحسين فعالية عمليات البحث, معظم هذه الحلول ركزت على خوارزميات ترتيب الصفحات (Page Ranking) و معدل تردد الكلمة (.Term Freq) و لكن التركيز على استخدام علم دلالات الألفاظ و علاقة دلالات الألفاظ مع المحتوى رغم أهميته الكبرى ما زال قليلاً لأسباب مختلفة.

يهدف هذا البحث لإيجاد تصميم محرك بحث يعتمد على علم دلالات الألفاظ (Semantics)يمكن استخدامه للوصول إلى المعلومات ذات الطبيعة غير البنيوية مثل صفحات الوب,و يساعد في تحسين دقة و فعالية عملية البحث.

تم إجراء مجموعة من الاختبارات لاستنتاج تصميم محرك البحث و تقييم نتائج استخدام علم دلالات الألفاظ في التعامل مع عمليات البحث على صفحات الوب.

تم إجراء هذا البحث في جامعة تشرين في الفترة بين 2012/7/1 و 2013/3/15 .

الكلمات المفتاحية : الأنطولوجيا ، المصادر المفتوحة , علم دلالات الألفاظ ،محرك البحث لوسين.

* أستاذ – قسم النظم و الشبكات الحاسوبية– كلية الهندسة المعلوماتية– جامعة تشرين– اللاذقية– سورية.
** مدرس – قسم النظم و الشبكات الحاسوبية– كلية الهندسة المعلوماتية– جامعة تشرين– اللاذقية– سورية.
*** طالب دراسات عليا (دكتوراة)– قسم النظم و الشبكات الحاسوبية– كلية الهندسة المعلوماتية– جامعة تشرين– اللاذقية– سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (35) العدد (8) 2013
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (35) No. (8) 2013

Using Open Source in Enhancing Information Retrieval Systems Performance

Dr. Ahamd S Ahmad[*]
Dr. Kassem Kabalan[**]
Abdel Hamid Kreaa[***]

(Received 7 / 4 / 2013. Accepted 10 / 11 / 2013)

∇ ABSTRACT ∇

There are many methods and suggestions proposed to improve the efficiency of search in order to catch up with the increasing speed of information boom on the web. Most of these proposals are concentrated on term frequency and page rank algorithms and yet very few of them focus on semantic relationship of content.

The objective of this research project is to provide a semantic relationship model that can be used for semi-structured or unstructured information on the web to help improve the accuracy and efficiency of search engine.

Keywords: Ontology ,Open Source, semantics, Lucene Search Engine.

[*] **Professor, Department of Computer Systems and Networks, Faculty of Information Engineering, Tishreen University, Latakia, Syria.**
[**] **Assistant Professor, Department of Computer Systems and Networks, Faculty of Information Engineering, Tishreen University, Latakia, Syria..**
[***] **Postgraduate Student, Department of Computer Systems and Networks, Faculty of Information Engineering, Tishreen University, Latakia, Syria.**

مقدمة :

يعتبر علم استعادة المعلومات علماً حديثاً مما يسمح بوجود احتمالات كبيرة لتطويره. و عندما تريد مؤسسة ما تطوير محرك بحث خاص بها يوافق حاجاتها و متطلباتها تقوم هذه المؤسسة بتوظيف فريق من المبرمجين للقيام بذلك. سمح تطور الإنترنت بانتشار البرامج المصدرية التي تتمتع بإمكانيات كبيرة و تؤمن مساعدة المبرمجين و تسمح بتعاون عالمي لتطويرها, و هو ما أدى إلى ظهور مجتمع معلوماتي يسمى المصادر المفتوحة لتصميم النظم.

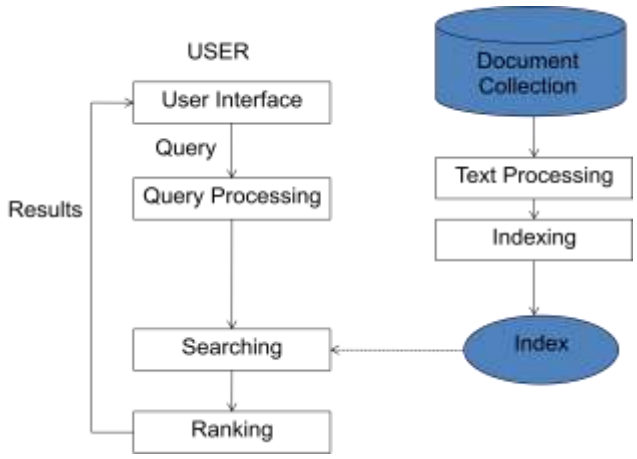
أصبحت كلمة المصادر المفتوحة تعني تصميم و تطوير و توزيع البرمجيات و لغاتها المصدرية للعموم. هذه العملية أدت إلى تحسين كبير في نوعية البرمجيات و تطورها كماً و كيفاً نظراً لمشاركة عدد كبير من المطورين في تحديثها و تنقيحها و إبداء الملاحظات عن الأخطاء و توثيقها و عدد الآراء المشاركة في عملية التصميم, كل ذلك بفضل وجود شبكة الإنترنت التي ساعدت بالإضافة إلى ذلك على ولادة مواقع إنترنت تحوي كميات كبيرة من برامج المصادر المفتوحة, من هذه المواقع sourceforge.net الذي يستضيف أكثر من 342000 مشروع و لديه أكثر من مليوني مستخدم مسجل لغاية العام الحالي 2013 [1].

1- نظم استعادة المعلومات:

هو علم إيجاد مواد عادة ما تكون وثائق ذات طبيعة غير بنيوية و التي غالباً ما تكون نصوص من ضمن مجموعة كبيرة من الوثائق, و التي تلبي حاجة المستخدم في الحصول على معلومات من ضمن مجموعة كبيرة من الوثائق موجودة على حاسب محلي أو مخدم أو على الوب [2].

الفكرة الرئيسة تكمن في تلبية حاجة المستخدم إلى معلومة فيقوم بعملية بحث ضمن مواد موجودة للحصول على معلومات تبدو أنها مناسبة أكثر,

لإتمام ذلك كان لا بد لنظم استعادة المعلومات أن تتألف من عدة أقسام كما يظهر الشكل (1) :



و الذي يتضمن الأقسام الأساسية التالية [3]:

الفهرسة (Indexing).

البحث (Searching).

الترتيب (Ranking).

الشكل (1) – مخطط يوضح عملية استعادة المعلومات

الفهرسة هي المسؤولة عن إعادة تمثيل و تنظيم المواد [illegible].

البحث هو المسؤول عن استخلاص المعلومات من الفهارس التي توافق حاجة المستخدم إلى المعلومة.

الترتيب (التصنيف) بالرغم من كونه عنصر اختياري فهو مهم في عملية الاستعادة لكونه يقوم بترتيب النتائج اعتماداً على خوارزميات تحاول تحديد النتيجة الأكثر ملاءمة لحاجة المستخدم.

2- علم دلالات الألفاظ على الوب Semantic Web :

يؤمن هذا العلم طريقة تصبح فيها الحواسيب أكثر قرباً من فهم المعلومات و طريقة ترتيبها. عندما يضع المستخدم في محرك البحث الحالي استعلاماً مثل " أفضل عشاء في دمشق" فإن محرك البحث سوف يعيد المقالات

التي تحوي كلمات الاستعلام بدون النظر إلى العلاقات المنطقية بين كلمات الاستعلام و لكن حسب أولوية ترتيب صفحات المقالات لديه.

هذا العلم يوفر حلول جديدة للاستفادة من المعلومات و إدارتها على صفحات الوب و يعتمد على المعلومات الجزئية التفصيلية للوثيقة و علاقتها ببعضها للحصول على فهم أكبر لها, إذ يتم استخلاص المعلومات من اللغة الطبيعية و دعم محرك البحث لفهم المعنى الدقيق للكلمات أو النص من الناحية اللغوية و المنطقية. وبشكل مثالي سوف تكون نتائج البحث عبارة عن لائحة من المعلومات المرتبة بشكل جيد مع أقل ما يمكن من التكرار فيها [4].

هذا ما قد يعني أنه يجب إعادة ترتيب مواقع الوب تبعاً لمعاني و دلالات الألفاظ منطقياً وليس إحصائياً فقط. إضافة إلى المسح المتكرر للموقع للتأكد من معلومات الصفحات المضافة.

في بيئة معلومات الإنترنت غير المتجانسة يصبح من الضروري الاستعانة بأدوات تدعم عمل محركات البحث بهدف الحصول على نتائج استعلام أكثر دقة و هذا العلم هو إحدى هذه الأدوات التي تساعد على ذلك عن طريق توصيف معاني المعلومات بشكل واضح لأدوات استعادة البيانات.

أهمية البحث و أهدافه:

هذا البحث يتناول مشكلة مهمة في عملية استعادة المعلومات وهي الاستفادة من استخدام علم دلالات الألفاظ في عمليات البحث, من خلال اقتراح آلية تسمح باستخدام علم دلالات الألفاظ في محركات البحث عن طريق مجموعة من الخطوات:

1- تمثيل المستندات موضوع البحث بقائمة من الكلمات المفتاحية التي تمثل المعنى الدلالي الوحيد للمستند, هذه القائمة تكون ذات حجم معقول قابل للإدارة و تشكل الأساس الذي نبني عليه هذه الآلية.

2- يتبع ذلك التصميم المنطقي للنماذج التي تسمح بتمثيل المستندات و كيفية تنفيذ العمليات التي تقوم بالبحث ضمن المستندات عن الكلمات المطلوب البحث عنها.

3- تطبيق هذه الآلية لتحقيق محرك بحث فعال, و إجراء تجارب تتحقق إحصائياً من فعالية التصميم المقترح.

سنقوم من خلال بحثنا بمحاولة استخدام برمجيات المصادر المفتوحة كأدوات في عملية استعادة المعلومات للوصول إلى محرك بحث قادر على استعادة المعلومات على شبكة الإنترنت و الاستفادة منها في إزالة الغموض في عبارات الاستعلام الخاصة بالبحث عن المعلومات.

وذلك من خلال :

1-التأكيد على الحاجة إلى أدوات استعادة معلومات فعالة مثل محركات البحث مع فوائد استخدام برمجيات المصادر المفتوحة خلال عملية التطوير.

3- تسليط الضوء على محرك البحث Lucene لإظهار مكوناته و أقسامه و مدى فائدتها في عملية البحث عن المعلومات و تطوير محركات البحث.

طرائق البحث ومواده :

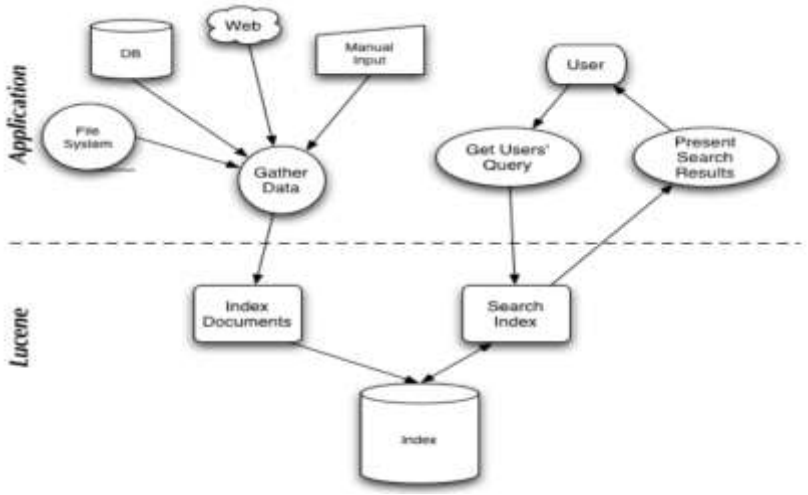
5-1-المصادر المفتوحة و استعادة البيانات :

مع انتشار برمجيات المصادر المفتوحة بشكل واسع أصبح من السهل إيجاد مشاريع تكون البرامج المصدرية لها جاهزة بشكل أولي و قد يجد المبرمج من يساعده خلال عملية تصميم أو تعديل المشروع.

و من ضمن هذه المشاريع يوجد عدد من محركات البحث و الأكثر شعبية منها هو Lucene لوسين من الشركة العالمية المشهورة Apache , و هو عبارة عن مكتبة تحوي برمجيات استعادة المعلومات ذات أداء عالي تسمح للمطورين بإضافة إمكانات البحث و الفهرسة على تطبيقاتهم, و استخدام هذه المكتبة كنقطة بداية سيوفر على المطورين وقتاً و جهداً كبيرين كما سيؤدي بالنهاية إلى تطوير برمجيات المكتبة لأن المطور سيضيف الميزات اللازمة لمشروعه و يقوم بتصحيح العيوب التي قد يجدها و إضافة بيئة عمل جديدة وإعداد الشروح و المستندات المتعلقة بالإضافات و التعديلات.

كما في بقية المكتبات فإن لوسين يقدم استخلاصاً (Abstraction)جيداً في عملية الفهرسة (indexing) والبحث (Searching) وتقدم واجهة بينية سهله الاستخدام، و هذا يعني أنه بإمكاننا استخدام لوسين في البرامج بدون أن تكون لدينا معرفة بآلية عمل الفهرسة والبحث وسوف نحصل على أداء عالٍ منها.

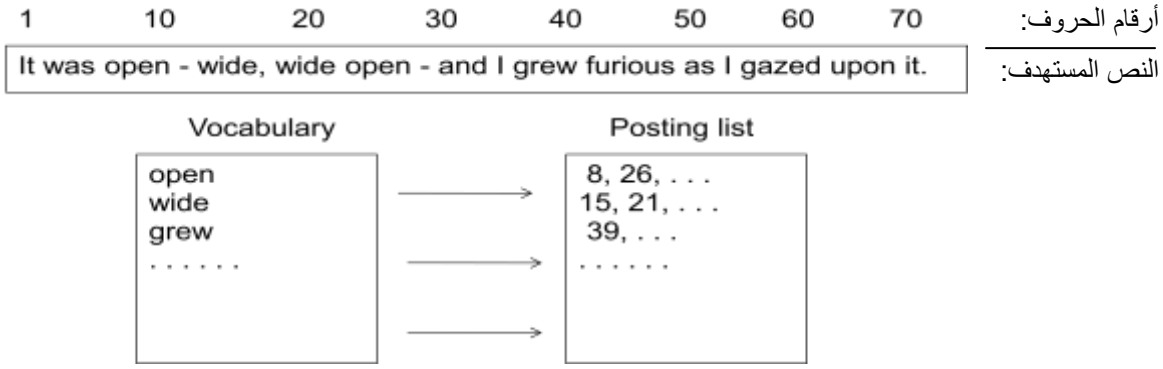
العديد من المشاريع كتبت بالاعتماد على لوسين منها Nutch و SearchBox و jSearch وغيرها الكثير والقائمة طويلة : (http://wiki.apache.o...-java/PoweredBy) إضافة إلى إمكانية النقل (porting) للغات البرمجية الأخرى ، فسوف يمكننا استخدام لوسين مع لغات مثل Perl , python , .NET, C++, java والوصول إلى الفهرس . و الشكل رقم (2) يوضح كيفية تفاعل التطبيقات مع لوسين.



الشكل (2) – مخطط يوضح تفاعل التطبيقات مع محرك البحث لوسين

كما هو واضح من الشكل (2) فإننا نستطيع أن نقوم بعمل فهرس للملفات من جهازنا أو بيانات مدخله من المستخدم أو صفحة ويب أو حتى قاعدة بيانات وبعد ذلك يمكننا البحث عنها بالطريقة التي نرغب، إذ يدعم لوسين استعلامات متقدمة سواء باستخدام المعاملات المنطقية (and, or) أو حتى باستبعاد كلمة معينة من خلال استخدام (- ,+) وغيرها من أنواع البحث مثلا البحث الضبابي (.Fuzzy Search).

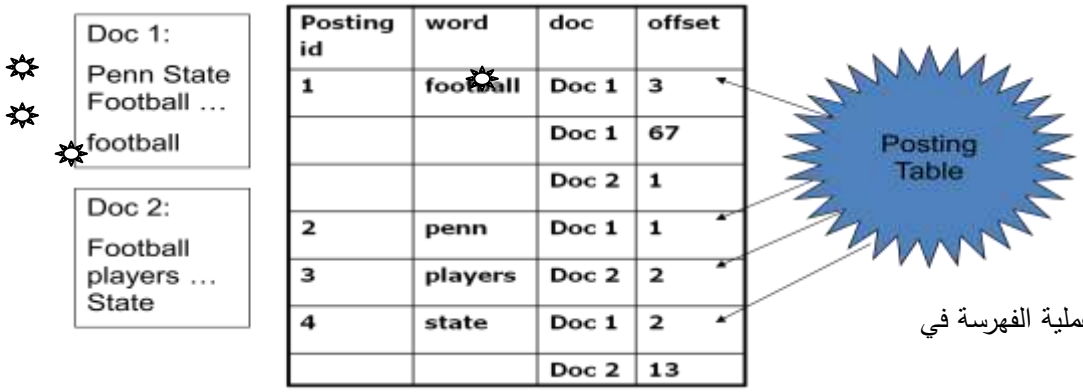
يستخدم محرك البحث لوسين طريقة تسمى الفهرس العكوس (Inverted Index) ,كما يوضح الشكل (3) التي يستخدم لفهرسة النصوص و هو يتألف من المفردات (Vocabulary) وتحوي الكلمات التي يتألف منها النص ولائحة العناوين (Posting list) التي تخزن عنوان كل كلمة من النص [5].



الشكل (3) – مخطط يوضح الفهرسة العكوسة

و يظهر الشكل (3) أن كلمة open موجودة في الموقع 8 والموقع 26 و كذلك تظهر مواقع الكلمتين wide و grew .

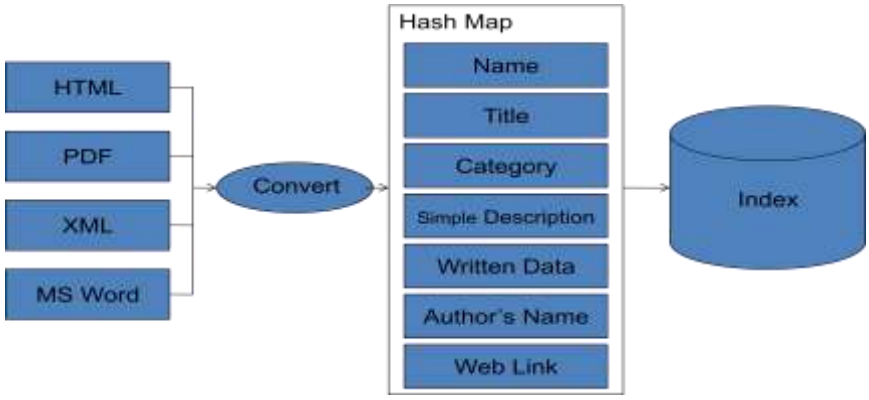
بينما يظهر الشكل (4) طريقة فهرسة أكثر من مستند [6] :



أما عملية الفهرسة في

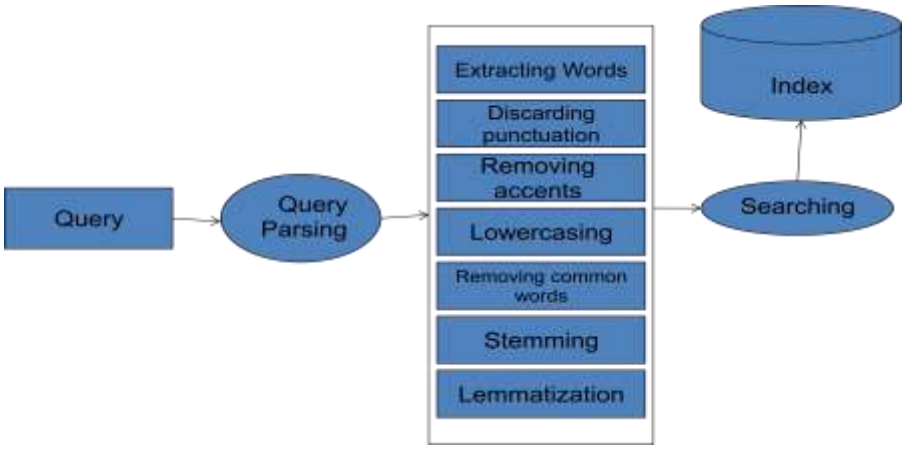
الشكل (4) – مخطط يوضح الفهرسة العكوسة لأكثر من مستند (مستندين).

هذا المحرك فيمكن توضيحها وفق الشكل (5) [7] :.



الشكل (5) – مخطط يوضح عملية الفهرسة في محرك البحث لوسين

بينما يظهر الشكل (6)عملية البحث ضمن الفهرس في محرك البحث لوسين [7] :



الشكل (6) – مخطط يوضح عملية البحث في محرك البحث لوسين

5-2-طريقة ترتيب نتائج البحث في محرك البحث لوسين:

في لوسين يتم ترتيب النتائج بطريقة تسمى TF-IDF(Term Frequency-Inverse Document Frequency) [8] والتي تعبر عن علاقة تربط بين عدد مرات تكرار الكلمة المطلوبة في وثيقة ما ضمن مجموعة من الوثائق المختلفة كما هو مبين في العلاقات التالية :

$$\mathrm{tf}_{t,d} = \frac{sum(td)}{sum(tD)}$$

$$\mathrm{idf}_t = \log_{10}\left(\frac{N}{\mathrm{df}_t}\right)$$

و يعطى وزن الوثيقة بالنسبة للكلمة المطلوبة بالعلاقة :

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$= \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$$

$$w_{t,d} = (1 + \log \mathrm{tf}_{t,d}) \times \mathrm{idf}_t$$

حيث :

td : هو عدد مرات تكرار الكلمة المطلوبة في الوثيقة.

tD : هو عدد مرات تكرار الكلمة المطلوبة في كل الوثائق.

N : هو عدد كل الوثائق.

df : هو عدد الوثائق التي تحوي الكلمة المطلوبة.

W : هو وزن الوثيقة بالنسبة للكلمة المطلوبة.

وأظهرت التجارب أن محرك البحث لوسين يتمتع بقدرة تنافسية عالية مع المحركات الأخرى مفتوحة المصدر من ناحية استخدام الذاكرة و زمن البحث (Searching Time) و حجم الفهرس (Index Size) و لكنه يظهر ضعفاً من ناحية الدقة عند أول خمس مستندات, و يظهر الجدول (1) مقارنة محرك البحث لوسين مع محركات أخرى مفتوحة المصدر [3]:

Search Engine	Indexing Time (h:m:s)	Index Size (%)	Searching Time (ms)	Answer Quality P@5
ht://Dig	(7) 0:28:30	(10) 104	(6) 32	-
Indri	(4) 0:15:45	(9) 63	(2) 19	(2) 0.2851
IXE	(8) 0:31:10	(4) 30	(2) 19	(5) 0.1429
Lucene	(10) 1:01:25	(2) 26	(4) 21	-
MG4J	(3) 0:12:00	(8) 60	(5) 22	(4) 0.2480
Swish-E	(5) 0:19:45	(5) 31	(8) 45	-
Swish++	(6) 0:22:15	(3) 29	(10) 51	-
Terrier	(9) 0:40:12	(7) 52	(9) 50	(3) 0.2800
XMLSearch	(2) 0:10:35	(1) 22	(1) 12	-
Zettair	(1) 0:04:44	(6) 33	(6) 32	(1) 0.3240

الجدول (1) – مقارنة أداء محرك البحث لوسين مع محركات بحث مفتوحة المصدر

حيث يظهر الجدول أن محرك البحث لوسين يحتل المرتبة الثانية من حيث صغر حجم الفهرس و المرتبة الرابعة لأقل زمن بحث و المرتبة العاشرة من حيث الزمن الذي يحتاجه لفهرسة الوثائق.

النتائج والمناقشة:

الهدف هو الوصول إلى المعلومات على الوب بشكل فعال أخذين بالحسبان أن شكل المعلومات على الويب مرتبة ضمن بنية ضعيفة أو بدون بنية على الإطلاق,و تنص الفرضية التي يستند إليها تصميم نموذج البحث المحسن على أنه :

"يمكن توصيف كل وثيقة من خلال قائمة من الكلمات المفتاحية و بالتالي يمكن تقليص حجمها إلى مستوى محدد و هذا يسمح بإيجاد لائحة بالكلمات المفتاحية التي تكون هي المفتاح في عملية التصميم"

و سيتم اعتماد استخدام الطرق الإحصائية للكلمات المفتاحية في التحقق من صحة الفرضية.

في الفرضية المقترحة يقوم نموذج البحث المحسن بالوصول إلى الوثيقة المحددة على الوب من خلال مجموعة من الكلمات المفتاحية والتي تشكل المعنى الدلالي للوثيقة بشكل خاص (هذه اللائحة نفترض أن حجمها قابل للإدارة).

سنناقش هنا هذه الفرضية من الناحية المنطقية و العمليات التي يمكن من خلالها تطبيق هذه النظرية و وضع تصميم أولي لنموذج البحث المحسن,الذي سيسمح بتحسن أداء عمليات البحث.

اعتماداً على عدد مرات تكرار الكلمات المفتاحية (لائحة الكلمات المفتاحية) يمكن ربط هذه اللائحة مع بعضها لتشكيل بنية ذات معنى دلالي محدد مما يمكننا من تشكيل نموذج معطيات لهذه اللائحة في فضاء مستندات على الوب و من ثم استخدام هذا النموذج في تصميم نموذج البحث المحسن.

هذه الفكرة تتضمن ثلاث خطوات :

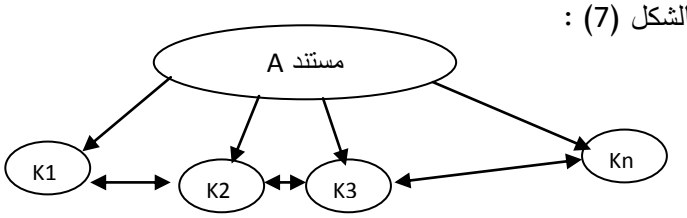
- إيجاد لائحة بالكلمات المفتاحية.
- تحديد العلاقات بين الكلمات المفتاحية في اللائحة.
- استخدام هذه اللائحة بعد ربطها مع بعضها بالعلاقات المناسبة ضمن التصميم.

هذه الخطوات تتكرر ضمن حلقة حتى نتمكن من تكوين بنية مفاهيم (انطولوجيا) مناسبة يمكن وضعها ضمن التصميم لبناء ما يسمى البحث باستخدام دلالات الألفاظ و هو ما سيساعد على تحسين دقة البحث.

بفرض أن R هو فضاء التجربة و هو يحوي مجموعة من وثائق الوب n و بفرض أن r_i هي عنصر من عناصر هذا الفضاء حيث:

$$R=\{ r_i \mid 1 \le i \le n\}$$

و بفرض لدينا الوثيقة A (عبارة عن مستند قد يكون نوعه PDF أو صفحة وب) تملك لائحة بالكلمات المفتاحية Ki كما هو موضح بالشكل (7) :



الشكل (7) – مخطط يوضح العلاقة بين المستند A و الكلمات المفتاحية Ki .

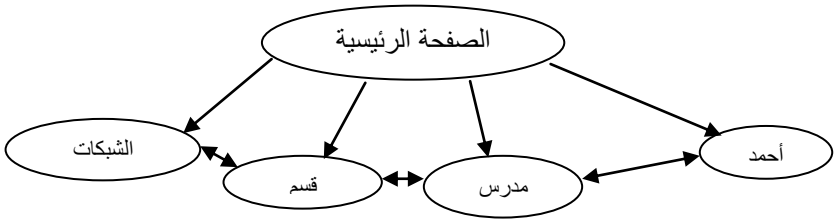
يمكن توصيف العلاقات بين A و اللائحة K1,K2,.....,Kn بما يلي:

- K1,K2,.....,Kn تحدد بشكل فريد الوثيقة A و الوثيقة A تحوي K1,K2,.....,Kn .
- K1,K2,.....,Kn هي مفاهيم مستقلة و ليس لها علاقة ببعضها البعض.

لفهم ذلك نفرض أن لدينا صفحة مدرس في الجامعة يعمل في قسم الشبكات و يملك الاسم أحمد, إن لائحة الاستعلام المفتاحية سوف تكون بالشكل التالي :

{"مدرس","الشبكات","قسم","أحمد"}

كما هو موضح بالشكل (8):



الشكل (8) – مخطط يوضح العلاقة بين الصفحة الرئيسية و لائحة الاستعلام المفتاحية

لإثبات صحة هذه النظرية سنقوم بإجراء اختبارات الغاية منها هي التأكد من صحة فرضية البحث التي سيبنى على أساسها تصميم نموذج البحث الجديد. و يظهر الجدول (2) مواصفات الاختبارات

الجدول (2) – جدول يوضح مواصفات الاختبارات

مواصفات التجربة	التجربة الأولى	التجربة الثانية
نوع الوثائق	HTML	PDF
خوارزمية ترتيب النتائج	TF-IDF	Page Rank [9]
منصة العمل	Lucene	Google API
فضاء العمل	صفحات موقع جامعة (حوالي 300000 صفحة)	صفحات غوغل (حوالي 10 مليار صفحة)

6-1- الاختبار الأول:

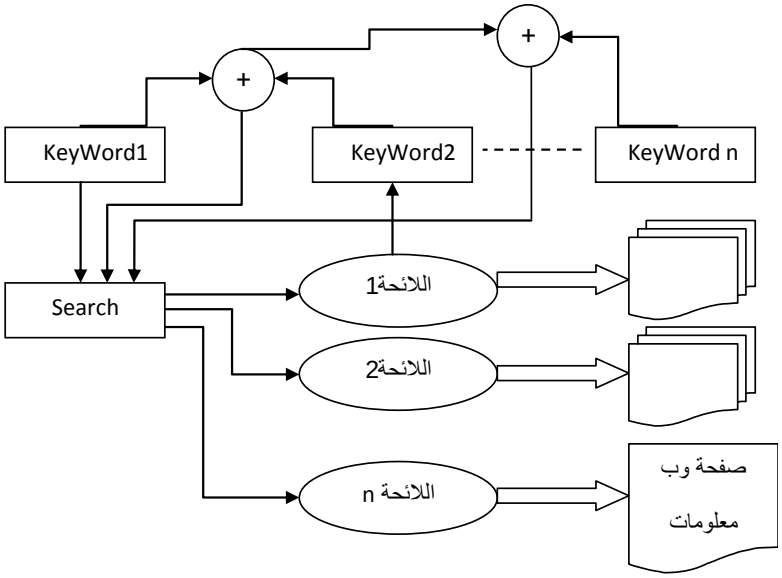
نقوم باستخراج لائحة بالكلمات المفتاحية بشكل يدوي من فضاء التجربة, نقوم بإجراء عملية بحث على كلمة وتكون الإجابة التي يعيدها محرك البحث هي لائحة نطلق عليها اسم (اللائحة1).

من أجل أي عملية بحث تالية فإن كلمات البحث المفتاحية التي يقدمها المستخدم يتم مطابقتها مع الكلمات المفتاحية من اللائحة1

إضافة كلمة ثانية في عملية البحث سيؤدي إلى لائحة من الإجابات يعيدها محرك البحث نطلق عليها اسم (اللائحة2).

يمكن استخدام الكلمة الأولى و الثانية للبحث في المرة الثالثة و هكذا . و بعد عدد من عمليات اختيار الكلمات يتم إعداد لائحة من الكلمات المفتاحية.

و تبعاً لهذه العملية فإنه مهما تم إضافة كلمات إلى اللائحة فإن محرك البحث سوف يعيد نفس محتوى الإجابات .وهذه اللائحة من الكلمات توصف كل تفاصيل المعلومات التي يجب العثور عليها, و ذلك كما هو موضح في الشكل (9).



الشكل (9) – يوضح خطوات الاختبار الأول

لتوضيح ذلك أكثر نقدم المثال التالي :

يريد المستخدم تحديد صفحة مدرس في الجامعة اسمه عيسى يعمل في قسم التاريخ, بفرض أن المستخدم قام بكتابة **(تاريخ)** كأول كلمة للبحث عنها.

كلمة تاريخ قد تكون حدثاً تاريخياً أو قصة أو فرع من فروع المعرفة التي تسجل و تحلل الأحداث التاريخية أو قد يكون المقصود بها دراما تاريخية أو ربما ما يتعلق بتاريخ جامعة تشرين أو مواد قسم التاريخ, و هكذا سيكون عدد الإجابات المعادة كبيراً (حوالي 500 إجابة). يتم تخزين النتيجة في زوج من البارامترات K1 لكلمة البحث (تاريخ) و اللائحة1 لنتيجة البحث. اعتماداً على اللائحة1 يقوم المستخدم بتقديم كلمة البحث الثانية و هي كلمة (قسم). كلمة قسم تعني كل أقسام الجامعة بما فيها الكليات و الإدارة و ...إلخ, بينما قسم التاريخ هو قسم محدد في الجامعة يعنى بتدريس مادة التاريخ , ينتج لدينا زوج جديد من البارامترات K2 لكلمة البحث الجديدة **(قسم التاريخ)** و اللائحة2 لنتيجة البحث الجديدة.

اعتماداً على اللائحة2 يقوم بعدها المستخدم بتقديم كلمة البحث الثالثة و الرابعة و هي كلمة **(المدرس عيسى)**.مما ينتج البارامترات الجديدة K3 و K4 و اللائحة3 و اللائحة4.

لنفرض جدلاً أن هناك سجل واحد يتوفر للمدرس عيسى في قسم التاريخ, نجد بعد متابعة العمليات السابقة و إضافة كلمات جديدة لتضيق نطاق البحث سنصل إلى حد لا يتغير معه حجم اللائحة الناتجة.

عندها تعتبر مجموعة البارامترات (K1,K2,K3,K4) التي تحوي **(تاريخ , قسم,مدرس,عيسى)** أقل مجموعة من الكلمات التي يمكن استخدامها لتحديد المعنى الدلالي لصفحة المدرس عيسى بشكل مميز (كلمات مفتاحية)

2-6- نتائج الاختبار الأول:

1- عند إدخال كلمة "**التاريخ**" التي تشكل جملة البحث الأولى في منصة البحث الشكل (10) أعطت في الجولة الأولى من البحث 180000 نتيجة الشكل (11).

الشكل (10) – يوضح واجهة الاختبار و جملة البحث الأولى

2- عند إضافة كلمة "قسم" إلى منصة البحث تتشكل جملة البحث الثانية

"**قسم التاريخ**"

و التي ينتج عنها 8560 نتيجة الشكل (12).

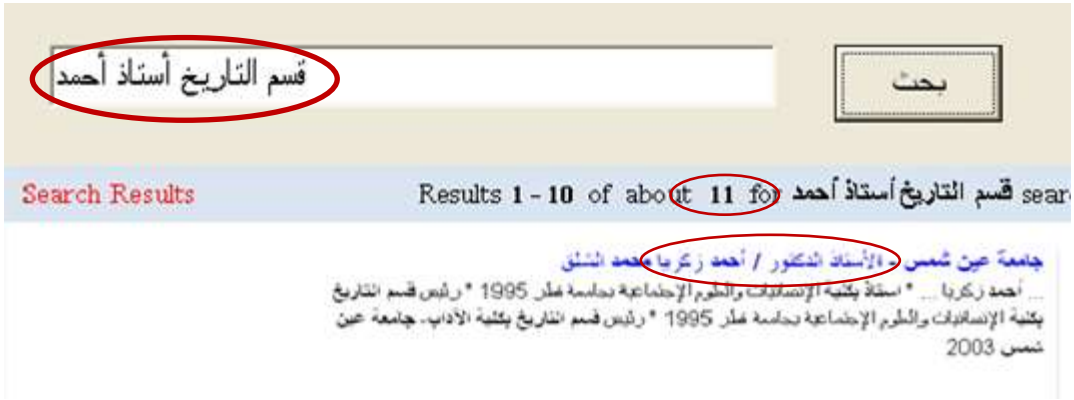
الشكل (11) – نتيجة البحث لجملة الاستعلام الأولى

3- عند إضافة كلمة "أستاذ" إلى منصة البحث تتشكل جملة البحث الثالثة :" **قسم التاريخ أستاذ**"

و التي ينتج عنها 134 نتيجة الشكل (13).

الشكل (13) – إدخال جملة البحث الثالثة و نتيجة البحث

4- عند إضافة كلمة "أحمد" إلى منصة البحث تتشكل جملة البحث الرابعة : " قسم التاريخ أستاذ أحمد", و التي ينتج عنها 11 نتيجة, الشكل (14) و تظهر النتائج أنه يمكن الوصول إلى **الصفحة المطلوبة**



الشكل (14) – إدخال جملة البحث الرابعة و نتيجة النتيجة

وبعد إجراء مجموعة من عمليات البحث على جمل بحث مختلفة تم تجميع نتائج الاختبار الأول في الجدول التالي (3) حيث يمكننا من خلاله ملاحظة ما يلي :

- في اللائحة رقم (1) عند إدخال كلمة " **موظف** " أعطت في الجولة الأولى من البحث لائحة تتكون من **2122** نتيجة, و بعد إضافة كلمة " **مدرس** " إلى جملة الاستعلام أعطت عملية البحث في الجولة الثانية لائحة تتكون من **567** نتيجة,و بعد إضافة كلمات مفتاحية أخرى إلى جملة الاستعلام لتصبح " **موظف , مدرس , أستاذ** "انخفض هذا العدد إلى **322** نتيجة.
- لائحة الكلمات المفتاحية " **قسم , التاريخ , أستاذ , أحمد, زكريا** " أعطت في الجولة الأولى من البحث لائحة تتكون من **180000** نتيجة, و بعد إضافة كلمات مفتاحية أخرى إلى جملة الاستعلام انخفض هذا العدد إلى **صفحة واحدة** و هي الصفحة المطلوبة.

الجدول (3) – جدول يظهر نتائج الاختبار الأول

رقم اللائحة	لائحة الكلمات المفتاحية	عدد الصفحات نتيجة البحث				
		جولة 1	جولة 2	جولة 3	جولة 4	جولة 5
1	موظف , مدرس , أستاذ	2122	567	322		
2	قسم , التاريخ , أستاذ , أحمد, زكريا	180000	8560	134	11	1
3	طالب , خريج , كلية , قسم , محمد	4350	1022	365	133	36
4	بحث , ورقة , موضوع , كاتب	632	126	42	1	

3-6- الإختبار الثاني :

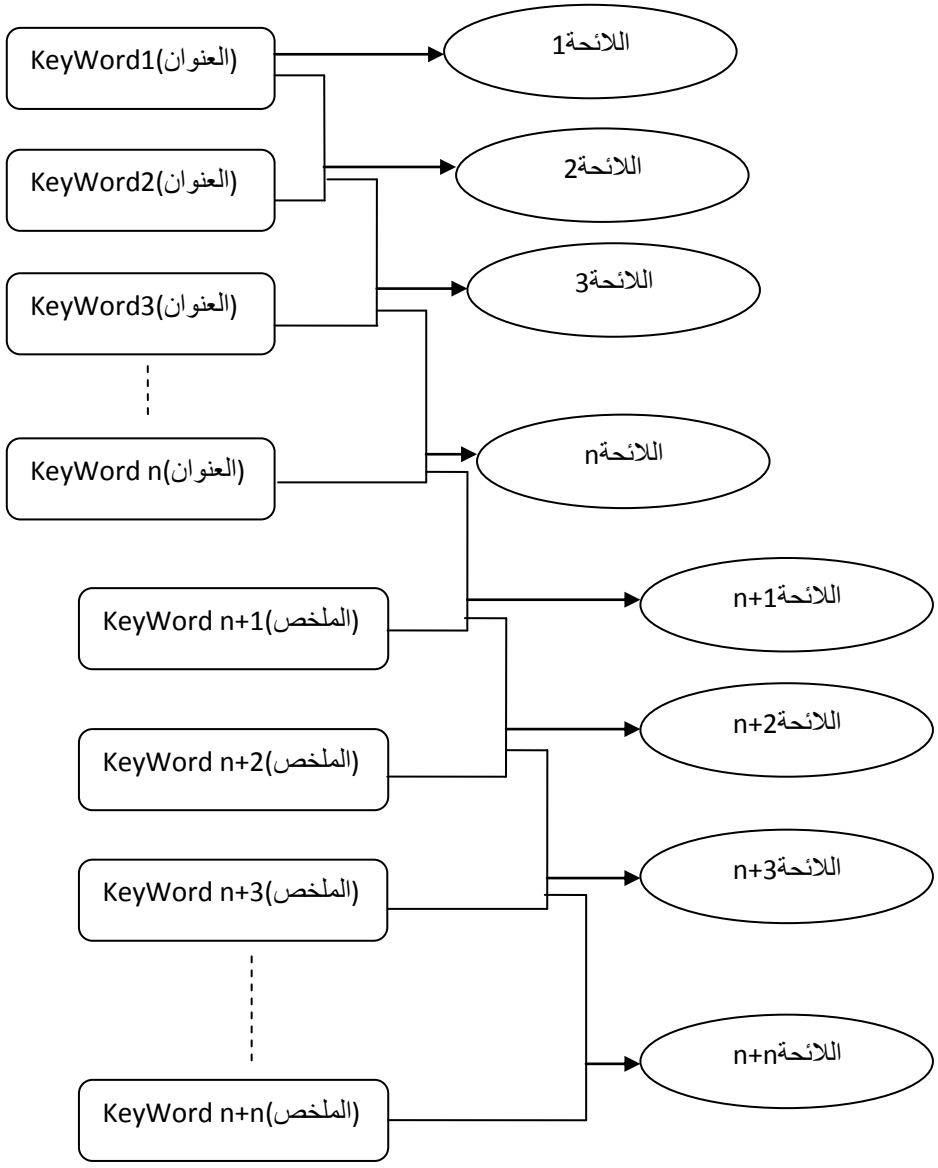
هذا الاختبار مخصص للبحث على الوب عن الوثائق التي ليس لها بنية واضحة ونعني هنا المستندات التي نوعها PDF. حيث سنحاول من خلال هذه التجربة تأكيد صحة فرضية نموذج البحث المصمم, و ستكون منصة البحث هنا واجهة محرك البحث غوغل google API (www.google.com).

يتم تجميع الكلمات تبعاً للعنوان و المؤلف و ملخص مستندات PDF و من ثم استعمال هذه الكلمات للبحث ضمن غوغل, و يتم تسجيل عدد النتائج المعادة.

لتحديد أي من وثائق PDF بعينها يتم تجميع مجموعة من الكلمات التي تعتبر كلمات مفتاحية و ضرورية حتى تتمكن منصة غوغل من البحث عنها و الوصول إليها.

بعد أن يتم تجميع الكلمات المفتاحية يتم تحليلها و تحليل النتائج إحصائياً.

و يوضح الشكل (15) مراحل سير التجربة :



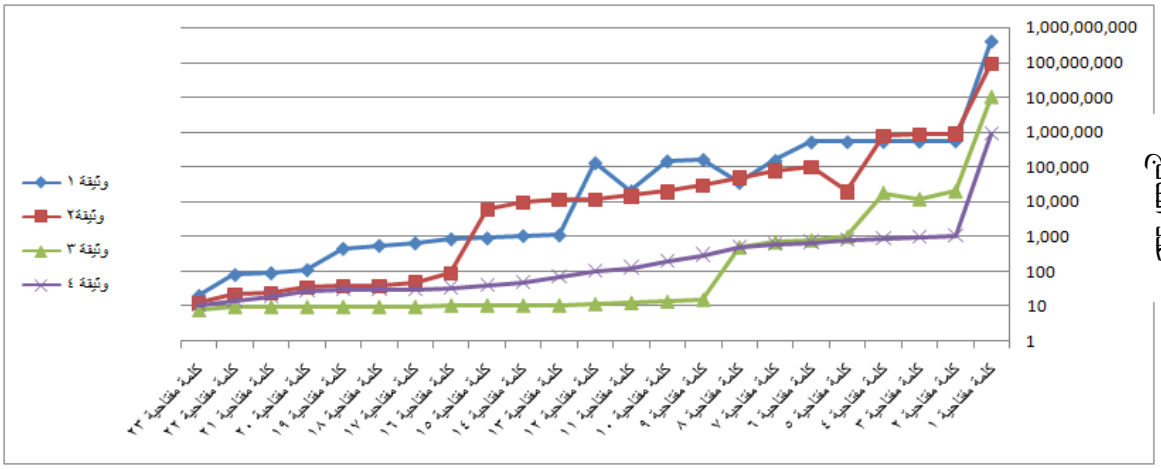
الشكل (15) – خطوات الاختبار الثانى

6-4- نتائج الاختبار الثاني:

الغاية من تحليل نتائج الإختبار هي التحقق من صحة فرضية نموذج البحث المحسن:

مرحلة تجميع البيانات : تم اختيار فضاء اختبار من ضمن عدد كبير من الصفحات تتجاوز 10 مليار صفحة حيث تم تجميع 100 وثيقة من نوع Pdf باللغة العربية و بشكل عشوائي و تم اختيار أربع وثائق من بينها.

1- مرحلة تحليل البيانات : بعد اجراء الاختبار تم أخذ إحصاءات لأقل مجموعة من الكلمات المفتاحية بتجريب كلمات مفتاحية, و يظهر الشكل (16) الذي يوضح عدد النتائج المعادة لكل كلمة مفتاحية , أن عدد هذه النتائج يميل للاستقرار بعد إضافة مجموعة من **الكلمات المفتاحية** حيث تعتبر هذه المجموعة أنها **تعبر عن الوثيقة بشكل مميز و هو ما يطابق الفرضية المقترحة.**



الشكل (16) – نتائج الاختبار الثانى

7- نموذج البحث المحسن:

اعتماداً على نتائج الاختبارات السابقة يمكن التأكد من صحة نظرية نموذج البحث المحسن و يظهر الشكل (17) بنية هذا النموذج حيث تتكون عملية اكتساب المعلومات من الخطوات التالية:

1- يتم تحليل الوثائق من الأنواع المختلفة من أجل تحويلها إلى أدلة متسلسلة من النص الواضح.

2- يقوم محلل الفهرسة البسيط بتحويل النص إلى أدلة من خلال عمليات استخراج الكلمات و إسقاط علامات التنقيط و الكلمات المشتركة و تحويل الكلمة إلى الجذر أو تحويل الكلمات إلى صيغتها الأساسية.

3- يقوم محلل الفهرسة المعمق الذي صمم لتحليل نص الأدلة بإجراء نمذجة للنص الناتج حيث تم دمج الأدلة المتكررة من أجل تخفيض التكرار و في نفس الوقت يقوم تابعان للعمل على دعم هذه العملية من خلال نمذجة المعلومات و التي تعتبر عملية مهمة لتحسين فعالية محرك البحث.

4- عملية بناء الانطولوجيا حيث يستخدم نص الأدلة المنمذج في بناءها و الذي يسهم في تحسين عمليات البحث.

أما خطوات استعادة المعلومات تتم من خلال :

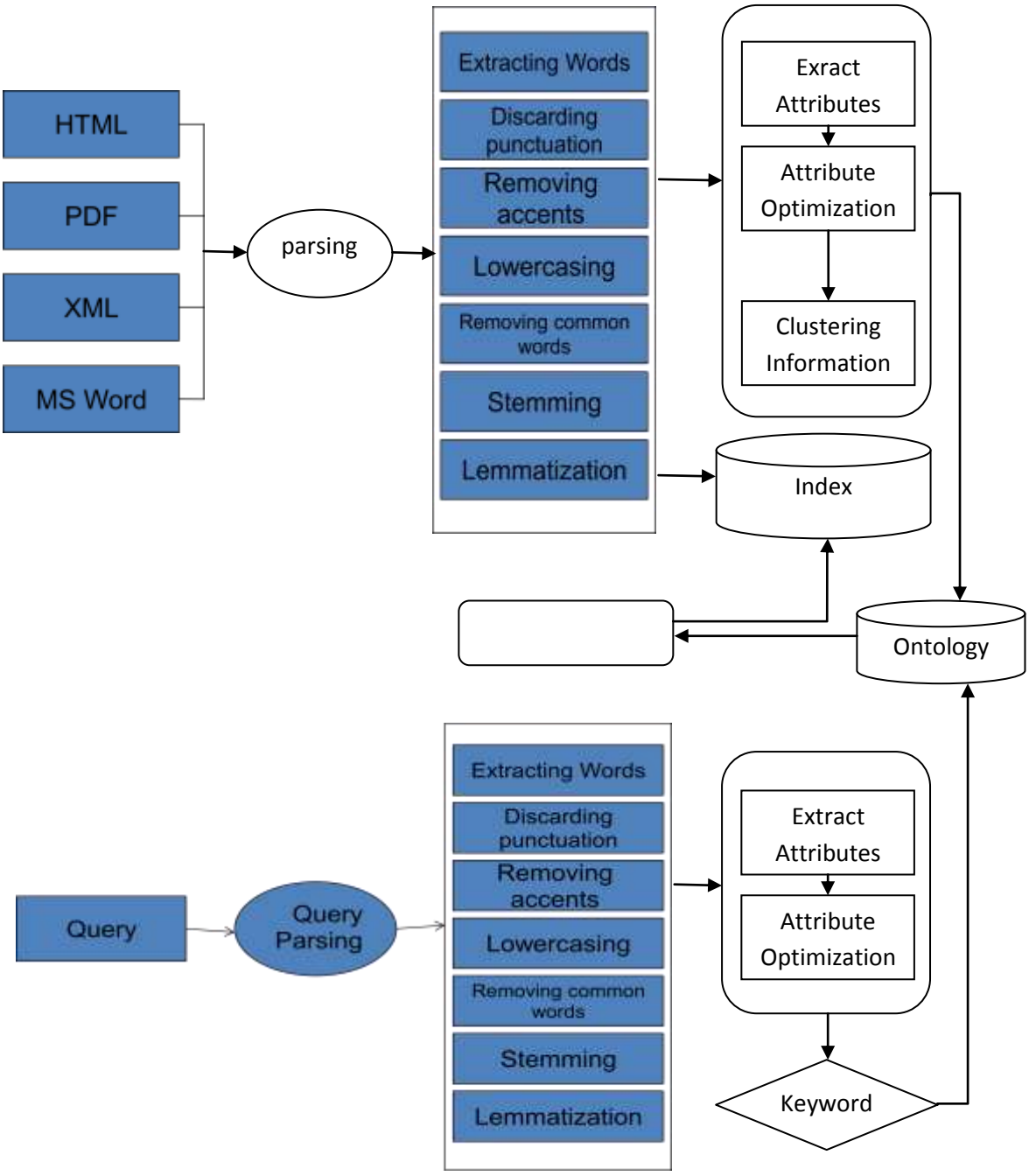
1- تحليل الاستعلام إلى مجموعة من الكلمات.

2- يقوم محلل الاستعلام البسيط بتحليل إضافي للاستعلام.

3- يقوم محلل الاستعلام المعمق باستخلاص المصطلحات من الاستعلام و وضعها بشكل نموذجي .

4- يتم بعدها الاعتماد على الانطولوجيا.

5- أخيراً تبدأ عملية البحث.



الشكل (17) – بنية نموذج البحث المحسن المصمم

الاستنتاجات والتوصيات:

1-يمكن عند تنفيذ النموذج ترتيب المعلومات تبعاً لمعناها و بالتالي استخلاص معلومات إضافية و جديدة تعذر الوصول إليها من قبل.

2-يمكن الحصول على أجوبة صديقة للمستخدم بناءً على نظام الاستعلام الجديد حيث تصبح الأسئلة هي الدخل و النظام سوف يعطي المستخدمين التوجيهات للحصول على المعلومات حسب رغبتهم و سيحصل المستخدمون على إجابات من وثائق مختلفة.

3-أظهرت الاختبارات أهمية تأثير الكلمات المفتاحية في عملية البحث و قد تم التأكد من أن عدد الكلمات المفتاحية محدود و دقيق و هو المؤثر الأهم في عملية البحث.

4-يتم استخدام الانطولوجيا للمساعدة في تحديد الكلمات المفتاحية في جمل الاستعلام و مساعدة المستخدم للحصول على النتائج المطلوبة

المراجع :

1- Sourceforge.net ,2013"What is SourceForge.net?"
http://sourceforge.net/apps/trac/sourceforge/wiki/What%20is%20SourceForge.net? (Accessed: Nov 2013)
2- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK, 2008, 482 Pages..
3-Christian Middleton, Ricardo Baeza-Yates. "A Comparison of Open Source Search Engines" 2008,46 Pages.
4-Semantic Web (2008) Available at: http://en.wikipedia.org/wiki/Semantic_web
(Accessed: Nov 2012)
5-Clarke, C & Cormack G (1995), Dynamic Inverted Indexes for a Distributed Full-Text
Retrieval System, TechRep MT-95-01, University of Waterloo, February 1995, 13 Pages.
6- S Kataria, T Paul .M Robbins , Document Indexing and Scoring in Lucene. IST 441 Spring 2009, 26 Pages.
7-Cutting, D (2006) Apache Lucene: Index File Format. Available at:
http://lucene.apache.org/java/2_1_0/fileformats.pdf (Accessed: Sep 2012)
8-http://lucene.apache.org/java/3_0_1/api/core/org/apache/lucene/search/Similarity.html (Accessed: Aug 2012)
9-Brin, S & Page L (1998) The anatomy of large-scale hyper textual web search engine,
Computer Science Department, Stanford University, 11 Pages.