Ministry of Higher Education and Scientific Research University of Kasdi Merbah Ouargla Faculty of New Technologies of Information and Communication Department of Computer Science and Information Technologies

ACADEMIC MASTER

Domain: Mathematics and Computer Science

Faculty: Computer Science

Specialty: Industrial Computing

PARALLEL ADAPTATIVE MEMORY FOR QUERY EXPANSION

Mr Korichi.M	**President**	**UKM Ouargla**
Mr Benkherourou.CH	**Examiner**	**UKM Ouargla**
Mr BEKKARI Fouad	**Supervisor**	**UKM Ouargla**

Presented by:

CHENNOUF IKRAM

GANA NESRINE

Discussed on: 17/06/2023

Year University: 2022/2023

Thanks

Firstly, after all the waiting and after the years of study, here we are now with you to share our joy of arriving at this moment, after good luck from ALLAH who inspired us with patience and courage to move forward towards obtaining a master's degree.
Secondly, we would like to thank our supervisor **Mr: Bekkari Fouad**, his valuable advice and help during the entire working period.
Our respects and gratitude also go to the members of the jury who did us the honor of judging this work.
We thank our teachers, KASDI MERBAH University OUARGLA for all the advice and encouragement we have benefited throughout this work.

Dedications

Finally, we do not forget the moral support provided by my family, especially **my Mather** and **my Sisters**; And God's mercy be upon you, my dear father.

And we thank all those who, from near or far, have contributed to the realization of this work. Thank you to my comrades and my best friends **NARIMAN** and **HADJER** and my fiance.

NESRINE

Dedications

Finally, we do not forget the moral support provided by my family, especially **my parents and my brothers**, and we thank all those who, from near or far, have contributed to the realization of this work.

Thank you to **my comrades and my best friends such as Feryal and Daouia and Khadidja .and my fiance Nadhir**.

IKRAM

Abstract

Searching for any information in any field usually makes use search engines such as google,the technique of this engines use information retrieval; which is retrieve the documents relevant for user query .For efficiency information retrieval done applies query expansion; which is adding words to initial query, has been implemented for clarity the initial query; there is two approaches global and local.

In our project we use co-occurrence for query expansion; for in order to find the best combination; this combination use large search space.In order to avoid of this space we use adaptive memory algorithm metaheuristic for find to get closer and better combinations and satisfy the user's query.

Key words: Information retrieval, query expansion, co-occurrence, metaheuristic, adaptive memory, parallelism.

الملخص

عادة ما يستخدم البحث عن أي معلومات في أي مجال محركات البحث ، تعتمد تقنية هذه المحركات استرجاع المعلومات ؛ وهو استرداد المستندات ذات الصلة باستعلام المستخدم. للحصول على كفاءة استرجاع المعلومات ، يتم تطبيق توسيع الاستعل ام ؛ الذي يضيف كلمات إلى الاستعلام الأولي ، يتم تنفيذه لتوضيح الاستعلام الأولي ؛ هناك مقاربات عديدة منها العالمي والمحلي .

في مشروعنا نستخدم التواجد المشترك لتوسيع الاستعلام التابع للمقاربة العالمية ؛ من أجل العثور على أفضل تركيبة ؛ تستخدم هذه المجموعة مساحة بحث كبيرة لتجنب هذه المساحة ، نستخدم خوارزمية الذاكرة التكيفية ميثاهوريستيك للعثور على تركيبات أقرب وأفضل وتلبية استعلام المستخدم.

الكلمات المفتاحية : استرجاع الملعومات ، تمديد الاستعلام ، مصفوفة التواجد المشترك، الميثاهوريستيك ، الذاكرة التكيفية، التوازي .

Résumé

La recherche de n'importe quelle information dans n'importe quel domaine fait généralement appel à des moteurs de recherche tels que google, la technique de ces moteurs utilise la recherche d'informations; qui récupère les documents pertinents pour la requête de l'utilisateur. Pour l'efficacité de la récupération des informations, applique l'expansion de la requête; qui ajoute des mots à la requête initiale, a été implémenté pour plus de clarté la requête initiale; il existe de nombreuses approches telles que l'approche globale et l'approche locale.

Dans notre projet, nous utilisons la cooccurrence pour l'expansion de la requête; pour afin de trouver la meilleure combinaison; cette combinaison utilise un grand espace de recherche. Afin d'éviter cet espace, nous utilisons une métaheuristique d'algorithme de mémoire adaptative pour trouver des combinaisons plus proches et meilleures et satisfaire la requête de l'utilisateur.

Mots clés: Recherche d'informations, développement de requêtes, co-occurrence, métaheuristique, mémoire adaptative, parallélisme.

Contents

List of Figures

List of Tables

General Introduction

We live in a time of speed, a time full of developments in various areas of daily life, especially in the field of technology, especially information technology, i.e. the multiplicity of information and knowledge, and this is difficult to reach in the absence of the World Wide Web ,this is through the use of strategies and techniques such as information retrieval strategy that it can meet the needs of users In the case of demand at the lowest costs and with the least effort.

Information retrieval is technique of extracting pertinent documents from large data collections;Information retrieval is a field of study centered on retrieving information in huge data sets from the internet, databases, and other sources.An IR system's job is to locate relevant information among a set of documents that corresponds to a user's query.In Information Retrieval,the process is composed of two main phases, indexing and search; the search process is carried out through several models, including the natural model the boolean model, and probability model; and vector space model which we will rely on in our study.

Additionally, it is possible to expand the queries to improve the efficiency of the retrieval This is known as query expansion, the main objective of query expansion, is to enhance the initial query by include more relevant terms. Terms can be added manually, automatically, or with user assistance. Query expansion approaches can be classified mainly into two approaches, namely the global approach which based on knowledge structures, and the second approach is local analysis which based on search results.

In our work, we will specialize in the global approach, and in particular with co-occurrence based query expansion which is advocated using co-occurrence statistics to find potential semantic relationships between search terms and document terms and to use these relationships to broaden the user's field of query.

A class of optimization algorithms known as metaheuristic algorithms is created to handle complicated optimization issues when more conventional approaches may be effective, evolution, and physical processes. The capacity of metaheuristics to search huge solution spaces using stochastic and probabilistic methods distinguishes them from other search algorithms.

There are several ways to apply metaheuristic algorithms to enhance information retrieval and query expansion. For example, they can be used to create new search words that are relevant to a user's query, cluster relevant search results into groups, and improve the ranking of search results. The application of metaheuristic algorithms in information retrieval and query expansion research has produced promising results. Such as genetic algorithms, simulated annealing, and adaptive memory.

In general, metaheuristic algorithms have demonstrated great potential to improve the speed and efficacy of information retrieval and query expansion methods. By utilizing the power of these methods, we may enhance the user experience in search engines and other information retrieval systems by more effectively extracting useful material from huge data sets.

With the use of numerous processors or cores, a problem is divided into smaller sub-problems using the parallelism technique. With this method, bigger problems that might not be possible to solve with a single processor can be solved more quickly.

Combining parallelism and metaheuristics can be an effective approach for resolving difficult optimization problems. Multiple answers can be produced simultaneously by using parallelism, which helps improve search space exploration and make better use of technological resources. By dividing up the search across several processors or cores, metaheuristics are additionally able to operate in parallel environments.

QE approaches based on adding best words but we add the best combination from all possible combinations extracted from a bag of words built from a data co-occurrence matrices. This approach results a huge potential search space, for this reason we propose to use a parallel metaheuristic witch is adaptive memory optimisation metaheiuristic. Our objective is to explore and exploit efficiency the search space searching for the closest and optimal solutions in record time .

In the first chapter, we defined information retrieval, and we mentioned query expansion and its approaches.

In the second chapter, we defined metaheuristic and its algorithm; and we explained how done parallelism of its.

In the third chapter, we mentioned the tools that we used in order to obtain the results that we wanted to reach.

CHAPTER 01

INFORMATION RETRIEVAL AND QUERY EXPANSION

Introduction:

With the increase of knowledge and the diversity of information in various fields, in addition to the human need to acquire some of this knowledges and information, its always resorts to search engines such as GOOGLE. and other engines, without which it is difficult to obtain and access this , which is obtained through the process of information retrieval, as it is interested and showing what suits the user's request,In order to make the retrieval more efficient and provide relevant information, been used query extension, which is done by several techniques, including the co-occurrence matrix.

In this chapter, we will know each of the information retrieval process, as well as mention its advantages, the steps to achieve it, and what we need to obtain better results through the query expansion.

1 Information retrieval definition (IR):

In the 1950s, the field of information retrieval (IR) was started. The field has greatly advanced in the last forty years. A wide range of users utilize various IR systems on a daily basis. Among a sizable number of documents, which are frequently stored on computers, IR locates textual documents that meet a user's information needs. When using their email and web search engines, millions of people today engage in IR[30].The science of information retrieval (IR) is traditionally empirical. The evaluation forums are good examples of how test collections with documents rated for relevance by human assessors have made it simple to conduct offline trials to improve retrieval efficacy[27].

IR is the technique of extracting pertinent documents from large data collections. Traditional information-retrieval techniques are unable to handle the task of producing high-quality search results due to the growing data collection and the increased need for retrieval results of the highest caliber. Ontology, a recently developed knowledge organizing system, is important in improving the function of information retrieval in knowledge management[46].

Information retrieval system(IRS):

Generally speaking, research in an IRS consists of comparing the representation internal representation of the query to the internal representations of the documents in the collection. There request is formulate by the user in a request language which may be the language natural, a keyword-based language or Boolean language. It will be transformed into an equivalent internal representation, during an interpretation process[1].

Due to imprecise and vague query wording, information retrieval (IR) deals with uncertain information. The main objective of an information retrieval system (IRS) is to sort documents according to how relevant they are to the user's query, separating relevant from non-relevant records in a pool of data[39].

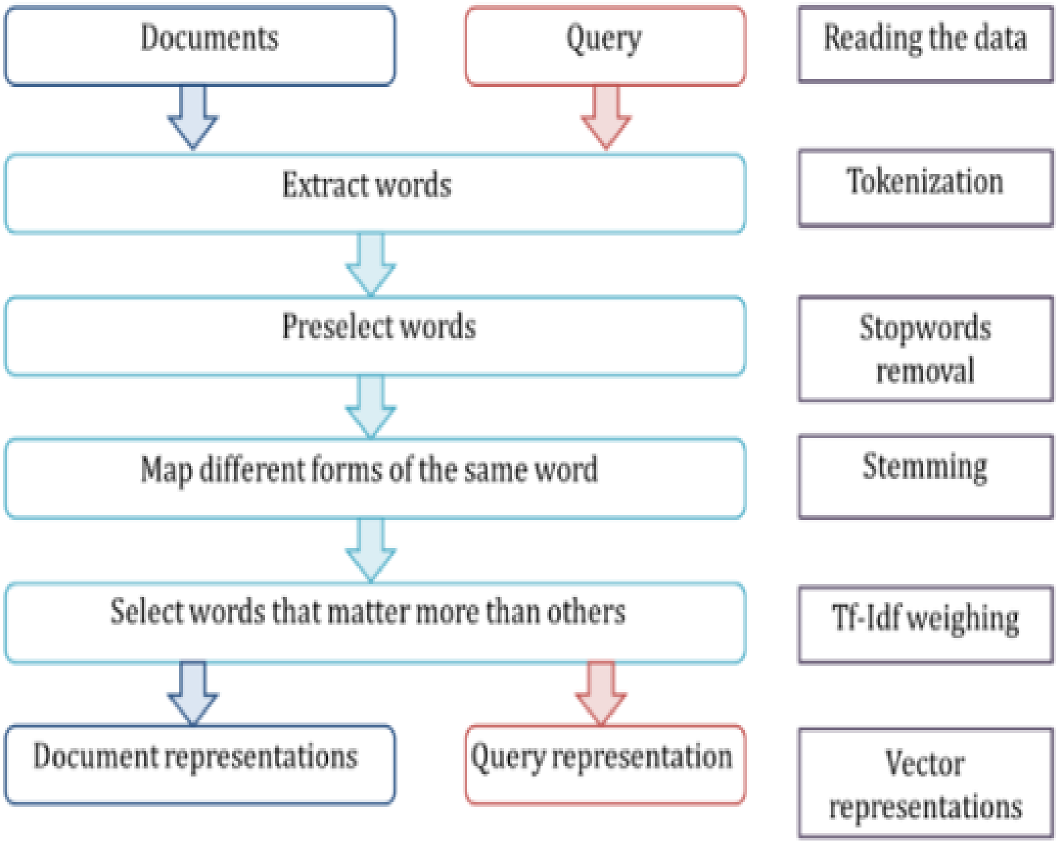


Figure 1: The IR process.

The IRS process involves converting a user's information from a list of actual citations into documents that are stored and contain information the user will find valuable. The IRS manages and stores the data on documents. The system assists users in locating the pertinent data they require; the IRS expressly does not return the data; instead, it returns the location and presence of the document that the data may have been found in. IRS does not include irrelevant documents and only includes the pertinent documents that satisfy the user information.

An IR system's job is to locate relevant information among a set of documents that corresponds to a user's query. This is a difficult task due to a number of factors: The document database normally contains unstructured information, texts are frequently written in free-flowing conversational language, and they frequently cover a wide range of topics. In certain situations, locating specific information[30].

1.1 IR system steps:

An IR system is a set of rules and procedures, for performing some or all of the following operations:

1. Indexing.
2. Searching.

1.2 Indexing:

By organizing and organising the data to make search and retrieval processes more effective, indexing plays a critical function in information retrieval systems. The general steps in the indexing process are as follows[28]:

1.2.1 Word extracting (Tokenization):

Tokenization the text to create smaller, more manageable pieces. Tokens are typically words, but depending on the indexing system, they can also be phrases, numerals, or other meaningful units.

1.2.2 Stop Word Removal:

Stop words—common, meaningless words—should be eliminated. Stop words include the words "and," "the," "in," and "is." Since they don't have any semantic meaning and would just clutter the index, these words are frequently left out of the indexing process.

1.2.3 Stemming:

To handle word ending changes, normalization words by taking them back to their most basic form. And changes words back to their root or dictionary form, whereas stemming entails taking frequent word ends off of words. Delete the repeated words in each document and then extract the words in all documents without repetition.

1.2.4 Inverted index:

Created a mapping between each token and the documents in which it appears. An inverted index is the most typical data structure used for indexing. Each token in an inverted index has a list of document references where it appears.

1.2.5 Weighting Term:

Each document's tokens should be given a weight to indicate its value or relevance. These weights can be determined using a variety of techniques, such as Term Frequency-Inverse Document Frequency (TF-IDF).

The TF-IDF takes into account a term's rarity across the entire document collection as well as its frequency in a given document. Its best-known formulation is[13]:

$$tf(t,d) = \log(1+n) \tag{1}$$

tf is the number of occurrences or the frequency **f** of the term **t** in the document **d** considered.

$$wTFIDF(t,d) = tf(t,d) * log\frac{N}{df(t)}. \tag{2}$$

df its documentary frequency, the number of documents in which it appears, **N** is the number of documents.

1.3 Searching models:

There are existing information retrieval model, such as the Boolean model, and Vector Space Model, and Probability Model,etc...

1.3.1 Boolean model:

The Boolean model was the first information retrieval model and is likely the one that has received the most criticism.

Boolean algebra is the theory set that it uses. It consists of three elements.(i.e., the NOT, the OR, and the AND) to form a query.

- It fails to order the list of documents that were found in the results.
- Each document is associated with a certain set of terms or keyword words[30].

1.3.2 Probability model:

Its application is based on four related principles: the related mind independence principle, the word independence principle, the literary relevance principle, and the probability ordering concept. The model creates a probability model for documents and queries based on probability theoryand uses the model to determine how similar documents and searches are.

The probability model, which is represented by the weight of the keywords, is based on the distribution of question keywords in related and irrelevant papers.

Documents are initiated in order based on how likely it is that they will be relevant given the user query. Documents and user queries are both represented by the binary vectors d and q[30].

1.3.3 Vector space model(VSM):

Is built on a set of rules that are used in pattern recognition and other domains to model text. The very abstract text is split, filtered, and classified by the VSM, which also applies statistics to the text's word frequency data. The computer processes the text in according to predetermined rules and transfers statistics to word frequency data[46].

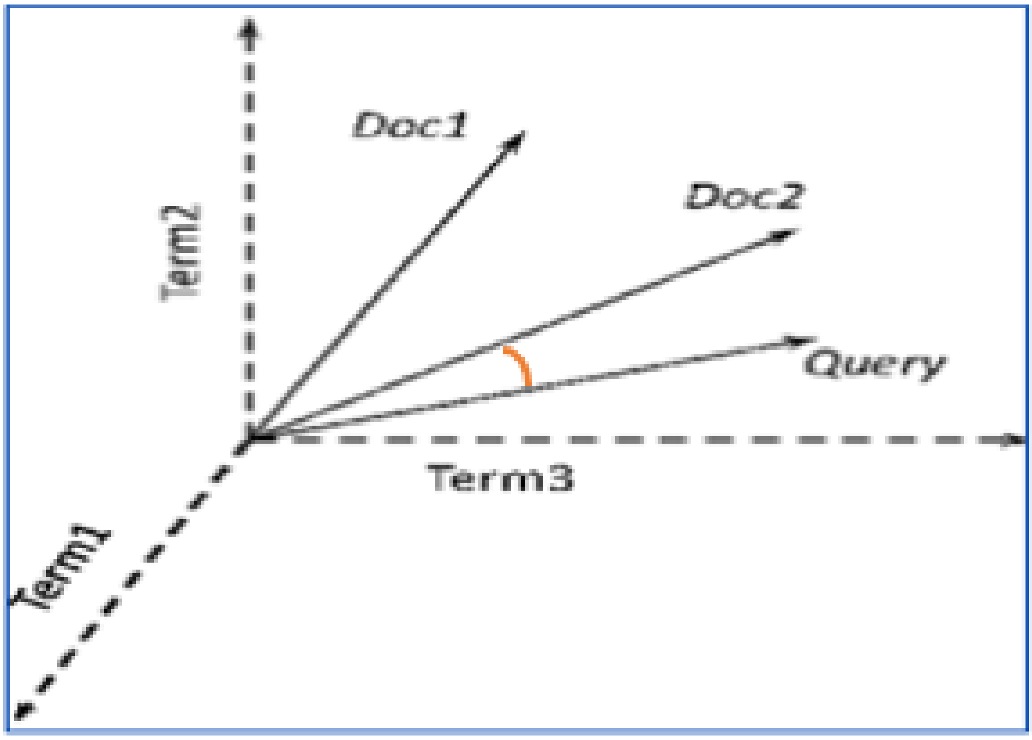


Figure 2: Vector space model

- **General Description of vector space model :**

  It aims to order documents in descending order of similarity to the user query. Users' queries and documents are each represented as a vector, and a cosine function is used to calculate the angle between the two vectors. The term-weighting scheme known as the tf-idf weighting has been introduced by the vector space model[30].

- **Advantages of vector space model:**
  - The benefits of vector data Compared to the raster data approach, vector data can more accurately depict topographic features. All feature types can be accurately represented in vector data models. When describing the position and dimensions of all topographic elements, points, lines, and polygons are reliable tools.
  - The vector space model is superior to the standard boolean model in the following ways: A simple linear algebraic model. Terms are not binary weights. allows for the continuous comparison of query and document similarity. enables ranking of materials based on potential relevance. partial matching is possible.

2 Query expansion for information retrieval:

In Information Retrieval is possible to expand the queries to improve the efficiency of the retrieval.

Expanding the query aims to access documents related to the query entered by the user.

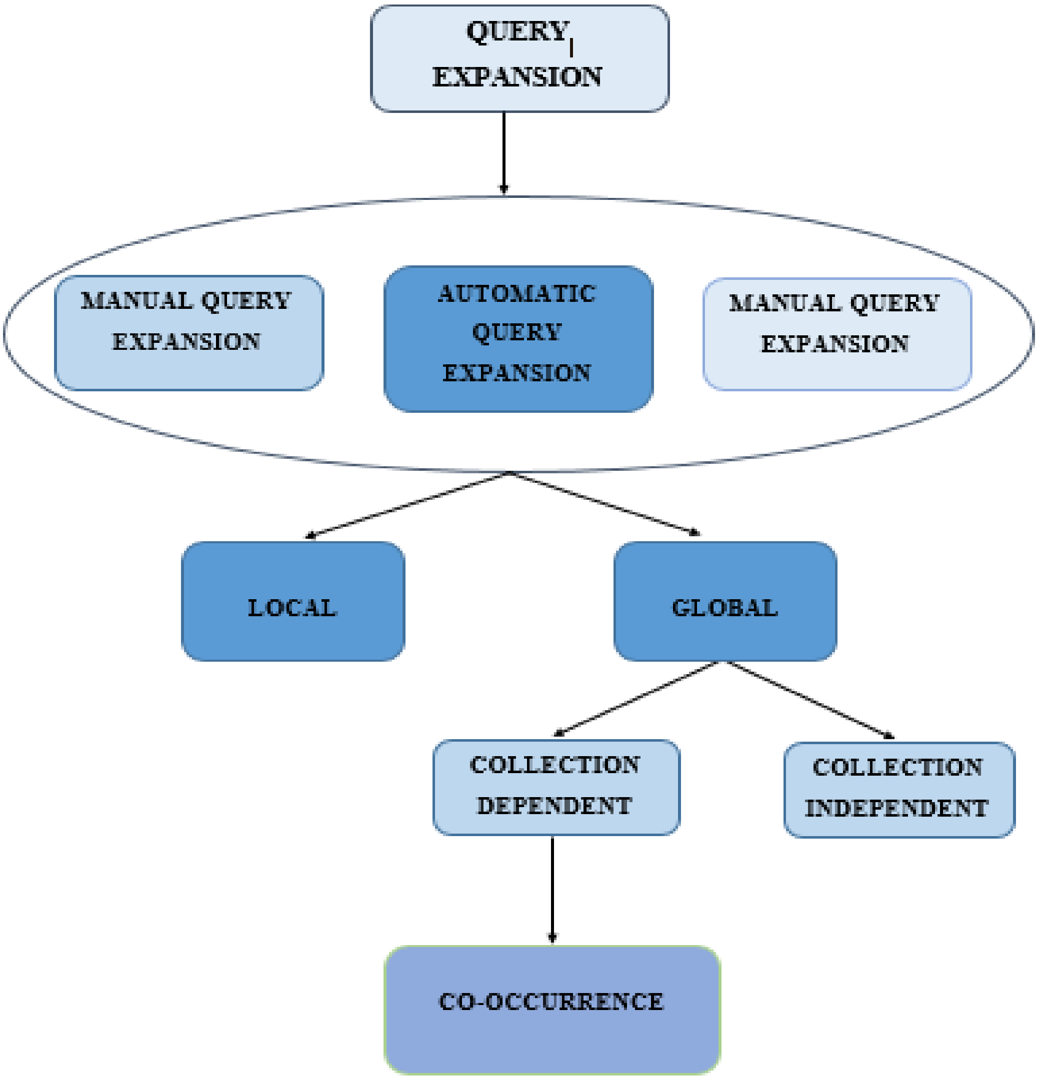
2.1 Query expansion process definition (QE):

In literature, query expansion has a long history. It was first used as a method for literature indexing and searching in an automated library system by Moron and Kuhns in 1960. Through "relevance feedback" and its characterisation in a vector space model, Rocchio was the one who first made QE popular. The goal of relevance feedback is to improve the end result by incorporating user feedback into the retrieval process. By rating the results' relevancy to the original query, the user specifically provides feedback on the documents that were found. Further developments and applications of Rocchio's work include cluster-based information retrieval and collection-based term co-occurrence[5].

One of the best methods for addressing term scarcity, which is common in web search queries, is query expansion (QE). In recent years, a number of semantic QE strategies, including linguistic and ontology-based, have been put forth to overcome the vocabulary mismatch problem.

Although studies, such as those by Verberne et Al., have looked into query expansion in the context of professional search, there aren't lots of commercial systems actually being used. This could be partly attributed to the difficulties posed by the inquiries' inherent structure. Providing only recommended terms as additions or complete replacements is acceptable. Instead, term suggestions should be both pertinent to and distinctive to each sub expression they include[24].

Using query expansion, the user is helped to create queries that allow for the retrieval of valuable results. The main objective of query expansion, also referred to as query augmentation, is to enhance the initial query by include more relevant terms. Terms can be added manually, automatically, or with user assistance[24].



5

Figure 3: Approaches of Query Expansion.

A search engine will add additional search terms to a User Weighted query using the query expansion procedure. There are three types of query expansion techniques: manual, interactive, and automatic. Manual query expansion refers to techniques the user can employ to change the query on their own, independent of the system. When a system expands a query that was interactive, the user has some input into that process. The term "automatic expansion" describes techniques that change a query without the user's input.

2.1.1 Automatic query expansion(AQE):

The relative inefficiency of information retrieval systems is mostly a result of how inaccurately a query made up of a few keywords represents the actual information needed by the user. Automatic query expansion (AQE), which adds new features with a similar meaning to the user's original query, is one well-known method to get around this limitation.

Although AQE has a lengthy history in the information retrieval field, it has only recently attained a level of experimental and scientific maturity, especially in lab settings like TREC. This study provides a comprehensive overview of numerous contemporary methods for AQE that make use of various data sources and use quite distinct ideas and techniques[18].

2.2 Query expansion steps:

2.2.1 Extracting:

Read the documents and queries then remove stop word and tokenization and stemming.

2.2.2 Ranking:

Ranking the terms according the weight of each term.

2.2.3 Selecting:

Selecting some number of terms which retrieve a good results.

2.3 Query expansion approaches:

In the existing state-of-the-art query expansion approaches can be classified mainly into two classes: global analysis and local analysis.

2.3.1 Local approach (Based on Search Results):

The process of expanding a query by analysis of the query itself, the user's context, or the local document collection is known as local analysis in automatic query expansion. This strategy stresses the utilization of information that is accessible and pertinent to the particular query being expanded

Only a small part of the initially retrieved documents are used in local analysis to extend queries.

Expansion terms are typically taken out of the relevant documentation. If users provide sufficient accurate relevance judgements[22].

Multiple techniques can be used in local analysis to expand the query. Typical techniques include:

- **Proximity-based expansion:** Expanding the reach of the query by taking into account terms that are nearby to the initial query terms within the document collection. It is possible to add terms as expansions that appear close to the search terms in pertinent documents.

- **Feedback algorithms:**[10] The local analysis' feedback technique enables users to give clear input on the practicality of documents and helps in the discovery of terms that are expected to improve retrieval efficiency. Automatic query expansion can improve the retrieval effectiveness by more correctly capturing the user's information needs by incorporating user feedback and repeatedly improving the query.

Feedback algorithms are essential for exploiting user feedback to improve query representation in the context of local analysis for automatic query extension.

- **Pseudo-Relevance Feedback(PRF)**: The local analytic strategy for query expansion incorporates PRF (Pseudo-Relevance Feedback) algorithms. By utilizing the relevance evaluations of the documents returned in response to an original query, PRF tries to improve the retrieval efficiency of a search engine.The PRF (Pseudo-Relevance input) technique expands the initial query by using input from highly ranked documents in the context of automatic query expansion. In local analysis, it is frequently employed[36].

The process of expanding a query by analysis of the query itself, the user's context, or the local document collection is known as local analysis in automatic query expansion. This strategy stresses the utilization of information that is accessible and pertinent to the particular query being expanded.

Expansion terms are typically taken out of the relevant documentation. If users provide sufficient accurate relevance judgements.

This strategy stresses the utilization of information that is accessible and pertinent to the particular query being expanded, available within the local context[22].

2.3.2 Global approach (Based on Knowledge Structures):

A disciplinary knowledge structure is a hierarchical knowledge system made up of knowledge units (elements) and the interactions between them that are found within a particular field. Discipline knowledge structure analysis is covered in two main approaches in the current research. First, subject-matter specialists qualitatively explain knowledge structures in light of their personal professional expertise and research background. Second, to uncover interaction links between information items in textual data, disciplinary knowledge structures are quantitatively examined based on bibliometric techniques[12].

In order to improve the retrieval for relevant data by expanding user queries, global analysis based on knowledge structures for automatic query expansion approaches includes a thorough examination of techniques and methodologies that make use of structured knowledge sources, such as anthologies, knowledge graphs, or semantic networks. The goal of this analysis is to comprehend the advantages, disadvantages, and potential upgrades of knowledge structure-based query expansion methods[37].

Measures of semantic similarity are essential for determining how closely query terms and knowledge structure ideas are connected. The accuracy of query expansion can be considerably impacted by the measure selection. To give more accurate expansion results, more contextual information, such as the user's query context or the retrieval context, might be included.

In the end, a comprehensive examination of query expansion strategies based on knowledge structures seeks to promote knowledge of and the creation of methods that make use of structured knowledge sources to improve information retrieval.

Adding words that are synonyms or linked to the original query is another method of query expansion. By doing this, information can be improved by utilizing the knowledge kept in a thesaurus or other (global) information source.

Thesauri are frequently used as a technique in information retrieval systems to identify synonymous terms and linguistic entities that are formally distinct but semantically similar. As the thesaurus has been the focus of research for almost 40 years, several different approaches have been put forth.

A query vector is created by combining word vectors connected to the query's words. To identify the pertinent documents, the query and document vectors are contrasted. To create factor vectors, the query vector can be split up into a number of factor clusters. The ranking of the documents within the factor cluster is then determined by comparing the factor vectors to the document vectors.

Information retrieval systems' methods and procedures for expanding a user's query are referred to as global analysis of query expansion approaches. These methods take the complete collection of documents into account and search out new words or ideas that can enhance the effectiveness of retrieval.

Global approach concentrates on examining a larger number of documents or outside sources to find pertinent expansion terms.

The goal of a global analysis approach based on query expansion approach knowledge structures is to improve the efficiency of information retrieval systems by studying and utilizing the knowledge structures of different query expansion approaches[37].

By expanding user inquiries with new phrases or concepts that are probably to convey the intended meaning more thoroughly, this global analysis approach seeks to increase the accuracy and relevancy of search results. It aims to overcome the constraints and biases present in individual query expansion techniques, resulting in more robust and contextually aware information retrieval systems. In query expansion methods, taking into account more aspects than only the query's and its linked documents' immediate context is known as global analysis. It concentrates on examining a larger number of documents or outside sources to find pertinent expansion terms. Here are some essential components and methods for query expansion in global analysis[37]:

- **Thesaurus-based Approaches:** For automatic query expansion, thesauri or restricted vocabularies can be used. These sources offer organized collections of terms and the connections between them in terms of meaning. Thesaurus-based methods find additional terms that are connected to the concepts by mapping the initial query terms to the concepts in the thesaurus. The query is then expanded to include all of these related terms.

- **Knowledge Graphs:** Knowledge graphs, like those seen on Wikipedia or DBpedia, include organized data on entities and the connections between them. These knowledge graphs can be used to find related things or concepts and perform automatic query expansion. The entities associated to the initial query words in the knowledge graph can be thought of as the query's expansion terms.

- **Latent Semantic Analysis (LSA):**is a method for finding latent semantic connections between phrases and documents by applying matrix factorization. The construction of a term-document matrix and its singular value decomposition (SVD) are required for automatic query expansion using LSA. The discovery of semantically connected terms in the resulting latent semantic space enables the query to be expanded.

- **Co-occurrence Analysis:** This approach of global analysis examines the patterns of word co-occurrence in the document collection. It recognizes frequently used terms in combination and takes them into account as possible expansion terms. For instance, the system could expand the query to include ”data mining” if the terms ”machine learning” and ”data mining” regularly appear together.

These are a few common approaches for employing global analysis to automatically expand queries. Every technique has its own advantages and disadvantages, and the success of each depends on the particular information retrieval task and dataset. To improve the automatic query expansion process, researchers are always investigating and creating novel approaches.

2.4 Co-occurrence based on query expansion:

Since the 1990s, the idea of term co-occurrence has been utilized to pinpoint some of the semantic connections between terms found in text documents. Rijsbergen (Rijsbergen, 1977) proposed the use of co-occurrence statistics to identify some sort of semantic relationships between query and document terms and to take advantage of them to expand the user's inquiries. In actuality, the premise of the idea is that "if an index term is good at discriminating relevant from non-relevant documents, then any closely associated index term is likely to be good at this."

A method used in information retrieval to increase the relevancy of search results is co-occurrence-based query expansion.

The automatic method used by Cosine and Tanimoto has improved the co-occurrence based query expansion methodology even more. This technique expands queries from users by using the co-occurrence frequency of terms in a document collection. Within a document collection, the cosine similarity and Tanimoto coefficient of two terms are calculated to determine how similar they are to one another. This strategy helps users find more relevant documents and improves the precision and recall of search results by expanding user searches with related terms[42].

Following are some well-known co-occurrence coefficient measuring methods:

$$Tanimoto(ti,tj) = \frac{cij}{ci + cjcij} \tag{3}$$

$$Dice(ti,tj) = \frac{2cij}{ci + cj} \tag{4}$$

$$Cosine(ti,tj)\frac{cij}{\sqrt{cicj}} \tag{5}$$

where ci and cj represent the relative numbers of documents in which terms ti and tj occur, and ci,j represents the total number of documents in which both part.

These coefficients are used to assess how closely related the terms the vectors represent are. The results is a rating of potential expansion terms, with the best terms at the top[43].

The accuracy of the co-occurrence based query expansion strategy has been regularly researched, and cutting-edge techniques like the automatic Dice, Cosine, and Tanimoto approach are being used. These methods are vital to improving search results' relevancy and optimizing the user experience in information retrieval generally.

In the selection method, the equation is used to choose the most likely terms.

$$rel(q, te) = \sum_{ti \in q} qiCC(ti, te) \tag{6}$$

where CC is one of the following: Tanimoto, Dice, or Cosine co-occurrence coefficient.

In order to find indexing terms that are similar to those that have been stated in a user query, term co-occurrence data has been widely employed in document retrieval systems. These similar terms can then be used to supplement the initial query statement. The retrieval effectiveness of the enlarged questions is frequently no better than or even worse than the retrieval effectiveness of the unexpanded queries, despite the plausibility of this technique to query expansion[33].

Co-occurrence is the general phenomena when words are frequently employed in the same context.

Another way to define co-occurrence is when one word in a text suggest the existence of another.Co-occurrence and collocation have certain characteristics in common, yet they stay distinct from one another.

In order to expand queries and improve search results, the co-occurrence technique is employed in natural language processing and information retrieval. Utilizing a co-occurrence based query expansion strategy has major disadvantages though. like as because we employ a large text on a dataset, co-occurrence matrices can be very massive and sparse. As a result, the matrix may end up having a lot of zeros, which may limit the expansion's utility[45].

The co-occurrence approach can be computationally expensive in large-scale information retrieval systems and may not be extensible to handle vast amounts of data[32].

By adding relevant terms to the original query, co-occurrence-based query expansion aims to increase the precision and comprehensiveness of search results. The search engine can obtain more pertinent content by figuring out and including related phrases in the original query, which improves search accuracy and user.

According to Figure 4, a term is chosen as an expansion term if it appeared even once in the top ten documents returned by the search engine alongside a query term. The terms are then listed in decreasing order. The final step is to choose the terms that apply to the entire question in order to get eliminate of polysemous terms. Characteristics of ambiguity[9].

1	1	2	3	4
1	2	3	4	1
2	2	3	3	3
4	3	3	2	1
3	3	3	4	4

(a)

	1	2	3	4
1	1	2	0	0
2	1	1	3	0
3	0	1	5	3
4	1	0	1	1

(b)

	1	2	3	4
1	0.0625	0.125	0	0
2	0.0625	0.0625	0.1875	0
3	0	0.0625	0.3125	0.1875
4	0.0625	0	0.0625	0.0625

(c)

Figure 4: Example of co-occurrence matrix

Conclusion:

QE approaches based on adding best words but we add the best combination from all possible combinations extracted from a bag of words built from a data co-occurrence matrices.

QE approaches based on adding best words but we add the best combination from all possible combinations extracted from a bag of words built from a data co-occurrence matrices.

We will try to solve these problems through the use of metaheursticc algorithms, and this is what we learn about in the next chapter.

CHAPTER 02

THE METAHEURISTICS AND THE PARALLELISM

Introduction:

The utilization of metaheuristic optimization techniques allows for the solution of complex problems that traditional methods are unable to address. They are frequently used in conjunction with traditional optimization techniques to improve their effectiveness and produce better results.

The idea of carrying out several activities or calculations at once is known as parallelism. The use of parallelism can be used to speed up the execution of metaheuristic algorithms in the context of optimization and problem-solving. The total computational time can be greatly decreased by breaking the problem into smaller subproblems and processing them concurrently on various processors or computing resources.

In this chapter we know metaheuristic and classiffication of metaheurstic,then we know parallelism of its .

1 Metaheuristic:

1.1 Definition

In order to discover optimal or nearly optimal solutions to a certain optimization problem, a metaheuristic is an iterative technique that is independent of the problem at hand. Large and complex solution spaces can be effectively searched using metaheuristics, and they can also avoid local optima. They can be used to solve a variety of optimization problems, including combinatorial optimization, numerical optimization, and constraint satisfaction issues. Typically, they are motivated by natural phenomena like evolution, swarm behavior, or physical annealing processes.

When exact optimization methods are neither practicable or practical or when the problem at hand is too complex to be modeled conventionally, metaheuristics are frequently applied[41].

Metaheuristics can be used to a range of optimization problems, including combinatorial, continuous, and mixed-integer optimization. They are inspired by natural phenomena including evolution, swarm intelligence, and physical processes.

Metaheuristics can identify high-quality solutions to optimization problems that are normally difficult or impossible to solve using exact methods by cleverly and effectively navigating the search area.[34].

2 Classification of metaheuristic:

There are several ways to classify metaheuristics according the number of solutions[2]:

2.1 Metaheuristic algorithms based on a single solution:

These techniques begin their optimization process with a single solution, and update it as they move through iterations. It could result in trapping into local optima and also only partially explores the search space[44].

2.2 Population (multiple) solution based metaheuristic algorithms:

These algorithms start their optimization process by generating a population of solutions. With each generation or iteration, the population of solutions changes. The algorithms help prevent local optima since they have a great search space exploration and various solutions that work together to help one another. Additionally, they possess the ability to jump to the promising area of the search space. Therefore, the majority of problems in the real world are solved using population-based algorithms[44].

The metaheuristic algorithms can be categorized into four categories based on their behavior: evolution-based, swarm intelligence-based, physics-based, and human-related algorithms:

2.3 Algorithms based on evolution :

It takes inspiration from natural evolution and begins with a population of solutions that are generated at random. The best solutions are combined in these algorithms to produce new people. Mutation, crossover, and the best option are used to create the new individuals.

2.4 Algorithms based on swarm intelligence:

These algorithms are motivated by the social behaviors of insects, animals, fish, birds, etc. Particle Swarm Optimization (PSO), created by Kennedy and Eberhart, is the most widely used method. It takes its cues from a flock of birds flying about a search area in quest of the optimal spot (position). Swarm intelligence techniques include ant colony optimization, honey bee swarm optimization, monkey optimization, and others[3].

2.5 Algorithms that are based on physics:

These are motivated by the laws of physics that govern the universe. Physics-based algorithms include Harmony search, Simulated annealing, and others[19].

2.6 Algorithms that are affected by human behavior:

These methods were only influenced by human behavior. Every human has a unique manner of carrying out tasks that has an impact on how well they succeed. It encourages scientists to create the algorithms. Popular algorithms include the League Championship algorithm and the Teaching learning-based optimization algorithm (TLBO)[26].

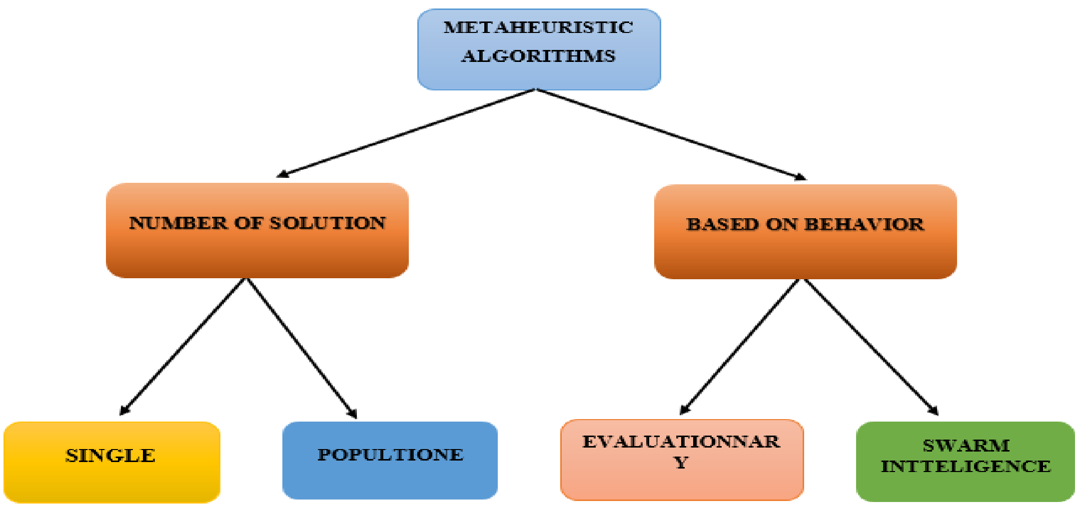


Figure 5: Classification of metaheuristic algorithms.

A metaheuristic algorithm's can also be categorized to : evaluation method and use of local search :

2.7 Evaluationary methods:

The process for developing a metaheuristic algorithm must include evaluation. It entails evaluating these algorithms' performance to ascertain how well they handle challenging issues. Understanding the advantages and disadvantages of various metaheuristic algorithms through evaluation enables researchers and practitioners to build new and better algorithms. The effectiveness of metaheuristic algorithms is assessed using a variety of metrics, such as solution quality, convergence speed, and robustness. Convergence speed evaluates how quickly the algorithm finds a solution, while solution quality describes how near the algorithm is to reaching the ideal answer. The algorithm's robustness is measured by how well it performs even when the problem or its parameters change[6].

2.7.1 Scatter search:

In contrast to other evolutionary techniques like genetic algorithms, scatter search is based on the idea that systematic approaches to problem-solving offer important advantages over relying just on randomness. It makes use of search diversification and intensification techniques that have worked well for a number of optimization issues.

Its Very computational and the time-consuming and costly computationally. Scatter search may not be able to considerably enhance a poor starting solution.

2.7.2 Genetic algorithms :

The basic idea of genetic algorithms is to simulate the process of evolution of sexual species. One of the theories of evolution is that two fundamental mechanisms, during sexual reproduction, make it possible to create a new individual different from its parents. The first is the crossing of the latter, which consists of combining two halves of the genetic heritage of each of the parents to constitute the genetic heritage of the child.

The performance of genetic algorithms is greatly influenced by the selection of many parameters, including population size, crossover and mutation rates, and selection strategies. Finding the ideal values for these characteristics might be difficult.

The genetic algorithm, like many other metaheuristic algorithms, may become stuck in local optimum and fail to locate the global optimum.

Complex problems can be slow to solve: Since a genetic algorithm must evaluate the objective function numerous times, it might be slow for complex problems with a wide search area[21].

2.7.3 Ant colony algorithms:

Ants have the particularity of using volatile substances called pheromones to communicate. They are very sensitive to these substances, which they perceive thanks to receptors located in their antennae. These substances are numerous and vary according to the species.

Ants can deposit pheromones on the ground, thanks to a gland located in their abdomen,, which can be followed by their congeners.

Ant colony algorithm performance may suffer as problem size increases. Ant colony algorithm performance can be significantly impacted by the quality of the initial solution

Ant colony algorithm performance may suffer as problem size increases.

Its performance can be significantly impacted by the quality of the initial solution[17].

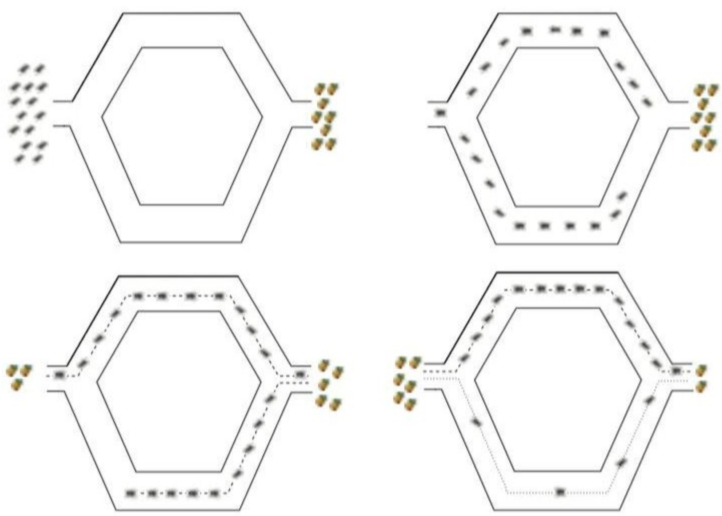


Figure 6: the influence of experience on the choice of ants.

2.8 Local search methods :

Local search methods are a type of metaheuristic algorithms that solve problems by iteratively examining the area near the existing solution. Until a good solution is identified, the plan is to make small modifications to the existing solution and evaluate the new one.

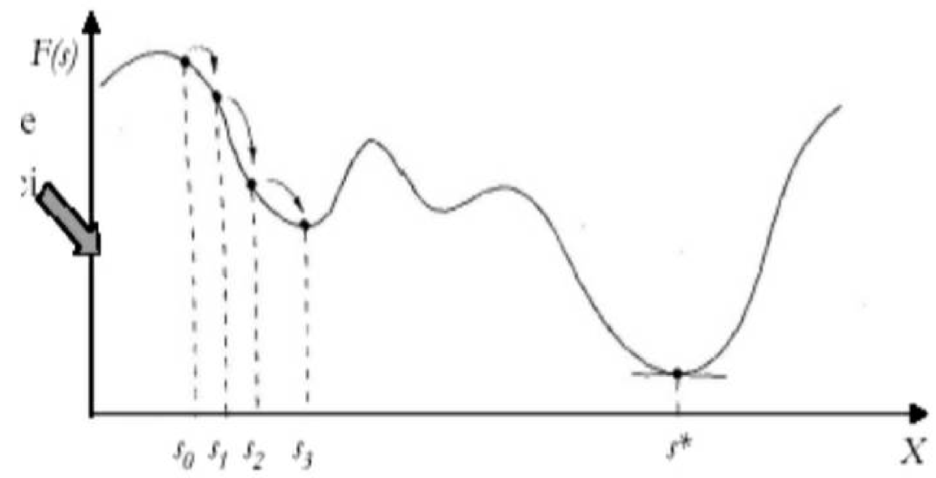


Figure 7: Example of local search of exploration of a solution .

These local search techniques are commonly used in metaheuristics.

Local search tries to intensify by trying to replace the present candidate solution with a neighbor that has a better cost function. The neighbourhood structure is crucial to the effectiveness of a local search algorithm[35].

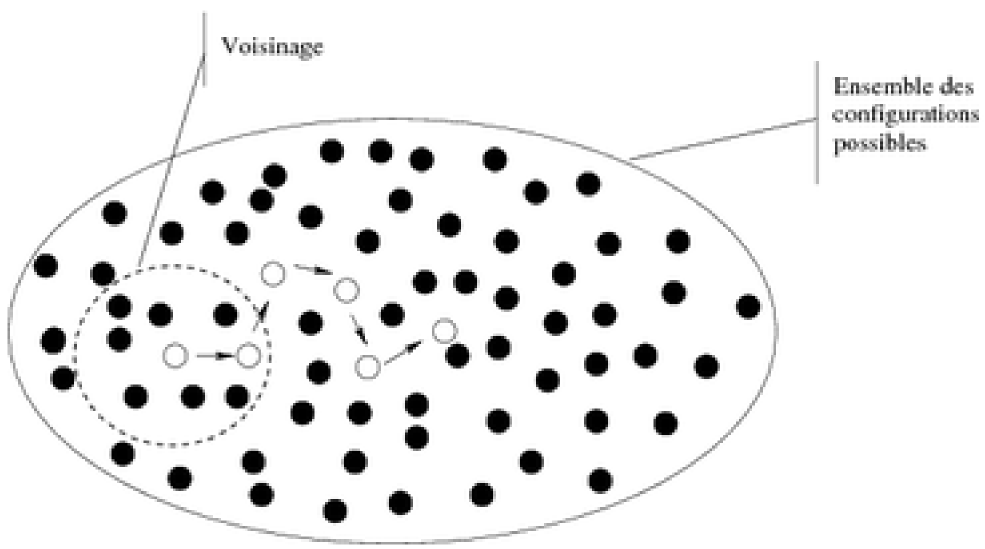


Figure 8: Representation of the general local search procedure.

Figure 8 illustrates the operation of the general algorithm for local research.

In an abstract way, local research can be summarized as follows:

1. Start with a solution.

2. improve the solution

3. Repeat the process until the stop criteria are met.

These local search techniques are commonly used in metaheuristics:

2.8.1 Simulated annealing:

Simulated annealing is based on an analogy between a physical process (annealing) and the optimization problem. Simulated annealing as a metaheuristic is indeed based on work aimed at simulating the evolution of a solid towards its minimum energy state. The classic description of simulated annealing presents it as a probabilistic algorithm, where a point evolves on the search space.

Needs an excellent starting solution: The success of simulated annealing is greatly influenced by the quality of the first solution. It could take a while for the algorithm to converge to a viable solution if the original solution is far from the global optimum.

Indeed, the method is based on the Metropolis algorithm (or Metropolis-Hastings method, named after the author of the generalization of the method)[23].

2.8.2 Research with tabu :

A local heuristic search technique using Tabu Search is guided to explore the solution space beyond local optimally. The use of adaptive memory by Tabu Search, which results in a more adaptable search behavior, is one of its key features.

As a result, memory-based tactics are a defining characteristic of tabu search approaches, which are built on the search for ”integrating principles,” which allow for the proper fusion of various types of memory with efficient methods for utilizing them. A unique discovery is that these principles occasionally have enough power to produce efficient problem-solving behavior on their own, with little reliance on memory, is based on the idea that a poor strategic decision will frequently provide more information than a wise random one.

Instead of being limited to imitating the processes seen in lower tiers of natural events and behavior, the tabu search's emphasis on adaptable memory makes it possible to leverage the kinds of methods that underlie the best of human problem-solving. A few key characteristics of tabu search's fundamental components are listed below.

The goal of tabu search is to establish linked principles that can strengthen the basis of intelligent search as well as new and more efficient ways to utilize the concepts embodied[20].

2.8.3 Adaptive memory:

Rochat and Taillard (1995) initially proposed the adaptive memory programming as a technique to improve the TS to resolve the VRP. The primary concept behind the AMP is the capability of assembling an excellent solution out of various parts of other good solutions. similar to how two children are produced from two parents in genetic research Goldberg (1989)[29].

The memory is initialized through an informal TS during phase 1 of the created algorithm, in contrast to the standard AMP, where memory initialization is completed beforehand. During this phase, good solutions are saved and sorted according to their score. After that, a new solution is developed using the data in the memory and improved by the tabu search throughout each iteration of the second phase.

The memory is updated with the best answer discovered. The specifics are provided below:

- **Step 1:** Initializing memory with a loose local search. After that, the route memory is updated with all of the routes for the various solutions.

- '**Step2:** Components of the route memory are combined to create a new solution. A probabilistic principle is used to choose a route, favoring the one with the highest score. Up until there are no more routes in the memory, the process is repeated. The CW algorithm is used to complete the new solution for the remaining consumers.

- **Step 3:** Memory update by storing all of the optimal routes identified by the TS, assuming they aren't already in the memory pool.

It is based on three stages:[7]

1. **Information :** Corresponds to the representation of information. It is modelled in the form of a memory. The goal is to memorize all or some of the solutions generated by the research.

2. **Intensifying :** Corresponds to the exploitation of information. Research is being deepened at the local level in an attempt to improve the relevance of available information. Its strategy is to:

   - The best solutions are stored. The common properties are revealed. Research is directed towards the regions thus defined.

3. **Diversification :** Matches the search for new information. Research is deepened across the entire research space (global level) to increase the amount of this information, exploring new regions (global level). Its strategy is to:

   - Memorize the most visited solutions and impose a system of penalties.

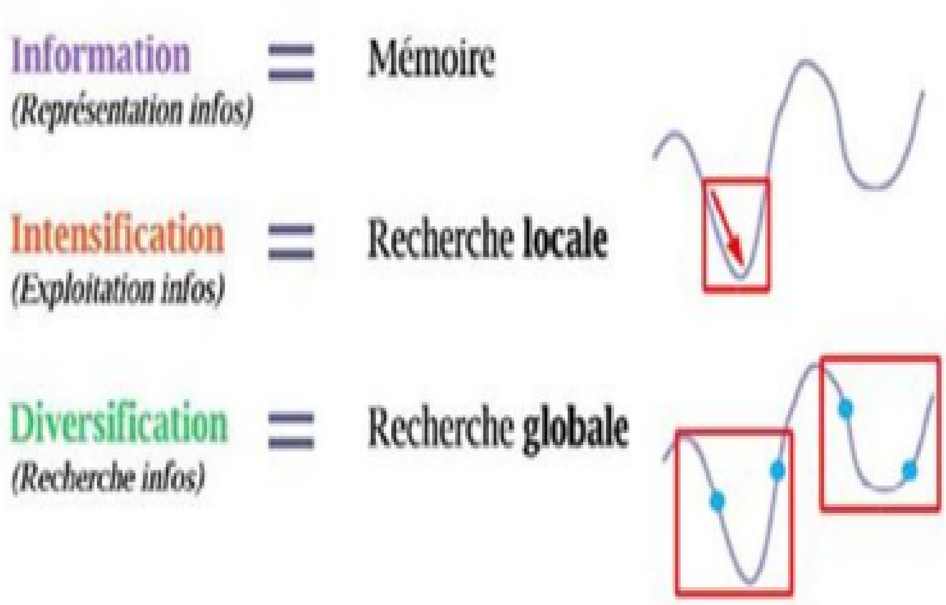


Figure 9: The steps of adaptive memory research method.

The operating principle of the adaptive memory method:

This method works with a central memory responsible for storing the components of the best solutions encountered. These components are combined to create new solutions. If the combination does not produce an eligible solution, a repair process is implemented. A Local Search algorithm is then applied and the resulting solution components are considered to be part of the central memory.

At the beginning of the research, the central memory contains components from a wide variety of solutions, so the process of combining will tend to create a variety of new solutions. As the research progresses, the central memory will tend to store only the components of a very limited set of solutions. Research is gradually becoming a process of intensification[7].

Adaptive memory programming:

Adaptive memory programming:

1. Initialize memory.
2. Repeat, as long as a stopping criterion is not satisfied:
   (a) Generate a provisional solution from the information contained in the memory.
   (b) Improve using local search to get a solution
   (c) Update the memory by incorporating the information elements contained.

The ability of metaheuristic algorithms to explore, take advantage of, and learn from previous encounters is strengthened by their adaptive memory, which helps them avoid local optima and provide better solutions by directing their search towards promising areas of the solution space.

3 The parallelism:

Parallel computing can be defined as the simultaneous use of multiple computing resources (a single computer with multiple processors, an arbitrary number of computers connected by a network or an arbitrary number of computers connected by a network) to solve a computer problem, and one can imagine a program that runs using multiple central processing units. A problem decomposed into discrete parts that can be solved concurrently each part is a series of instructions. These instructions execute simultaneously on different central processing units[7]. .

3.1 Motivations:

- Processor speed will not continue to increase due to physical constraints (data transfer speed cannot exceed the speed of light).
- Parallel computing can be:
  - the only solution to process certain large calculations in a given time.
  - the simplest or least expensive solution to process certain calculations in a given time.
- Parallel computing provides a solution for problems requiring high fault tolerance.
- The universe is highly parallel.

3.2 Commonly used parallel programming models:

There are several parallel programming models in common use citing:

- The Parallel Data model.
- The shared memory programming model.
- The message passing model.

3.2.1 Data parallelism:

This model of parallelism is used to exploit the concurrency that occurs from applying the same flow of instructions to different elements of a data structure. For example, adding 2 to all items in a row, or salary increment for all employees with 5 years of service. Data parallelism programs are characterized by the following characteristics:

- each operation on each piece of information can be considered as an independent task.
- The normal granularity of a data-parallel computation is small.
- The concept of data locality does not manifest naturally.

It was noted that data parallelism compilers often require the programmer to provide information about how the data should be distributed across processors, i.e. of how the data should be divided on the tasks. Then the compiler can translate the parallel data program into SIMD form, and produce a communication code automatically[11].

3.2.2 Shared memory:

In the shared memory programming model tasks share a common address space, reading and writing is done asynchronously, access control to shared memory is ensured with various mechanisms such as locks and semaphores. An advantage of this model from the programmer's point of view, for him there is no need to explicitly indicate how to communicate data from producers to consumers, this is because of the absence of the notion of ownership of data. However understanding and maintaining data locality is more difficult, therefore it is more difficult to write deterministic programs.

Another model is the virtual shared memory model (Virtual Shared Memory VSM) and in application a Shared Memory programming model in a distributed memory environment[40].

Distributed Shared Memory (DSM) is an extension of the shared memory programming model on systems that do not have physically shared memory. Access is done using the usual read and write operations. Unlike message passing, in a DSM system a process that wants to perform operations on some data does not need to know its location; the system will search and find them automatically.

In most DSM systems, shared data can be replicated to increase parallelism and application efficiency.

While scalable parallel machines are mostly based on distributed memory, many users may find it easier to write parallel programs using a shared-memory programming model. This makes DSM a very promising model, if it can be implemented effectively [4].

3.2.3 Message passing

Message overflow is probably the most widely used model of parallel programming today. Message-passing programs create multiple tasks, each task encapsulates its local data and each task is identified by a unique name, and the tasks interact by sending and receiving messages to and from called tasks.

Theoretically this model allows the dynamic creation of tasks, the execution of several tasks by a processor, or the execution of different programs by different tasks. However, in practice most message passing systems create a fixed number of identical tasks at program startup and do not allow tasks to be created or destroyed during program execution, P2PMPI is a typical example .

These systems are said to implement a single program multiple data (SPMD) programming model because each task executes the same program on different data. This model is sufficient for a range of parallel programming problems but hinders some parallel algorithm development .

From a programming point of view, this model embeds this form of subroutine libraries. These libraries provide routines for initializing and configuring the messaging environment as well as sending and receiving data packets. Currently, two high-level message-passing libraries are most popular for scientific and industrial applications[38]:

- The PVM (parallel virtual machine) of the Oak Ridge National Laboratory
- MPI (message passing interface) defined by the MPI forum.

Another classification is based on the type of memory, there are generally two classes shared memory machines and distributed memory machines.

..

3.3 Parallelism of metaheuristics dfinition:

Parallel computing can be defined as the simultaneous use of multiple computing resources.

Metaheuristics are simply algorithms from which we might derive functional or data parallelism from a computational perspective. However, many meta-heuristics are hampered by a shortage of data and functional parallelism. For example, the local search loop of the generic tabu search exhibits significant data dependencies between iterations, especially when the tabu criterion is used and when memory and the tabu status are updated.

Similar to how moving from one generation to the next occurs in traditional evolutionary approaches, replacing the current solution of the generic simulated annealing operation cannot be done in parallel, necessitating the sequential execution of the inner loop[14].

4 Classification on the parallelism of metaheuristic:

According to the source of parallelism used, they divided the parallelization techniques used with metaheuristics. [14]:

- **Type 1:**This parallelism source is typically detected through a heuristic approach iteration. Moves are analyzed in parallel or moves' limited functional or data parallelism is taken advantage of.

This technique, also known as low-level parallelism, is rather simple and concentrates only on speeding up computations; it makes no attempt to improve exploration (unless when the parallel process is given the same total wall-clock time as the sequential method) or solution quality.

- **Type 2:** By dividing the set of decision variables, this method achieves parallelism. The solution space is smaller after partitioning, but more partitioning is required to allow for investigation of the entire solution space.

  Naturally, the set of solutions explored when using this concurrent implementation differs from that when using the same heuristic method sequentially.

- **Type 3:** Several concurrent investigations of the solution space lead to parallelism.

4.1 Parallelism of type 1:

Parallelizations of type 1 may be attained by performing the operations simultaneously or by simultaneously evaluating the many movements that make up a search technique iteration. Directly reducing the time it takes for a particular solution method to execute is the goal of type 1 parallelization schemes.

The parallel implementation of the method follows the same exploration path through the problem domain as the sequential implementation and produces the same solution when the same number of iterations are permitted for both the sequential and parallel versions of the method and the same operations are carried out at each iteration. As a result, convenient application of common parallel performance measurements is made. Some implementations adjust the sequentialmethod without changing the fundamental search strategy in order to make use of the additional computing power available[14].

4.2 Parallelism of type 2:

Parallelism in Type 2 methods results from the division of the choice variables into separate subsets. Each subset receives the specific heuristic, and variables outside the subset are regarded as fixed. The implementation of type 2 strategies often takes place within a master-slave framework :

- A master process divides the decision variables. The master could change the partition while it's conducting the search. Modifications may be made at predetermined or improvised intervals, or, more frequently, when the method is restarted, at intervals that are adjusted.
- Slaves autonomously and concurrently study the divisions that have been given to them.Moves can only take place within the partition, with the other variables being fixed and unaffected by the moves that are executed, or the slaves can have access to all of the variables.
- When slaves have total access to the neighborhood, the master must carry out a more difficult operation by merging the incomplete solutions gained from each subgroup to create a complete solution to the issue[14].

4.3 Parallelism of type 3:

With the first two parallelization techniques, just one search path is produced. Type 3 parallelization procedures are those that include many concurrent searches in the solution space. The same heuristic approach may or may not be used by each concurrent thread.

In order to find the greatest overall answer, they may begin with the same or distinct initial solutions and communicate throughout the search or only at the very end. The former are sometimes referred to as cooperative multi-thread strategies, whilst the latter are known as independent search methods. Communications can be carried out synchronously or asynchronously, event-driven, or at fixed or arbitrary times.

While utilizing a multi-thread strategy to speed up computing, one often strives to make each thread execute a quicker search than the sequential method.Each class of meta-heuristics uses a different implementation of this technique.

In this type there are several algorithms such as:

- **Island Model:** Using the island model, distinct subpopulations known as islands are subjected to simultaneous execution of multiple instances of the metaheuristic algorithm. Each island executes local search operations and evolves independently utilizing its own collection of solution candidates. Solutions are periodically traded across islands to advance global exploration and exploitation[25].

  the island model (Island model) This model is also called the parallel model of distributed genetic algorithms or the parallel model with coarse granularity, where each subpopulation is assigned to a processor. The selection, crossing and mutation operations are carried out at the level of each subpopulation. Communication between the subpopulations is ensured by the migration of individuals between them; without this Sub-populations are rapidly converging towards non-optimal solutions.

  A good migration strategy can yield comparative results; migration is controlled by several parameters such as:

  - **Neighborhood Structure:** determines topology and density of an inter-sub-population connection. If the connection density is very strong the set will behave as a single algorithm; and if the connection is low each will behave as an independent population.
  - **The choice of individuals to be migrated:** defines the nature of the individuals to be migrated can be the best, bad or random.
  - **Migration time:** defines when individuals should be immigrants; the inter-population migration is either synchronous is determined in advance withfixed intervals, or asynchronous.

4.4 Hardware of processing in parallel :

specialized hardware exists including Graphics Processing Unit(GPUs), to perform complex calculations and graphics processing in parallel .

4.4.1 Graphics Processing Unit(GPU):

A graphics processing unit, or GPU, is a specialized electronic circuit whose main purpose is to quickly manipulate and change memory to speed up the generation of images and films on a display screen. But GPUs are not just for graphics processing; they have also shown to be quite efficient at speeding up a wide range of other computing activities, including machine learning[8].

4.4.2 GPU Adaptive memory:

In this method, we focused on the use of parallel computation to reduce the time of the impact task by consuming time, which is the evaluation process, as we took advantage of the power granted by the GPU to do the cos distance calculation in a parallel manner.

The method can be summarized in the following steps:

1. A random starting process based on the local memory, which was created from the basket of available words.
2. Evaluate the first solution.
3. Generate the first solution(voisinage):Adjacent is an extension that differs from the current extension by a certain number of words n, and it is called adjacency of n order, we used adjacency of n order 1.
4. Invalidate the neighbor by parallel arithmetic.
5. Obtain the local optimal solution.
6. Update the local memory using the optimal solution.
7. Return to step 3 as long as the stopping condition is not met

Conclusion :

When the size of the input data of a difficult problem increases, the size of the search space also increases, in this case the use of metaheuristics in their original versions does not give the desired results, to increase the search power of metaheuristics parallelism is a technique that is generally used. Parallelism consists of executing several instructions simultaneously on one or more machine(s) capable of doing this. The parallelism of metaheuristics is achieved in several ways which subsequently generate a classification of different types.

In the next chapter, we will know the program that we relied on in formulating the work, followed by a number of results and comparisons

CHAPTER 03

EXPERIMENTAL AND RESULT

Introduction:

Through the first chapter and second ;which are the beginning of the work whose results we present in this chapter; through Analyzing co-occurrencesfor query expansion while using adaptive memory A powerful tool for exploring and analyzing the connections between elements in a dataset is the metaheuristic. This metaheuristic technique offers an effective strategy to optimize the arrangement of objects based on their co-occurrence patterns by utilizing the idea of adaptive memory, which integrates learning from prior solutions to direct the search process for find the best combination for initial query of user,.

This study's goal was to use the Tanimoto similarity measure to apply the adaptive memory metaheuristic to a co-occurrence matrix. The frequency or presence/absence of items occurring together in a dataset is represented by the co-occurrence matrix. The co-occurrence link between two elements is stronger the higher their value is in the matrix.

In this chapter, we know the language we used, as well as the dataset that we worked with, and we present the results that we reached

1 Working tools:

1.1 Python definition :

Python is a modern, simple-to-learn, and strong programming language. With dynamic typing and dynamic binding, it provides efficient high-level data structures and an easy-to-use but effective method of object-oriented programming. It facilitates software modularity and code reuse by supporting modules and packages. Python's interpreted nature and beautiful syntax make it the perfect language for scripting and quick application development across a wide range of platforms.

For every major platform, the Python interpreter and the extensive standard library are freely distributable and accessible in source or binary form. It may be rapidly expanded with new C or C++-based functions and data types. Python is a good choice as an add-on language for flexible software.

Python is a great choice for creating the higher-level components of complex scientific programs [Hinsen97] and for running simulations on parallel computing platforms like SMPs or PC clusters. Python codes can integrate highly with other libraries created in compiled languages and are swiftly built and maintained[15].

1.2 Dataset CISI:

The Centre for Inventions and Scientific Information (”CISI”) gathered the information, which consists of text information about 1,460 documents and 112 related queries. Its goal is to be used to create information retrieval models in which a given query will produce a list of document IDs that are pertinent to the query.

The proper list of query-document matching, also known as the ”gold standard” or ”ground proof,” is contained in the file ”CISI.REL. Your model's performance can be evaluated by comparing it to this ”gold standard” by doing so[16].

1.3 CUDA:

NVIDIA developed the parallel computing platform and application programming interface (API) model known as CUDA (Compute Unified Device Architecture). It enables programmers to take advantage of NVIDIA GPUs' processing capacity for general-purpose applications like highly parallel computations. PyCUDA is one of the most well-liked libraries for CUDA programming in Python. Python programmers may easily and effectively create CUDA applications with PyCUDA.

Developers must install the CUDA Toolkit, which contains the required drivers and libraries for GPU programming, in order to use PyCUDA. After being installed, PyCUDA enables users to launch CUDA kernels, allocate and move data between the CPU and GPU memory, and return results back to the CPU.

In conclusion, using CUDA with Python, especially with the PyCUDA module, enables developers to take advantage of NVIDIA GPUs' parallel processing capacity for high-performance computing jobs. It makes it simpler to use the computing power of GPUs for a variety of applications by offering an approachable and effective way to develop GPU-accelerated Python programs[31].

2 The setup of our work

In our project,and after we read all the woods related to each the documents and the queries related to each document,we did the following steps,and each step is also divided by stages:

2.1 Preprocessing steps:

2.1.1 Reading documents and queries:

We read and display all that is in the documents,meaning we find the document number, followed by a paragraph that includes its content of sentences and words(**figure 10**).The some applies to queries(**figure ??**).


```
{'1': " 18 Editions of the Dewey Decimal Classifications Comaromi, J.P. The present study is a
history of the DEWEY Decimal Classification.  The first edition of the DDC was published in
1876, the eighteenth edition in 1971, and future editions will continue to appear as needed.  In
spite of the DDC's long and healthy life, however, its full story has never been told.  There
have been biographies of Dewey that briefly describe his system, but this is the first attempt
to provide a detailed history of the work that more than any other has spurred the growth of
librarianship in this country and abroad. ", '2': 'Use Made of Technical Libraries Slater, M.
This report is an analysis of 6300 acts of use in 104 technical libraries in the United Kingdom.
Library use is only one aspect of the wider pattern of information use.  Information transfer in
libraries is restricted to the use of documents.  It takes no account of documents used outside
the library, still less of information transferred orally from person to person.  The library
```


Figure 10: Read documents.

2.1.2 Extracting of documents and queries:

After read the words of all documents and all queries in dataset CISI; we sort each document and each query where we get a dictionary the number of each document ;followed by a series of words in it,the**figure 11** shows it.


```
'mathematical', 'definition', 'ancient', 'parallel', 'discussions', 'probability', 'one',
'course', 'legislate', 'meaning', 'term', 'depends', 'alice', 'pointed', '``', 'master', "''",
'user', 'term', 'hand', 'use', 'single', 'term', 'document', 'cover', 'two', 'distinct',
'meanings', 'especially', 'usage', 'designed', 'secure', 'acceptance', 'doctrine',
'attributing', 'mathematical', 'validity', 'represents', 'serious', 'situation', 'merely',
'careless', 'ambiguity'], '29': ['questions', 'concerning', '``', 'information', 'need', "''",
"o'connor", 'j.', 'expression', '``', 'satisfying', 'requester', "'s", 'information', 'need',
"''", 'often', 'used', 'meaning', 'obscure', 'literature', '``', 'information', 'need', "''",
'relation', 'retrieval', 'suggests', 'three', 'different', 'though', 'inconsistent', 'possible',
'interpretations', 'however', 'interpretations', 'fundamentally', 'unclear', 'various',
'obscurities', 'involved', 'indicated', 'critical', 'questions', 'write', 'information', 'need',
'invited', 'answer'], '30': ['vocabulary', 'building', 'control', 'techniques', 'wall',
'eugene', 'rationale', 'given', 'creation', 'maintainance', 'information', 'center',
'controlled', 'indexing', 'retrieval', 'vocabulary', '..', 'basic', 'vocabulary', 'principles',
'1', 'use', 'natural', 'language', '2', 'development', 'hospitality', 'new', 'concepts', '3',
'provision', 'adequate', 'cross-referencing', '4', 'formatting', 'easy', 'use', '..',
'terminalogical', 'conventions', 'necessary', 'development', 'control', 'useful', 'vocabulary',
'summarized', 'techniques', 'applying', 'conventions', 'construct', 'thesaurus', 'described',
```


Figure 11: All words in documents.

And we also get number of the query followed by the words in it.

2.1.3 Remove stop words:

Through what we obtained in the previous step,we examine again by reading each of the documents and queries.Then we insert a function that removes the stop words present in each of them.In the end we get :

A dictionary of document without stopwords,this is what (**figure12**) shows.


```
{'1': ['18', 'editions', 'dewey', 'decimal', 'classifications', 'comaromi', 'j.p.', 'present',
'study', 'history', 'dewey', 'decimal', 'classification', 'first', 'edition', 'ddc',
'published', '1876', 'eighteenth', 'edition', '1971', 'future', 'editions', 'continue',
'appear', 'needed', 'spite', 'ddc', "'s", 'long', 'healthy', 'life', 'however', 'full', 'story',
'never', 'told', 'biographies', 'dewey', 'briefly', 'describe', 'system', 'first', 'attempt',
'provide', 'detailed', 'history', 'work', 'spurred', 'growth', 'librarianship', 'country',
'abroad'], '2': ['use', 'made', 'technical', 'libraries', 'slater', 'm.', 'report', 'analysis',
'6300', 'acts', 'use', '104', 'technical', 'libraries', 'united', 'kingdom', 'library', 'use',
'one', 'aspect', 'wider', 'pattern', 'information', 'use', 'information', 'transfer',
'libraries', 'restricted', 'use', 'documents', 'takes', 'account', 'documents', 'used',
'outside', 'library', 'still', 'less', 'information', 'transferred', 'orally', 'person',
'person', 'library', 'acts', 'channel', 'proportion', 'situations', 'information',
'transferred', 'taking', 'technical', 'information', 'transfer', 'whole', 'doubt', 'proportion',
'major', 'one', 'users', 'technical', 'information', 'particularly', 'technology', 'rather',
'science', 'visit', 'libraries', 'rarely', 'relying', 'desk', 'collections', 'handbooks',
'current', 'periodicals', 'personal', 'contact', 'colleagues', 'people', 'organizations',
'even', 'regular', 'library', 'users', 'also', 'receive', 'information', 'ways'], '3': ['two',
```


Figure 12: Remove the stop words in documents.

2.1.4 Stemming:

After reading all the words found in the documents, normalization words by taking them back to their most basic form .


```
{'1': 'problem concern mak describ titl difficul involv autom retriev artic approxim titl us
relev cont artic titl', '2': 'act pertin dat oppos ref entir artic retriev autom respons inform
request', '3': 'inform sci giv definit poss', '4': 'im recognit method autom transform print
text computer-ready form', '5': 'spec train ordin research businessm nee prop inform man
unobstruct us inform retriev system problem lik encount', '6': 'poss verb commun comput hum
commun via spok word', '7': 'describ pres work plan system publ print origin pap comput sav
byproduc artic cod data-processing form us retriev', '8': 'describ inform retriev index langu
bear sci gen', '9': 'poss autom gram context analys artic includ inform retriev system', '10':
'us abstract mathem inform retriev e.g group the', '11': 'nee inform consolid evalu retriev sci
research', '12': 'giv method high spee publ print distribut sci journ', '13': 'criter develop
object evalu inform retriev dissemin system', '14': 'fut autom med diagnos', '15': 'much inform
retriev dissemin system wel autom libr cost wor research industry', '16': 'system incorp
multiprogram remot stat inform retriev ext us fut', '17': 'mean obtain larg volum high spee
custom us inform retriev output', '18': 'method encod autom match autom draw structures extend
two dimend lik structural formula chem compound', '19': 'techn machin match machin search system
cod match method', '20': 'test autom inform system', '21': 'nee provid personnel inform field',
'22': 'autom inform med field', '23': 'amount us book libr rel nee autom inform system', '24':
```


Figure 13: Stemming of queries.

Some with what we've done with the documents we're doing in the queries.

We identified the words of each document without repetitions,i.e.a dictionary in which the number of the document is followed by a series of words in it without repetition,as shows in **figure13**,we get a list of all the words in all the documents without repetition,which we named **features of documents** whose length is 7630 words.

```
['18', 'edit', 'dewey', 'decim', 'class', 'comarom', 'j.p.', 'pres', 'study', 'hist', 'first',
'ddc', 'publ', '1876', 'eighteen', '1971', 'fut', 'continu', 'appear', 'nee', 'spit', "'s",
'long', 'healthy', 'lif', 'howev', 'ful', 'story', 'nev', 'told', 'biograph', 'brief',
'describ', 'system', 'attempt', 'provid', 'detail', 'work', 'spur', 'grow', 'libr', 'country',
'abroad', 'us', 'mad', 'techn', 'slat', 'm.', 'report', 'analys', '6300', 'act', '104', 'unit',
'kingdom', 'on', 'aspect', 'wid', 'pattern', 'inform', 'transf', 'restrict', 'docu', 'tak',
'account', 'outsid', 'stil', 'less', 'transfer', 'or', 'person', 'channel', 'proport', 'situ',
'whol', 'doubt', 'maj', 'particul', 'technolog', 'rath', 'sci', 'visit', 'rar', 'rely', 'desk',
'collect', 'handbook', 'cur', 'period', 'contact', 'colleagu', 'peopl', 'org', 'ev', 'regul',
'also', 'receiv', 'way', 'two', 'kind', 'pow', 'essay', 'bibliograph', 'control', 'wilson',
'p.', 'rel', 'writ', 'knowledg', 'inevit', 'ent', 'contain', 'along', 'much', 'els', 'gre',
```

Figure 14: Features of all documents.

the same for words of queries.

2.1.5 Co-occurrence:

We created the matrix of co-occurrence where the columns of this matrix is the features of documents and the line is the features of queries.The **figure 15** shows what we got.

```
[[ 2. 10.  0.  1. 40.  0.  3. 80. 81. 21.]
 [ 0.  5.  0.  1. 23.  0.  1. 33. 54. 13.]
 [ 2.  8.  1.  3. 18.  0.  2. 35. 41.  9.]
 [ 2. 17.  4.  4. 51.  1.  3. 83. 77. 17.]
 [ 2. 10.  1.  2. 15.  0.  0. 28. 43.  5.]
 [ 0.  2.  0.  0.  8.  0.  0. 11. 11.  0.]
 [ 0.  1.  0.  0. 12.  0.  1. 25. 14.  4.]
 [ 1.  5.  0.  0. 28.  0.  0. 26. 35.  3.]
 [ 1.  7.  0.  2. 46.  0.  0. 72. 51.  7.]
 [ 0.  6.  1.  1. 15.  0.  0. 32. 31.  4.]]
(1130, 7630)
```

Figure 15: Co-occurrence matrix.

2.1.6 Tanimoto matrix:

Tanimoto matrix based on co-occurrence matrix,but there is a difference in the account where we apply the low we found in the first chapter(**figure 18**).

```
[[5.34619259e-09 2.15931353e-08 0.00000000e+00 2.61450412e-09
  5.85592778e-08 0.00000000e+00 8.23314343e-09 7.45037530e-08
  6.90309129e-08 4.49084941e-08]
 [0.00000000e+00 3.04335035e-08 0.00000000e+00 8.48002865e-09
  7.57343876e-08 0.00000000e+00 9.24668949e-09 5.69867188e-08
  8.26339559e-08 7.73154787e-08]
 [1.55819378e-08 4.38810982e-08 7.70500171e-09 2.25103315e-08
  5.49223305e-08 0.00000000e+00 1.62722694e-08 5.72093626e-08
  5.95939678e-08 4.82645601e-08]
 [4.82233907e-09 3.34892539e-08 9.59038330e-09 9.45057693e-09
  6.94003375e-08 2.51979203e-09 7.42681426e-09 7.30573040e-08
  6.22147360e-08 3.32307852e-08]
 [2.88450151e-08 9.09050949e-08 1.42116649e-08 2.74237986e-08
  6.60516006e-08 0.00000000e+00 0.00000000e+00 5.95679318e-08
  8.00637431e-08 4.40953896e-08]
```

Figure 16: Tanimoto matrix.

2.2 Selecting and ranking:

selecting a specific number where the weight of each word is calculated for each query;this by using a tanimoto matrixAnd arranged according to the weight of each word from the words of the queries,

```
{'1': {'supposit': 0.2053308700217579, 'morn': 0.19392920149417478, 'claus':
0.18376086230792077, 'interpoint': 0.1769208778273212, 'n': 0.1759960668686853, 'sel':
0.1746394591984409, '752': 0.17318046590499822, 'baxt': 0.1664130910504147, '24,953':
0.1633391501982058, 'throws': 0.16236954110251017, 'monopo': 0.15895706454541442, 'langendo':
0.15845194867014015, 'krusk': 0.15791141953549143, 'borel': 0.15483920449101507, "'describing":
0.15473190773111817, 'ross': 0.15119400385312512, "'anyhow": 0.14782444529242714, 'yul':
```

Figure 17: Tanimoto matrix.

2.3 Architecture of programming Adaptive memory:

this is the steps of our programming started from preprocessing then indexation to GPU adaptive memory.

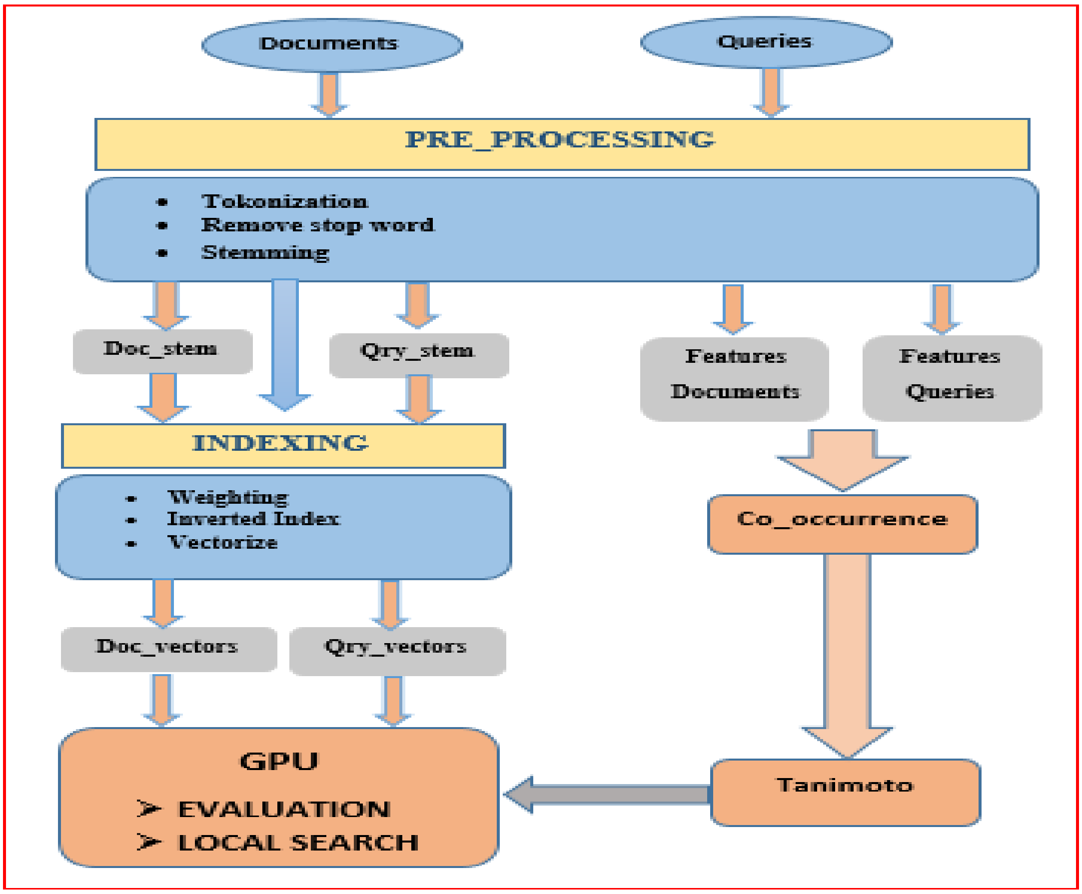


Figure 18: Steps of work.

3 Measures in evaluating IR

3.1 MAP (Mean average Precision):

– **Average Precision** – Mean of the precision scores for a single query after each relevant document is retrieved, where relevant documents not retrieved have P of zero.

- Commonly 10-points of recall is used!

– **MAP** is the mean of average precisions for a query batch

3.2 P@KEY:

Precision at X documents retrieved (in Web searching).

Problem: the cut-off at x represents many different recall levels for different queries - also P@x.

3.3 MRR:

The Mean Reciprocal Rank (MRR) is an order-aware metric, which means that, unlike recall@K, returning an actual relevant result at rank one scores better than at rank four.

Another differentiator for MRR is that it is calculated based on multiple queries. It is calculated as:

$$MRR = \frac{\sum \frac{1}{rankq}}{n} \tag{7}$$

n is the number of queries,q a specific query.

rankq :the rank of the first *actual relevant *result for query q.

4 Results:

4.1 MAP AND MRR:

The table shows a significant improvement of two value MAP and MRR, due to the method's ability to handle more possibilities and more combinations than the direct method.

	GPU-AM	ORIGINAL-Q
MAP	0.166407	0.107855
MRR	0.642857	0.43312

Table 1: Comparison between GPU and Serial

4.2 P@5:

The improvement was not noticeable enough, due to several reasons, including:

1. Failure to adjust the parameters of metaheuristic.

2. And not to control the length of the original query.

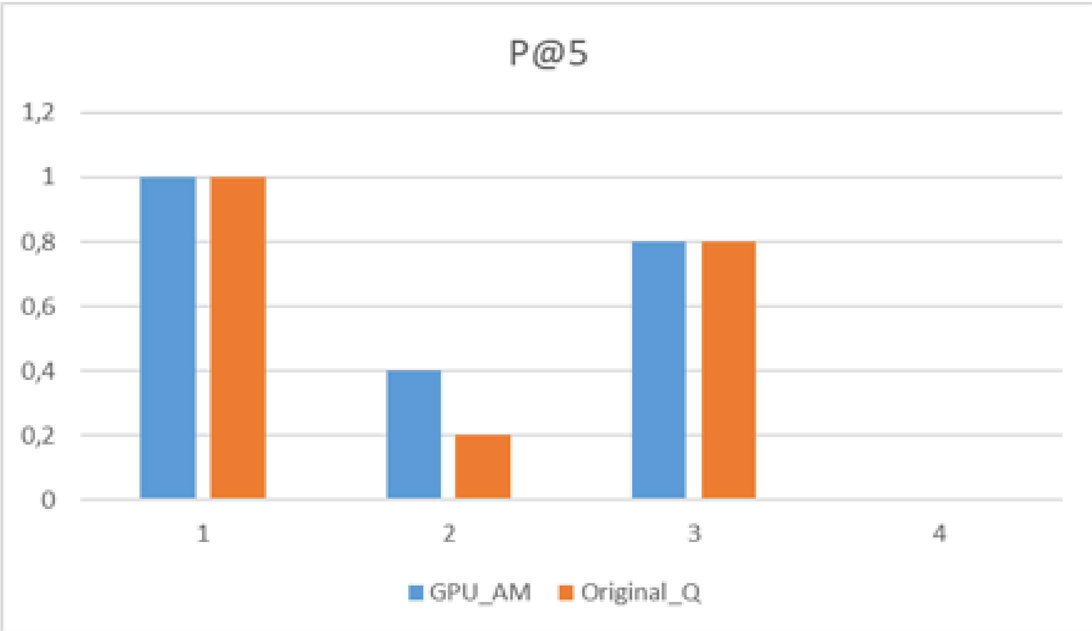


Figure 19: Results p@5 for 4 queries.

P@20:

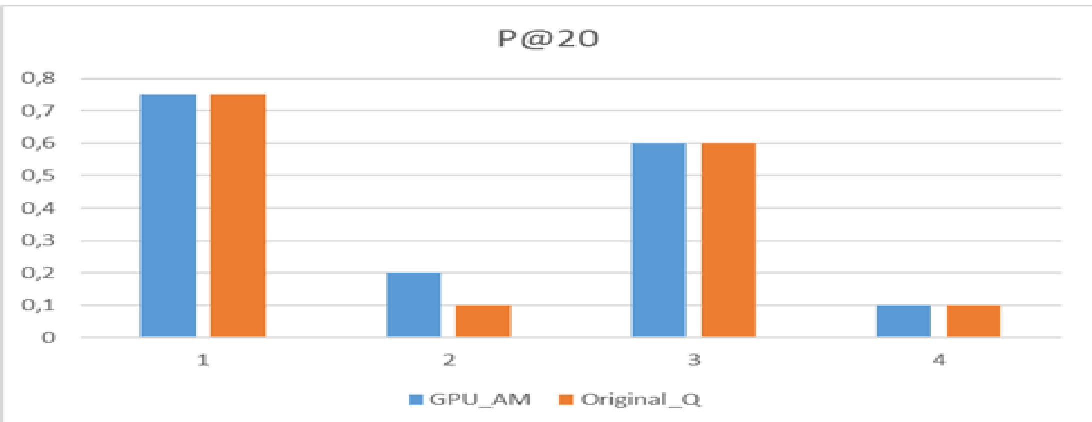


Figure 20: Results p@20 for 4 queries.

General Conclusion

Our goal in this work is to create parallel applications that work on information retrieval based on the expansion of the query, and this is through the use of the co-occurrence matrix approach.In this work, we created an adaptive memory application with GPU evaluation, as well as serial.

After that, we tested this approach on the database, where the results showed that the approach we worked on was good, acceptable and not expensive, but we noticed that we were unable to implement it on a powerful machine, which led to a limit to the approach.

We concluded from this work that the programming and design of parallel applications is a work that differs radically from the design and implementation of ordinary applications, where several factors must be taken into account, including the target machine and the nature of the target program, and the possibility of integrating programming technologies, this makes the work much greater than the design and application of a regular program.

We wanted to run this program on a robust machine that would allow true measurement of program effectiveness,this is due her absence ; We also encountered the problem of adjusting parameters of metaheuristic.

As well as the implementation of software on other databases such as TREC, CACM,..

We were hoping to implement the program by using MPI and connecting it to the GPU, but the time was not enough.

References

[1] Lasmedi Afuan, Ahmad Ashari, and Yohanes Suyanto. A study: query expansion methods in information retrieval. In *Journal of Physics: Conference Series*, volume 1367, page 012001. IOP Publishing, 2019.

[2] Prachi Agrawal, Hattan F Abutarboush, Talari Ganesh, and Ali Wagdy Mohamed. Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *Ieee Access*, 9:26766–26791, 2021.

[3] Aytaç Altan. Performance of metaheuristic optimization algorithms based on swarm intelligence in attitude and altitude control of unmanned aerial vehicle for path following. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6. IEEE, 2020.

[4] David Avis and Charles Jordan. mplrs: A scalable parallel vertex/facet enumeration code. *Mathematical Programming Computation*, 10(2):267–302, 2018.

[5] Hiteshwar Kumar Azad and Akshay Deepak. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735, 2019.

[6] Mahdi Azizi, Siamak Talatahari, and Amir H Gandomi. Fire hawk optimizer: A novel metaheuristic algorithm. *Artificial Intelligence Review*, 56(1):287–363, 2023.

[7] Halima BENKADDOUR and Ramzi ARIBI. *Meta-heuristiques parallèles pour la résolution des problèmes difficiles*. PhD thesis, 2013.

[8] André R Brodtkorb, Trond R Hagen, and Martin L Sætra. Graphics processing unit (gpu) programming strategies and trends in gpu computing. *Journal of Parallel and Distributed Computing*, 73(1):4–13, 2013.

[9] Abey Bruck, Tulu Tilahun, et al. Enhancing amharic information retrieval system based on statistical co-occurrence technique. *Journal of Computer and Communications*, 3(12):67, 2015.

[10] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. Retrievability based document selection for relevance feedback with automatically generated query variants. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 125–134, 2020.

[11] Yuanyuan Chen, Yisheng Lv, and Fei-Yue Wang. Traffic flow imputation using parallel data and generative adversarial networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1624–1630, 2019.

[12] Qikai Cheng, Jiamin Wang, Wei Lu, Yong Huang, and Yi Bu. Keyword-citation-keyword network: A new perspective of discipline knowledge structure analysis. *Scientometrics*, 124:1923–1943, 2020.

[13] Vincent Claveau. Vectorisation, okapi et calcul de similarité pour le tal: pour oublier enfin le tf-idf (vectorization, okapi and computing similarity for nlp: Say goodbye to tf-idf)[in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN*, pages 85–98, 2012.

[14] Teodor Gabriel Crainic and Michel Toulouse. *Parallel strategies for meta-heuristics*. Springer, 2003.

[15] Lisandro Dalcin. Mpi for python, 2019.

[16] Digvijay Desai, Aniruddha Ghadge, Roshan Wazare, and Jayshree Bagade. A comparative study of information retrieval models for short document summaries. In *Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021*, pages 547–562. Springer, 2022.

[17] Johann Dréo and Patrick Siarry. Métaheuristiques d'optimisation vues sous l'angle de l'échantillonnage de distribution. *Apii Jesa Journal Europeen Des Systemes Automatises*, 42(1):9, 2008.

[18] Yasir Hadi Farhan, Shahrul Azman Mohd Noah, and Masnizah Mohd. Survey of automatic query expansion for arabic text retrieval. 2020.

[19] Fernando Fausto, Adolfo Reyna-Orta, Erik Cuevas, Ángel G Andrade, and Marco Perez-Cisneros. From ants to whales: metaheuristics for all tastes. *Artificial Intelligence Review*, 53:753–810, 2020.

[20] Fred Glover, Manuel Laguna, and Rafael Martí. Principles and strategies of tabu search. In *Handbook of Approximation Algorithms and Metaheuristics, Second Edition*, pages 361–377. Chapman and Hall/CRC, 2018.

[21] Eva Hopper. *Two-dimensional packing utilising evolutionary algorithms and other meta-heuristic methods*. PhD thesis, University of Wales. Cardiff, 2000.

[22] Qing Huang, Yang Yang, and Ming Cheng. Deep learning the semantics of change sequences for query expansion. *Software: Practice and Experience*, 49(11):1600–1617, 2019.

[23] Kashif Hussain, Mohd Najib Mohd Salleh, Shi Cheng, and Yuhui Shi. Meta-heuristic research: a comprehensive survey. *Artificial intelligence review*, 52:2191–2233, 2019.

[24] Minoru Kanehisa and Yoko Sato. Kegg mapper for inferring cellular functions from protein sequences. *Protein Science*, 29(1):28–35, 2020.

[25] Alper Kizil and Korhan Karabulut. Effects of parameters of an island model parallel genetic algorithm for the quadratic assignment problem. In *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 444–449. IEEE, 2019.

[26] Meeta Kumar and Anand J Kulkarni. Socio-inspired optimization metaheuristics: a review. *Socio-cultural inspired metaheuristics*, pages 241–265, 2019.

[27] Craig Macdonald and Nicola Tonellotto. Declarative experimentation in information retrieval using pyterrier. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, pages 161–168, 2020.

[28] Christopher D Manning. *An introduction to information retrieval.* Cambridge university press, 2009.

[29] Youssef Meliani, Yasmina Hani, Sâad Lissane Elhaq, and Abderrahman El Mhamedi. A developed tabu search algorithm for heterogeneous fleet vehicle routing problem. *IFAC-PapersOnLine*, 52(13):1051–1056, 2019.

[30] Doaa N Mhawi, Haider W Oleiwi, Nagham H Saeed, and Heba L Al-Taie. An efficient information retrieval system using evolutionary algorithms. *Network*, 2(4):583–605, 2022.

[31] Lena Oden. Lessons learned from comparing c-cuda and python-numba for gpu-computing. In *2020 28th Euromicro international conference on parallel, distributed and network-based processing (PDP)*, pages 216–223. IEEE, 2020.

[32] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Application of deep learning for object detection. *Procedia computer science*, 132:1706–1717, 2018.

[33] Helen J Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the american society for information science*, 42(5):378–383, 1991.

[34] João Luiz Junho Pereira, Matheus Brendon Francisco, Camila Aparecida Diniz, Guilherme Antônio Oliver, Sebastiao Simoes Cunha Jr, and Guilherme Ferreira Gomes. Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Systems with Applications*, 170:114522, 2021.

[35] Günther R Raidl, Jakob Puchinger, and Christian Blum. Metaheuristic hybrids. *Handbook of metaheuristics*, pages 385–417, 2019.

[36] Imran Rasheed, Haider Banka, and Hamaid Mahmood Khan. Pseudo-relevance feedback based query expansion using boosting algorithm. *Artificial Intelligence Review*, 54(8):6101–6124, 2021.

[37] Muhammad Ahsan Raza, Rahmah Mokhtar, Noraziah Ahmad, Maruf Pasha, and Urooj Pasha. A taxonomy and survey of semantic approaches for query expansion. *IEEE Access*, 7:17823–17833, 2019.

[38] John Reid, Bill Long, and Jon Steidel. History of coarrays and spmd parallelism in fortran. *Proceedings of the ACM on Programming Languages*, 4(HOPL):1–30, 2020.

[39] Anil Sharma and Suresh Kumar. Bayesian rough set based information retrieval. *Journal of Statistics and Management Systems*, 23(7):1147–1158, 2020.

[40] Anu Sharma, MK Sharma, and Rakesh Kr Dwivedi. Exploratory data analysis and deception detection in news articles on social media using machine learning classifiers. *Ain Shams Engineering Journal*, page 102166, 2023.

[41] Manik Sharma and Prableen Kaur. A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem. *Archives of Computational Methods in Engineering*, 28:1103–1127, 2021.

[42] Jagendra Singh and Rakesh Kumar. Lexical co-occurrence and contextual window-based approach with semantic similarity for query expansion. *International Journal of Intelligent Information Technologies (IJIIT)*, 13(3):57–78, 2017.

[43] Jagendra Singh and Aditi Sharan. A new fuzzy logic-based query expansion model for efficient information retrieval using relevance feedback approach. *Neural Computing and Applications*, 28:2557–2580, 2017.

[44] Asiri Umenga Weerasuriya, Xuelin Zhang, Jiayao Wang, Bin Lu, Kam Tim Tse, and Chun-Ho Liu. Performance evaluation of population-based metaheuristic algorithms and decision-making for multi-objective optimization of building design. *Building and Environment*, 198:107855, 2021.

[45] Xicheng Yin, Hongwei Wang, Pei Yin, Hengmin Zhu, and Zhenyu Zhang. A co-occurrence based approach of automatic keyword expansion using mass diffusion. *Scientometrics*, 124:1885–1905, 2020.

[46] Binbin Yu. Research on information retrieval model based on ontology. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–8, 2019.