SAAD DAHLAB UNIVERSITY – BLIDA 1

FACULTY OF SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

DOCTORAL THESIS

IN COMPUTER SYSTEMS ENGINEERING

CONTEXT-AWARE INFORMATION RETRIEVAL SYSTEMS: CONTRIBUTION TO A SEMANTICALLY ENRICHED, FOLKSONOMY-BASED TEXT-SEARCH.

BY

MELYARA MEZZI

Members of the thesis commitee:

Ould Khaoua M.	Professor	U. Blida 1	President
Benblidia N.	Professor	U. Blida 1	Supervisor
Bellatreche L.	Professor	U. Poitiers	Examinator
Oukid S.	Associate professor	U. Blida 1	Examinator
Boustia N.	Associate professor	U. Blida 1	Examinator
Aliane H.	Research associate	CERIST	Examinator

Blida, June 2018

UNIVERSITE SAAD DAHLEB – BLIDA 1

FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

THESE DE DOCTORAT

EN GENIE DES SYSTEMES INFORMATIQUES

SYSTEME DE RECHERCHE SENSIBLE AU CONTEXTE: CONTRIBUTION A UN MODELE SEMANTIQUEMENT ENRICHI POUR LA RECHERCHE D'INFORMATION TEXTUELLE BASEE SUR LES FOLKSONOMIES.

PAR

MELYARA MEZZI

Devant le jury compose de:

Ould Khaoua M.	Professeur	U. Blida 1	Président
Benblidia N.	Professeur	U. Blida 1	Directrice de thèse
Bellatreche L.	Professeur	U. Poitiers	Examinateur
Oukid S.	Maître de conférences	U. Blida 1	Examinatrice
Boustia N.	Maître de conférences	U. Blida 1	Examinatrice
Aliane H.	Maître de recherche	CERIST	Examinatrice

Blida, June 2018

SUMMARY

Information Retrieval (IR) became indispensable to our modern knowledge-based society. Modern information environments are becoming large and complex as well as ubiquitous, because the amount of available heterogeneous information grows exponentially each year. Almost every aspect of our lives and every profession are affected by the information available on the Internet. Indeed, we live in a search society - belief that (almost) everything is known, we just have to find the information. We search for everything the good book, the new movie in cinema, the best car, the most comfortable home, the best vacation plans, even the best search engines.

Full-text search on the World Wide Web (WWW) is perhaps the most widely used IR application, this application is concerned with the processing, indexing and retrieval of huge amount of textual documents.

This dissertation investigates whether the inclusion of a contextual dimension (i.e. “Content” of queries and documents and “User” in our case) in the IR process can improve the effectiveness of an IR System by better indexing the documents and computing the mappings between them and the queries more accurately. Thus, we propose a semantically enriched context-aware Information Retrieval System, based on Folksonomies or social tags so as to provide more relevant search results and cope with the traditional Information Retrieval issues that does not satisfy our modern society needs.

In this regard, two effective novel indexing and query-document mapping methods are proposed and evaluated. The first method focuses on the semantic aspects of documents and queries with a semantically enriched stemming algorithm based on the well-known Porter Algorithm. The second method focuses on the trustworthiness of the social environment of a user by exploring social-bookmarking as a new indexing technique through the use of Folksonomies as a new alternative to Ontologies in knowledge representation for IR purposes.

Keywords: *Information Retrieval, Semantic web, Stemming, Context-modelling, Social-bookmarking, Folksonomies, Natural Language Processing.*

RESUME

La Recherche d'Information (RI) est devenue indispensable à notre société moderne fondée sur le savoir. Les environnements d'information modernes sont de plus en plus importants, complexes et omniprésents, car la quantité d'informations hétérogènes disponibles augmente de façon exponentielle chaque année.

Presque tous les aspects de notre vie et de chaque profession sont affectés par les informations disponibles sur Internet. En effet, nous vivons dans une société de recherche en croyant que (presque) tout est connu, il suffit de le chercher et de le trouver. Nous cherchons tout, le bon livre, le nouveau film en cinéma, la meilleure voiture, la maison la plus confortable, les meilleurs plans vacances, même les meilleurs moteurs de recherche.

La recherche plein-texte sur l'internet est peut-être l'application de RI la plus utilisée. Cette application se base sur le traitement, l'indexation et la récupération d'une quantité énorme de documents textuels.

Cette thèse étudie si l'inclusion d'une dimension contextuelle dans le processus de RI (i.e. "Contenu" des requêtes et des documents et "Utilisateur" dans notre cas) peut améliorer l'efficacité d'un système de RI en indexant mieux les documents et en calculant les appariements entre ces derniers et les requêtes de manière plus précise. Ainsi, nous proposons un système de récupération d'informations sensible au contexte et sémantiquement enrichi, basé sur des Folksonomies ou des tags sociaux afin de fournir des résultats de recherche plus pertinents et de résoudre les problèmes de la RI traditionnelle qui ne satisfait plus les besoins de notre société moderne.

A ce titre, deux méthodes d'indexation et d'appariement entre requête et documents sont proposées et évaluées. La première méthode se concentre sur les aspects sémantiques des documents et des requêtes avec un algorithme de stemming (racinisation) sémantiquement enrichi basé sur l'algorithme connu de Porter. La deuxième méthode met l'accent

sur la fiabilité de l'environnement social d'un utilisateur en explorant le partage de signets comme une nouvelle technique d'indexation à travers l'utilisation des Folksonomies comme une nouvelle alternative aux ontologies dans la représentation des connaissances à des fins de recherche d'information.

Mots-clefs: *Recherche d'Information, Web Semantic, Stemming, Modélisation du Contexte, Partage de signets, Folksonomies, Traitement Automatique de la Langue.*

ملخص

أصبح استرجاع المعلومات (ام) لا غنى عنه في مجتمعنا الحديث القائم على المعرفة. بيئات المعلومات الحديثة صارت ذات أهمية متزايدة ومعقدة ومنتشرة، وذلك لأن كمية المعلومات غير المتجانسة يزيد أضعافا مضاعفة كل عام.

تقريبا كل جانب من حياتنا وكل مهنة تتأثر بالمعلومات المتاحة عبر شبكة الانترنت. والواقع أننا نعيش في مجتمع أبحاث ونؤمن بأن (تقريبا) كل شيء معروف ويمكننا ببساطة البحث والعثور عليه. لذلك نحن نحاول البحث عن كل شيء، الكتاب الجيد، الفيلم الجديد في السينما، أفضل سيارة، المنزل الأكثر راحة، أفضل خطط العطل، وحتى أفضل محركات البحث.

البحث عن معلومات نصية على شبكة الانترنت قد يكون التطبيق الأكثر استخدام في (ام). ويستند هذا التطبيق على تجهيز وفهرسة واسترجاع كمية كبيرة من السجلات النصية.

تبحث هذه الأطروحة ما إذا كان إدراج بعد سياقي (أي "محتوى" الاستعلامات والوثائق و "المستخدم" في حالتنا) في عملية استرجاع المعلومات يمكن أن يحسن فعالية نظام ام من خلال فهرسة أفضل للوثائق وحساب التعيينات بينهم والاستفسارات بشكل أكثر دقة. وبالتالي، فإننا نقترح نظام ام حساس للسياق وغني من الناحية الدلالية، يستند الى Folksonomies أو المفضلات الاجتماعية وذلك لتوفير نتائج بحث أكثر صلة والتعامل مع القضايا التقليدية الخاصة ب ام التي لا ترضي احتياجات مجتمعنا الحديث.

وفي هذا الصدد، نقترح أسلوبان جديدان فعالان للفهرسة والمطابقة بين الوثائق والاستفسارات. تركز الطريقة الأولى على الجوانب الدلالية للوثائق والاستفسارات مع خوارزمية الجذعية المخصبة دلاليا والمبنية على أساس خوارزمية بورتر المعروفة. ويركز الأسلوب الثاني على موثوقية البيئة الاجتماعية للمستخدم من خلال استكشاف المفضلات الاجتماعية باعتبارها تقنية جديدة للفهرسة واستخدام الFolksonomies كبديل جديد للأنطولوجيات في تمثيل المعرفة من أجل استرجاع المعلومات.

كلمات البحث: استرجاع المعلومات، ويب الدلالي، الجذعية، السياق المنمذج،Folksonomies ، المفضلات الاجتماعية، معالجة اللغة التلقائي.

ACKNOWLEGMENT

First of all, I would like to thank the members of the jury for kindly participating in the evaluation of this work.

A few people helped to bring this work to fruition and deserve thanks. I am grateful to my supervisor, Professor. Nadjia Benblidia, for her endless patience, priceless encouragement and guidance. Her support was of inestimable value in keeping me on track while giving me the space to pursue my own ideas. I am also indebted to Professor Djamel Bennouar, who believed in our ability to achieve something great. He undoubtedly, change the course of our lives for the better and contribute to rise us up, by giving us the opportunity to pursue this dream and make it come true. Further thanks goes to Doctor Messaouda Fareh, who was the first person to encourage me to do research. I will never forget her kindness, insightful feedbacks, constant support and righteous guidance. I would also like to acknowledge Professor Jimmy Huang and his fellow in York University, for having so kindly welcomed me in the Information Retrieval and Knowledge Management Research lab. He contributed greatly to the clarification of my ideas. I also thank Mister Abdelghani Bakhtouchi and my friend Sana Aroussi, who provided some of the most valuable advices I had in the early stages of this work and demonstrated with enthusiasm that academia is full of kind people willing to help each other in the sole goal to evolve research.

With a heart full of gratitude, I would like to acknowledge all the teachers and academic stuff I had the honour to meet throughout my life, and to my peers and colleagues at the University of Saad Dahlab Blida and S@TICOM Society, who made life as a PhD student so thoroughly enjoyable. Their comments, suggestions and tips filled small gaps in my knowledge when I needed a hand, and lightened my burden in more ways than one. Of my friends, particular thanks goes to “Bel ami” and “Fritte omelette” for standing steel for better, or worse; in sickness, and in health and for all the joyful and crazy moments we had (and we’ll have ان شاء الله)… a simple thanks would never be enough.

DEDICATION

In the realms of personal fulfilment, I would never have made it through without the precious support of my parents Noureddine and Aïcha and brothers Akram and Haythem, whose unconditional love, encouragement, and teasing were pivotal to my sanity and success… All I am and all that I will ever be is because of you… no one else could do what you have done for me… I am proud that, as a fabulous family, ALLAH has chosen you… you truly mean the world to me.

***

I would also want to thank my in-law family, for welcoming me, so kindly, as a new Mouzaï's member. A simple thank you will never be enough to describe my gratitude towards your generosity and encouragements.

***

And last but not least, I dedicate this work to My classmate, friend, teammate, soul mate, my husband Boualem Mouzaï for showing me that the impossible might indeed become possible by just believing… they say:" The deepest kind of peace and faith Are represented by the dove. It is thought to quiet our troubled thoughts and renew our mind and spirit"… Somehow, you were a different kind of dove. A dove that succeeded to break down my walls and brought peace of Mind to my troubled life… or maybe some trouble to such a peaceful girl… whatever it is, it changed my life for the better… and I believe that the best is yet to come ان شاء الله.

***

All the struggle we thought was in vain, all the mistakes, one life contained… they all finally start to go Away.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

GENERAL INTRODUCTION

1. Research background

Information retrieval (IR) is a paramount research area in the field of computer science and engineering. It is concerned with finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Since the near beginnings of civilization, human beings have focused on written communication. From cave drawings to scroll writings, from printing presses to electronic libraries, communicating was of primary concern to man's existence. Today, with the proliferation of digital libraries and electronic information exchange there is a clear need for improved techniques to organize large quantities of information. Applied and theoretical research and development in the areas of information authorship, processing, storage, and retrieval is of interest to all sectors of the community.

Over the last decades, there have been remarkable shifts in the area of Information Retrieval (IR) as huge amount of information is increasingly accumulated on the Web. The gigantic information explosion increases the need for discovering new tools that retrieve meaningful knowledge from various complex information sources. Thus, techniques primarily used to search and extract important information from numerous database sources have been a key challenge in current IR systems.

As recently as the 1990s, studies showed that most people preferred getting information from other people rather than from information retrieval systems. Of course, in that time period, most people also used human travel agents to book their travel. However, during the last decade, relentless optimization of information retrieval effectiveness has driven web search engines to new quality levels where most people are satisfied most of the time. For those reasons and much more, the field of

information retrieval has moved from being a primarily academic discipline to being the basis underlying most people's preferred means of information access.

2. Research problem

An information retrieval system aims at selecting relevant documents that meet user's information needs expressed with a textual query. During the years 1970-1980, various theoretical models have been proposed in this direction to represent, on the one hand, documents and queries and on the other hand to match information needs independently of the user. More recently and with the arrival of *Web2.0, known also as the* social Web, the effectiveness of these models has been questioned since they ignore the context in which the information is located.

Indeed, the Information Retrieval (IR) process begins with an anomalous state of knowledge (ASK). Then, many changes in knowledge state are involved. In short, IR is a purposeful process that alters the state of knowledge reacting to an information need or gap. A simple vision of an Information Retrieval System (IRS) was believed to be as follows: (a) the user expresses an Information need by formulating a question (called query); (b) the IRS answers the query and gives back results (texts, images, videos, etc.); and (c) the final phase is up to the user who has to evaluate and reformulate his/her query if the results do not satisfy his/her request.

Today, this vision became somehow obsolete, because the users, their queries, and the desired information were believed to be static. So, the relevance of a document was computed statically between the query and the set of documents ignoring the user, the device, the environment, and the specificities around the search activity which constitute the search context and are as a matter of fact highly variable factors. Besides, with the technology advances, information can nowadays, be accessed everywhere and at any time which add to the variability and the uniqueness of each search situation. And as no information is context

free, the inclusion of a contextual dimension in the classic IR process became a real challenge.

In short, we can say that Context includes all the intrinsic and extrinsic factors, which are related to a given search task and whose the direct or indirect inclusion in the IR process leads to enhance, whether implicitly or explicitly its effectiveness to convey the right information to the searcher.

Throughout years and with the advance of technology, search task became more flexible, allowing a wider range of choices between different sources of information, devices, and search categories. Moreover, the perspective of an eventual collaboration became possible, regardless of the location of the different searchers.

As Han, Wang, M., Wang, J. [16], we agree that task is the driving force that constitutes IR and real information behaviour. In order to find if there may be other contextual components, we choose sixteen valuable works that made use of context for different purposes. Our goal was to deepen our comprehension of the notion of context according to different use cases and to come out with a categorization of the context factors.

We found that the IR task is usually interlaced with seven contextual components, namely: user, queries, device, time, location, environment, and documents. We restricted our focus to those seven contextual factors and to test their coverage, we conducted a short survey among 434 anonymous online users (mostly Facebook and Linkedin users) about their search habits in order to understand the trends and users' intents in IR and come out with significant patterns for the upcoming research in Contextual IR (CIR).

We retain the inclination of users towards: (a) social network preferences proportionally to their own personal preferences, also (b) users concern about accuracy and time, and finally (c) shorter and thus more ambiguous queries.

In fact, the user is no longer a simple consumer of information but also involved in its production. To accelerate the production of information and improve the quality of their work, users tend to exchange documents with their social neighbourhood that shares the same interests. Therefore, the user, under the influence of his social environment, gives as much importance to the social prominence of the information as the textual similarity of documents at the query. In order to meet these new prospects, information retrieval is moving towards novel user centric approaches that take into account the social context within the retrieval process.

Thus, the new challenge of an information retrieval system is to model the relevance with regards to the search context (i.e. all the intrinsic and extrinsic elements that surrounds the user's search task). The second challenge is to provide accurate documents with a relevance that reflects as closely as possible the user's information need formulated by his or her query. It is in this specific context that fits our work. Our goal is to estimate the relevance of documents by integrating the contextual characteristics of the research task as well as the latent semantic in documents and queries.

To address this issue concerning the contextual factors that affects the search task, we have explored three related questions:

- How to make the query-document mapping process more efficient and the IRS' results more relevant, by integrating Natural Language Processing Technique to compute similarity beyond the terminological resemblance of the terms contained in queries and documents?
- How to index accurately the content of the document by integrating what is, according to the survey, the most important contextual factors "The user" and "The document" (i.e. the social dimension of the user and the content of the documents as well as the queries)? And finally

- How to make the content of the document readily available right after its publication, by proposing an on-the-go indexing technique?

In this thesis, we present a real effort in modelling a semantically enriched and context-aware Information Retrieval System with a new indexing technique and a new query-document mapping technique.

3. Research challenges

IR has become indispensable to our modern knowledge-based society. Modern information environments are becoming large and complex as well as ubiquitous, because the amount of available heterogeneous information grows exponentially each year. Almost every aspect of our lives and every profession are affected by the information available on the Internet.

In recent years and with the fast growth of the World Wide Web and the difficulties in finding desired information, efficient and effective information retrieval systems have become more important than ever, and the search engine has become an essential tool for many people.

This dissertation answers the question: How the query-document mapping process can be more effective and accurate for IR purposes? Specifically, it investigates the following thesis: Context elements (i.e. Content of the query and documents and the social dimension in our case) help considerably to improve the results provided by an IRS so they fits with user's need of information.

Moreover, in this thesis, a new perspective is provided to address the problem mentioned above. In particular, we focus on proposing new models to index documents and queries, to compute the similarity between them and finally to propose the most relevant possible results. Thus, the contributions of this thesis can be organised into five categories (1) Study of the importance of the inclusion of a contextual dimension to enhance the relevance and effectiveness of a search task, (2) Study of the possibility of a prospective standardization of context models, (3) Proposition of a semantically enriched context-aware stemming algorithm,

(4) Proposition of a folksonomy-based indexing algorithm, (5) Proposition of an on-the-go indexing technique.

4. Author's contributions

The contributions of this thesis can be summed up as follows:

- Nowadays, a search task is no more concerned with a query and a set of documents only, but it is related to a wide range of some extrinsic and intrinsic factors, so called "context", which became a great challenge these last few years. We conducted a survey with 434 internet users to understand their search trends and habits.
- Whereas the majority of works and research about context-awareness in ubiquitous computing provide context models that make use of context features in a particular application, one of the main challenges these last years has been to come out with prospective standardization of context models. As for Information Retrieval, the lack of consensual Context Models represents the biggest issue. In this thesis, we investigate the importance of good context modelling to overcome some of the issues surrounding a search task. Thus, after identifying those issues and listing and categorizing the modelling requirements, the objective was to find correlations between the appreciations of context quality criteria taking into account the user dimension. Likewise, the results of the online survey about search habits have been used such that many socio-demographic categories were considered and the Kendall's W evaluation performed together with the Friedman test provided very interesting results that encourage the feasibility of building large scale context models.
- We also proposed a modified version of the Context-aware Stemming algorithm itself based on the well-known Porter stemmer in an effort to maximizing the proportion of the meaningful stems and thus, the search effectiveness without compromising the other performance measures. Several stemmers were studied and a synergetic hybrid solution was proposed. Indeed, the Semantically Enriched Context-Aware Stemming algorithm (SECAS) combines features from algorithmic stemmers and dictionary stemmers with respect to conceptual indexing techniques in order

to improve retrieval performance; proposing root words much comparable to lemma. The experimental results conducted with the WT2G dataset show that our algorithm is noticeably more efficient; enhancing precision (up to 300%) as well as recall (up to 700%) as compared to Porter and CAS algorithms.

- Having proved that, nowadays, the opinion of the social network (or the environment) of a user has become as important as the opinion of the user himself in a given search task, we wanted to investigate this matter by exploring social-bookmarking as a new indexing technique through the use of Folksonomies as a new alternative to Ontologies in knowledge representation for IR purposes. The socialBM0311 (large scale social tagging dataset) was used to evaluate the proposed algorithm, whose resulting execution time and index size were considerably shrunk as compared to the first versions of our indexing algorithm.
- Our current focus include the proposition of an on-the-go indexing technique allowing the content of document (websites, blogs, and social-media posts) to be readily available. The idea is to combine the proved power of metadata, folksonomies, and traditional indexers in a two-phased indexing algorithm. First, users' tags will be used to immediately index the content of a document, then a more elaborated indexing technique (the folksonomy-based indexing algorithm) will be used to round off the indexing process and make the content efficiently retrievable. In this regard, a thorough study of a new concept "Personomies" (that is the user's information environment that has been built in time and that includes his/her contacts, his/her purchases, his/her research history, his/her emails, his/her RSS feeds, his/her comments on blogs, etc ...) is needed so as to make the IRS more user centric. The user dimension being undoubtedly the most important element of context to better define his or her search task.

5. Author's publications

- Melyara Mezzi, Nadjia Benblidia, and Xiangji Huang, Proposition of a Semantically Enriched Context-Aware Stemming Algorithm, Journal of Integrated Design and Process Science, Vol. Preprint, No. Preprint, pp. 1-21, June 2017

- Melyara. Mezzi, Nadjia. Benblidia,"Study of Context Modelling Criteria in Information Retrieval", International Journal of Information Technology and Computer Science (IJITCS), Vol.9, No.3, pp.28-39, 2017. DOI: 10.5815/ijitcs.2017.03.04
- Melyara Mezzi and Nadjia Benblidia, Aspects of context in daily search activities: Survey about nowadays search habits. DOI: 10.5220/0005480706270634, In proceddings of the 11th International Conference on Web Information Systems and Technologies (WEBIT-2015), pages 627-63.
- Messaouda Fareh, Omar Boussaid, Rachid Chalal, Melyara Mezzi, Khadidja Nadji. Merging Ontology by semantic enrichment and combining similarity measures, Int. J.Metadata, Semantics and Ontologies, Vol. 8, No. 1, 2013.

6. Thesis structure and outline

This dissertation is divided into seven chapters, preceded by a general Introduction about the background research, the thesis statement and motivations and concluded by directions for future work. The Chapter breakdown is as follows:

- Chapter 1 reviews important concepts and terminology in IR that complete an understanding of the challenge and motivations behind IR. Fundamental concepts in IR are reviewed, including IR approaches, strategies and models;
- Chapter 2 provides background for the Text-based IR Systems. It addresses the text-search problems, the searching techniques, and the reference model for search;
- Chapter 3 addresses the special case of web-based search. In this chapter, a detailed comparison between classical search and web-based search is made. Then, a study of the most influential search engine is provided. After that, the web Information Retrieval Systems and their challenges are presented;
- Chapter 4 explores the concept-based IR notion. First, the semantic web vision is presented. Then, the structure of the semantic web is given along with the specificities of conceptual IRS;

- Chapter 5 evaluates the most important contextual factors that affects the search tasks. First, the context significance in IR is studied. Then, the issues of the Information Retrieval task are presented. After, that the context modeling is introduced together with a detailed study of the modeling requirements. Finally, our online survey is presented together with its Kendall's W evaluation;
- Chapter 6 outlines our proposition of a semantically enriched context-aware stemming algorithm. First, the concept of stemming is presented together with the underlying stemming techniques and problems. Then, an overview of the SECAS algorithm is given with the experimental results. After that, we present the similarity computation module and discuss the obtained results;
- Chapter 7 tackles the Folksonomy-based indexing algorithm. First we talk about Knowledge Organisation Software and the emergence of the social web. Then, we introduce social tagging and folksonomies, theirs types and characteristics as well as a comparison with traditional taxonomies. Finally, we present our Folksonomy-based indexing technique and the turtle-dove matching technique which contribute greatly to the compression of index size and the reduction of the execution time.

CHAPTER 1 INTRODUCTION TO INFORMATION RETRIEVAL

1.1. Introduction

Information retrieval was concerned over the last 70 years with the problem of retrieving information from large bodies of documents with mostly textual content, as they were typically found in library and document management systems [1]. Thus, the area was perceived as being one of narrow interest for highly specialized applications and users. The advent of the WWW altered this opinion totally, as the web is a worldwide warehouse of documents with universal access.

Nowadays, the volume of information being created, generated and stored is becoming huge. Without adequate knowledge of Information Retrieval (IR) methods, the retrieval process for information would be cumbersome and frustrating [2]. Therefore, with more than one billion people accessing the Internet, and billions of queries being issued on a daily basis, modern Web search engines are facing a problem of daunting scale[1]. The main problem associated with the existing search engines is how to avoid irrelevant information retrieval and to retrieve the relevant ones.

This chapter presents a brief overview of IR basic concepts. Specifically, the history of IR, the motivations behind the blooming of Information retrieval and its main features, approaches, and strategies.

1.2. Basic concepts

The importance of IR keeps growing as the amount of digital information keeps expanding at an ever-increasing rate [2]. But, these stored documents, photographs and contents of books, and billions of Web pages are useful only if they can be easily found when needed.

[1] According to http://www.internetlivestats.com, Google, at the moment of writing, processes over 63,000 search queries every second on average, which translates to over 4.7 billion searches per day worldwide.

Information seeking behaviour is rooted in a need to find information. According to Saracevic [17], information is anything that can change person's knowledge. Thus, the Information Retrieval (IR) process begins with an anomalous state of knowledge (ASK). Then, many changes in knowledge state are involved. In short, IR is a purposeful process that alters the state of knowledge reacting to an information need or gap. A simple vision of an Information Retrieval System (IRS) was believed to be as follows [3]:

- The user expresses an Information need by formulating a question (called query);
- The IRS answers the query and gives back results (texts, images, videos, etc.);
- The final phase is up to the user who has to evaluate and reformulate her query if the results do not satisfy her request.

Today, this vision became somehow obsolete, because the users, their queries, and the desired information were believed to be static. So, the relevance of a document was computed statically between the query and the set of documents ignoring the user, the device, the environment, and the specificities around the search activity which constitute the search context and are as a matter of fact highly variable factors. Besides, with the technology advances, information can nowadays, be accessed everywhere and at any time which add to the variability and the uniqueness of each search situation. And as no information is context free, the inclusion of a contextual dimension in the classic IR process became a real challenge.

1.2.1. Definition

Information Retrieval (IR) is the science of searching for information in documents, searching for metadata which describe documents, or searching within databases, whether relational stand-alone databases or a hypertextually networked database such as the World Wide Web [4]. The core functionality of an IR system is the retrieval of data from a database

whose abstraction matches the description of an ideal object, inferred from a query.

IR used to be an activity that only a few people engaged in [5]: reference librarians, paralegals, and similar professional searchers. Now the world has changed, and hundreds of millions of people engage in information retrieval every day.

Actually, IR is a very broad field, containing topics on representation, storage, retrieval, ranking, evaluation, etc. of various media types, such as web pages, images, and videos [6-7]. In this section, we focus on reviewing relevant work on retrieval models for text-based documents as they are the underpinning of our thesis work.

Before moving on, we first introduce some terminology[2] [5, 7]***:***

- A *document* in information retrieval refers to the unit used in the indexing and retrieval process. It can be of different media types or at different levels of granularity for a given type (e.g., books, chapters, paragraphs, and sentences for text-based documents).
- A *term* is the basic element that constitutes a text-based document in our case.
- A *collection* is a set of documents used to address users' requests. Each request is an information need, i.e., a topic the user desires to know more about.
- A user communicates an arbitrary information need via a *query* to the search engine.
- A relevant document is the one that the user perceives as containing information of value with respect to their personal information need.

[2] More trivial definitions can be found in the glossary of this thesis.

1.3. Principle of IR

An Information Retrieval System (IRS) attempts to retrieve from a collection of documents, those relevant documents that correspond to a user's request. Models of information retrieval systems are characterized by three main components [4]: the representation of documents, the query language, and the matching mechanism.

- The documents' representation is generated by the indexing process which represents the content of a document as indexed-terms;
- Users submit their information needs to the system as queries expressed in the system query language;
- Then, a matching mechanism evaluates a user's query against the representations of documents and retrieves those documents which are considered to be relevant.

1.4. Historical view

IR is a well-established research area in computer science. The idea of IR is credited to Vannevar Bush after publishing his essay: "As We May Think" in 1945. Bush introduced a concept of IR system called as Memex that enables individuals to read and write content on a large scaled data. He described that Memex would operate as an indexed repository of knowledge and carry out a sequence of work faster than human experts [8]. This essay has significant influence on contemporary researchers seeking relevant information from various resources such as text, audio, and images. Since then, a great deal of effort to improve IR strategies has been exerted by many researchers.

Thus, Information retrieval -as we might think- did not begin with the Web [5]. In response to various challenges of providing information access, the field of information retrieval evolved to give principled approaches to searching various forms of content.

The field began with scientific publications and library records, but soon spread to other forms of content, particularly those of information

professionals, such as journalists, lawyers, and doctors. Much of the scientific research on information retrieval has occurred in these contexts, and much of the continued practice of information retrieval deals with providing access to unstructured information in various corporate and governmental domains.

For thousands years, people have realized the importance of archiving and finding information. With the advent of computers, it became possible to store large amounts of information; and finding useful information from such collections became a necessity. The field of Information Retrieval (IR) was born in the 1950s (as recalled by Mooers (1960)), out of this necessity [9]. But research in this area has been actively pursued for at least the last 100 years. Developed at the end of the 19th and the beginning of the 20th centuries, the first automatic retrieval systems used mechanical solutions to speed up lookup in library catalogues [10], and over the last sixty years, the field has matured considerably. Several IR systems are used on an everyday basis by a wide variety of users. Likewise, Information retrieval has undoubtedly become one of the most important research area in the field of computer science.

Figure 1.1 sums up the major steps in IR evolution throughout the last 70 years.

Research and industry efforts in IR bifurcate into two areas; the first being system-oriented research and development, and the second a user-oriented [11]. These concept will be further detailed in the upcoming subsections.

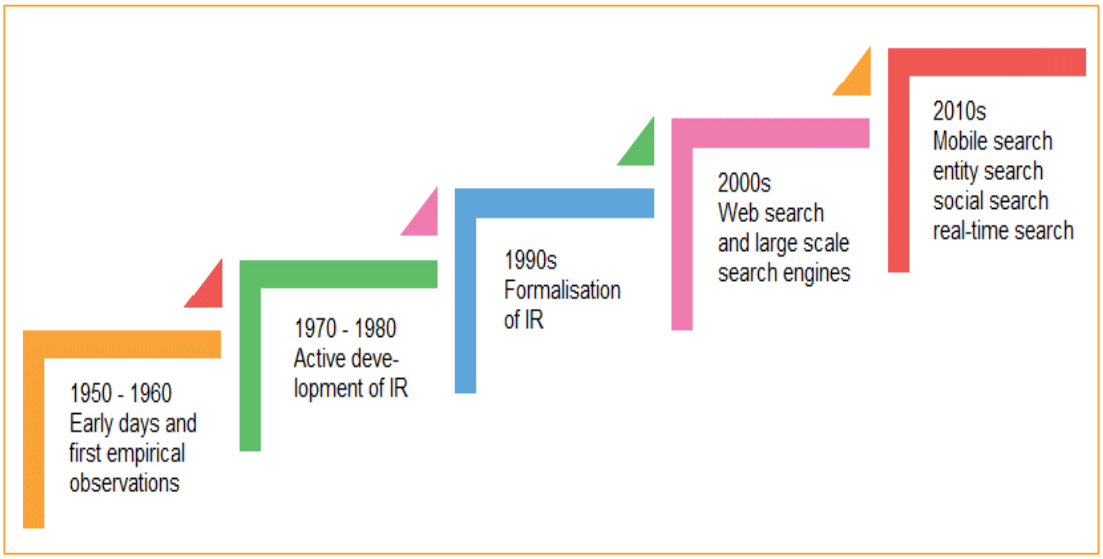


Figure 1-1: Evolution of Information Retrieval [12].

- *1950 - 1960:* Early days and first empirical observations.
  - Hypothesis on automated indexing (LUHN).
  - First experiments and development of guidelines for IR systems evaluation (CLEVERDON's Cranfield 1 and Cranfield 2).
  - Early experiments on the Vector Space Model for ranking (SALTON's SMART system).
- *1970 – 1980:* Active development of IR.
  - The establishment of the Vector space Model for ranking.
  - Ranking models based on Probability Ranking Principle (PRP).
- *1990s:* Further development and formalization of IR.
  - New applications and theoretical explanations.
  - Statistical Language Model (CORFT 1998); Development of large scale collection for IR systems evaluation (TREC).
- *2000s:* Web search, large scale search engines in the wild, anti-spam.
  - Machine learning to rank.
  - MapReduce, GPS, Hadoop.
- *2010s:* Mobile search, entity search, social search, and real-time search.

1.5. Motivations behind IR

The goal of an information retrieval system is to find information that meets the end user's information need. Broadly information retrieval is defined as "*a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information*" [13].

In other words, IR aims to find relevant information resources to a query from a collection of information resources. Queries are statements of information needs, and are usually formed as a series of keywords. An automated information retrieval system takes the query as input and outputs a ranked list of documents with different degrees of relevancy. Due to the purpose of effectiveness and efficiency, the documents in the collection are usually pre-processed into their indexed representations, and the queries are prepossessed into the corresponding representations [14]. *Figure 1.2* shows a basic IR system, where an IR weighting model matches documents' representations with a query representation and generates a list of relevant documents. In our work, we focus more on the modelling phase of the retrieval system, and propose new IR models to promote the retrieval performance, which is, providing more relevant documents.

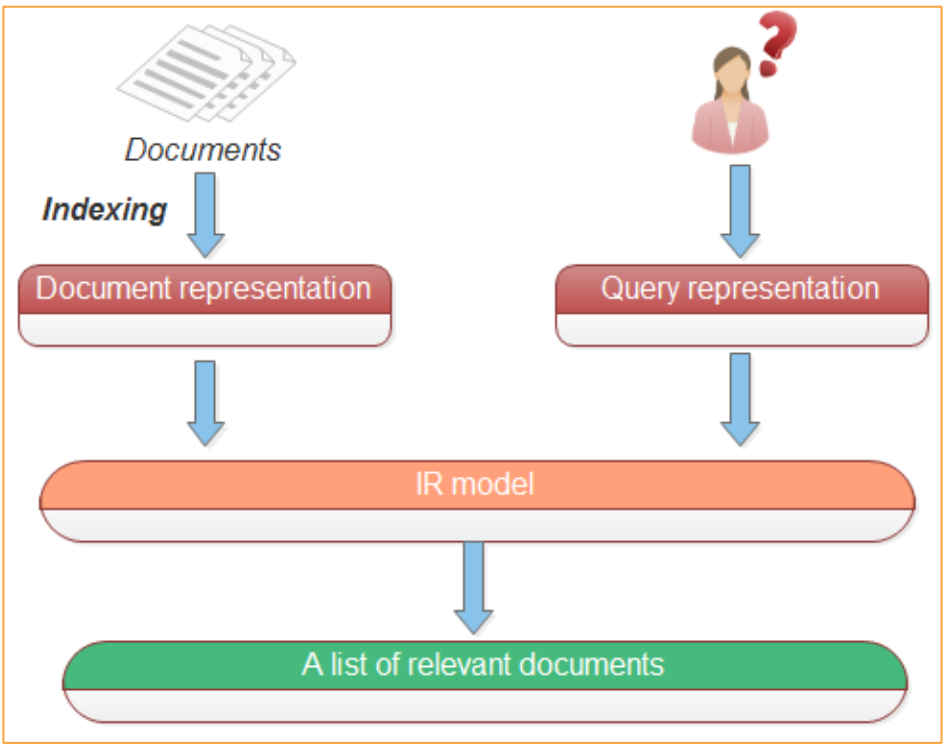


Figure 1-2: Basic Information Retrieval system [14].

1.6. Features of the Information Retrieval Task

Whether it is looking for a contact email address, finding the lyrics of a music track on one's smartphone, using a Web search engine to find a recipe with certain ingredients, or trying to find out the birth place of a

famous figure, these ostensive examples of IR are undoubtedly convenient since IR has become an important part of (almost) everyone's everyday life [15].

As stated by Saracevic, the information can be [17]: (a) Objects in the world potentially conveying information, (b) What is transferred from people or objects to person's cognitive systems, or (c) Components of internal knowledge in people's mind.

Furthermore, according to Han, Wang, M., Wang, J. [16], the request for information can either be external or self-initiated. In the same ground, Saracevic talked about direct (end-user) search and mediation search [17]. Direct searchers are people who seek information by and for themselves, whereas in mediation search, there is an intermediary who acts on the behalf of a person who is actually seeking for information. The mediation can either be informal when it comes to search information for colleagues, family, and friends, or formal when it comes to search for information as a searcher or a teacher. Moreover, two kinds of search are noticeable as reported by [18-20]: (a) Navigational (evidential, or pull-based) search, and (b) Thematic (informational, or push-based) search.

Navigational search deals with aware users having steady needs. In this case, IR is explicit and the process consists of comparisons with previous knowledge. Whereas, in thematic search, the user inputs the query that explains or describes information related to that the user wishes to collect or research [3]. Hence, IR is implicit and the process consists of seeking for new knowledge whether the needs are known, unknown and poorly defined, or changing. *Table 1.1* shows the advantages and disadvantages of each kind.

Table 1-1: Navigational search VS thematic search [3].

	Advantages	Drawbacks
Navigational search	Aware users Clear needs	Overcommitted users
Thematic search	Smoother experience	Fuzzy

Finally, Keywords and controlled vocabulary are two kinds of search terms [21]. This is why a clear distinction can be made between

Information Search and Information Retrieval. The first one concerns navigational search, whereas the second concern thematic search. In the remainder of this thesis, we treat search and information retrieval as synonymous concepts.

1.7. Information behaviour

With ideal human behaviour, we mean that users make no errors during the search process, or to be more precise, users scan all documents one after another, click every relevant document without making any judgment errors, read them and judge their relevance correctly.

In contrast, fallible human behaviour means that users may well err during the search process [11]. In other words they may skip some relevant documents, read non-relevant ones, judge them as relevant or judge the relevant ones as non-relevant by mistake.

Major differences in Information seeking behaviour are that, when looking for information as a searcher, the clearly defined information needs of a user remain constant to the very end, whereas when looking for information on the behalf of someone else, one's vague information needs gradually evolve [22].

Modern knowledge society would not be possible without IR, because of the ever-growing amount of information available on the Internet. While computing technology is nowadays ubiquitous, users interact with various computer interfaces with varying goals and time constraints in order to complete their tasks, which may be initiated by their work or leisure-related activity [11].

Human information behaviour consists of phenomena such as information needs, information seeking, searching, browsing, finding, judging, usage, communication, sharing, transfer, management, information habit, and information style, which in brief means any information-related human behaviour [23].

In the next sub-section, we first introduce traditional information retrieval, and then describe interactive information retrieval.

1.8. Information Retrieval Approaches

In traditional IR, the system-centred approach to information retrieval was the norm [22]. With the arrival of the Internet in the 1990s, users became able to search for information themselves using a web browser. This led to the emergence of the user-centred approach to information retrieval and now both approaches vie for supremacy.

1.8.1. System-centred approach

Information retrieval systems store and manage information items, e.g., text documents, as well as enable users to access them efficiently. By traditional Information Retrieval (IR) we mean system-oriented IR, which focuses on documents and document collections, matching algorithms to retrieve relevant information items to stated queries, and relevance judgments about documents in relation to queries.

Figure 1.3 depicts the traditional IR process, which is also called the laboratory model of IR. *Figure 1.3* is adapted from Ingwersen and Järvelin's schematized system-oriented IR Model [24]. The main focus of the system-oriented approach is the representation of documents and search requests as well as their matching process. The user's involvement is confined to relevance and possible feedback judgments. Moreover, the relevance judgments of documents were created once by persons who may be developers of the experimental environment. In this view of IR, documents are represented and stored in a database corresponding to the applied retrieval model. Thereafter, the user's information need is translated into a search request, which is in turn represented as a query for the matching process.

However, neither the task, which causes the user's information need, nor the user's real information context is taken into account in any way. Nevertheless, the matching algorithms deliver more or less relevant

documents according to the match between the presentations of documents and query.

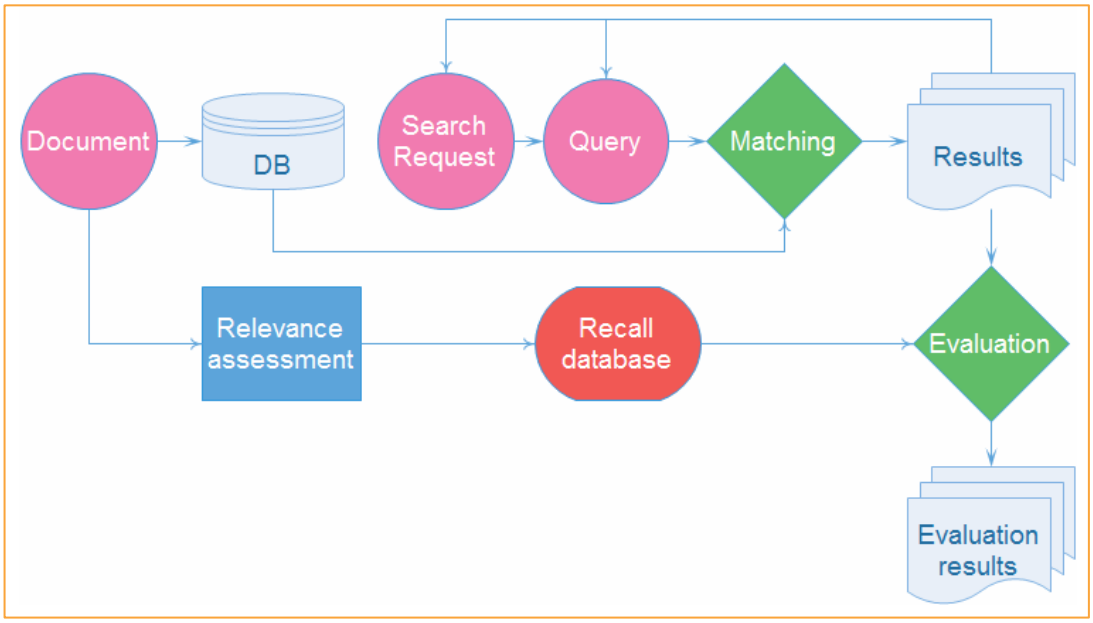


Figure 1-3: Traditional IR process [11].

In the model of system-centred information retrieval as depicted in *Figure 1.4*:

- On the left side of the diagram is a group of documents contained in a database. The indexer attaches an index, or metadata[3], to each document. In the case of traditional information retrieval, metadata are used to refer to the title of a document, the name of the authors, the name of the publisher and subject keywords, which alternatively work as its access points.
- On the right side of the diagram is the user who has information needs. The database is queried by expressing the information needs as an inquiry combining keywords.

[3] According to https://en.oxforddictionaries.com, a metadata is a set of data that describes and gives information about other data.

Thesauri, classification tables and subject headings are used as translation tools for matching the keywords used by the user in the query against the keywords attached to documents by the indexer.

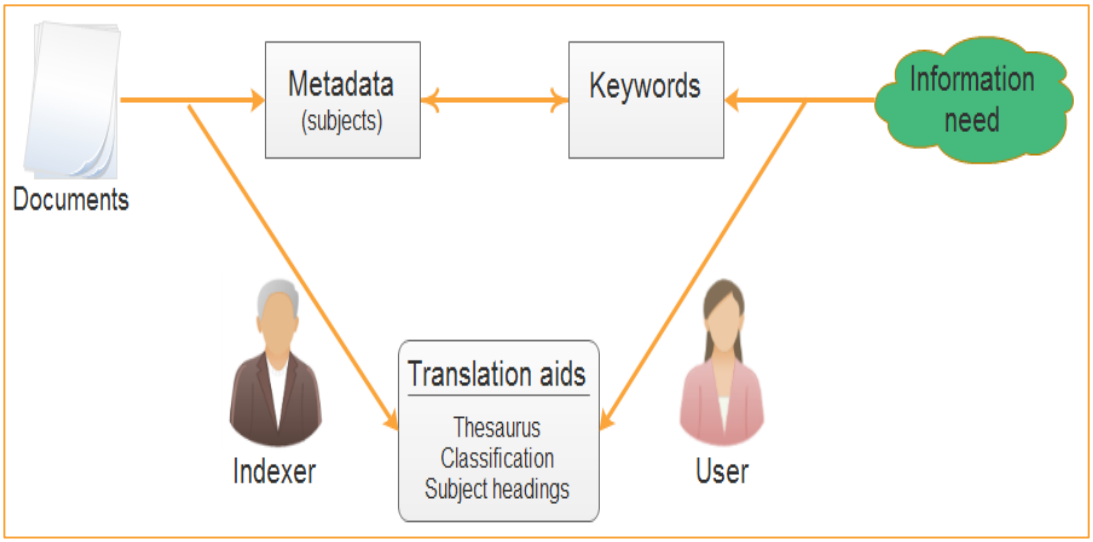


Figure 1-4: The system-centred Information Retrieval model [22].

The user represents information needs in queries using logical operations on keywords. There are three types of logical operations that can be used here: logical sum, logical product and logical difference.

Boolean operators (see *Figure 1.5*) define the relationships between words or groups of words and are used to broaden or narrow a search. Boolean operators used to qualify search parameters include [21]:

- Logical product “AND”: Narrow the search and retrieve records containing all of the words it separates;
- Logical sum “OR”: Broaden the search and retrieve records containing any of the words it separates;
- Logical difference “NOT”: Narrow the search and retrieve records that do not contain the term following it.

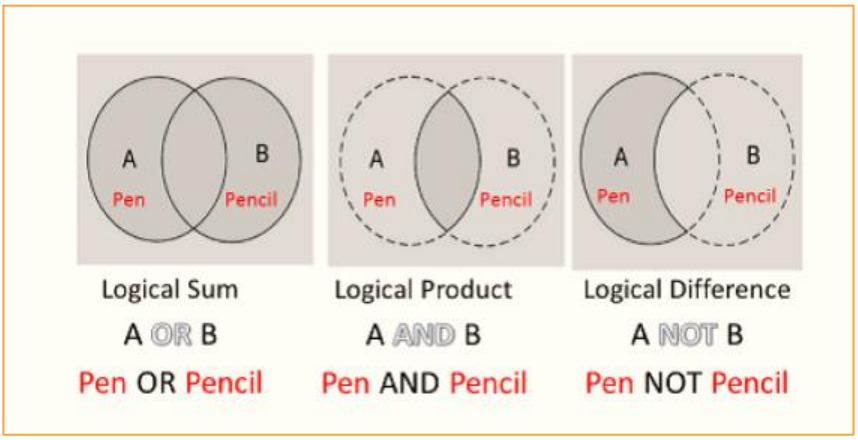


Figure 1-5: Boolean operations [22].

In figure 1.5 logical operations for the two keywords “pen” and “pencil” are presented [22]:

- In the case of a logical sum, the search will retrieve documents that contain either “pen” or “pencil” or both.
- In the case of a logical product, the search will retrieve documents that contain both “pen” and “pencil”.
- In the case of a logical difference, the search will retrieve documents that contain the keyword “pen” but which do not contain the keyword “pencil”.

In real-life searches, complex queries can be expressed by combining logical sum, logical product and logical difference operations. Moreover, in IR, the first query that is created not necessarily lead to search results that match the information needs. For this reason, a technique was derived called relevance feedback. This is a method of leading to better search results by revising the initial search results. There are three types of relevance feedback techniques: explicit feedback, implicit feedback and pseudo relevance feedback.

- *Explicit feedback* is the technique of getting the user to decide on the relevance of the documents retrieved as a consequence of the initial query, and forming the next query using the metadata of those documents that are determined to be relevant.

- In *implicit feedback*, the system detects whether the user has looked at each of the documents appearing in the search results of the initial query, and if so for how long and detects any browsing or scrolling action. Based on this, the system then determines relevance. The next query is then automatically formed using the metadata of those documents determined to be relevant. This technique is called “implicit” feedback because relevance is reckoned without the user knowing.
- In *pseudo relevance feedback*, once the set of relevant documents has been searched, the top few documents in the list of search results shown in order of relevance are judged as being highly relevant. The next query is then automatically formed using the metadata of these documents.

In this way, relevance feedback is used for deriving more relevant search results, by revising an initial query based on the content of relevant documents. In system-centred information retrieval, various other techniques are also used in order to increase the relevance of search results, such as by attaching weight to certain keywords.

1.8.2. The user-centered approach

The idea in user-centred information retrieval that “information needs change and cannot be defined clearly” differs from the premise in the system-centred information retrieval model that “information needs can be defined clearly and do not change.”

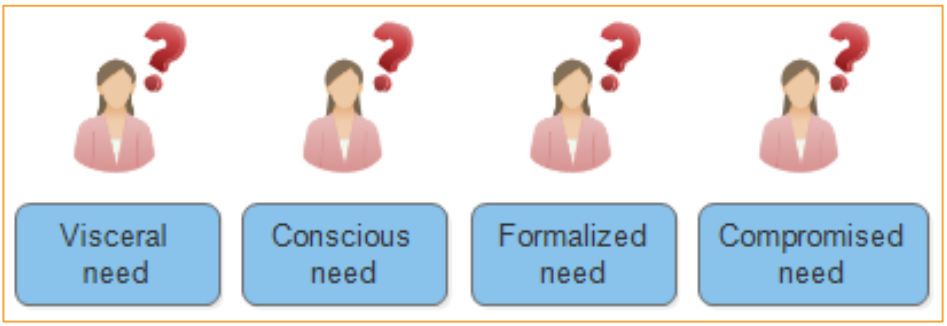


Figure 1-6: Evolving Information Needs [22].

The user-centred information search process has the following characteristics [22]:

- The information search process is a linear process that includes loops and trials and errors;
- The process starts with a broad topic, and gradually narrows down to a more focused topic;
- Information needs evolve during the information search process;
- Users continuously evaluate their own decisions;
- Knowledge structure changes during the information search process;
- Users have some strategy to end the information search process.

User-centred IR uses the concept of “exploratory search” (*Figure 1.7*). This figure compares and contrasts user-centred information retrieval against conventional information retrieval. It classifies information retrieval into three categories: Lookup, Learn and Investigate and regards searches performed for the purpose of learning and investigating as exploratory searches.

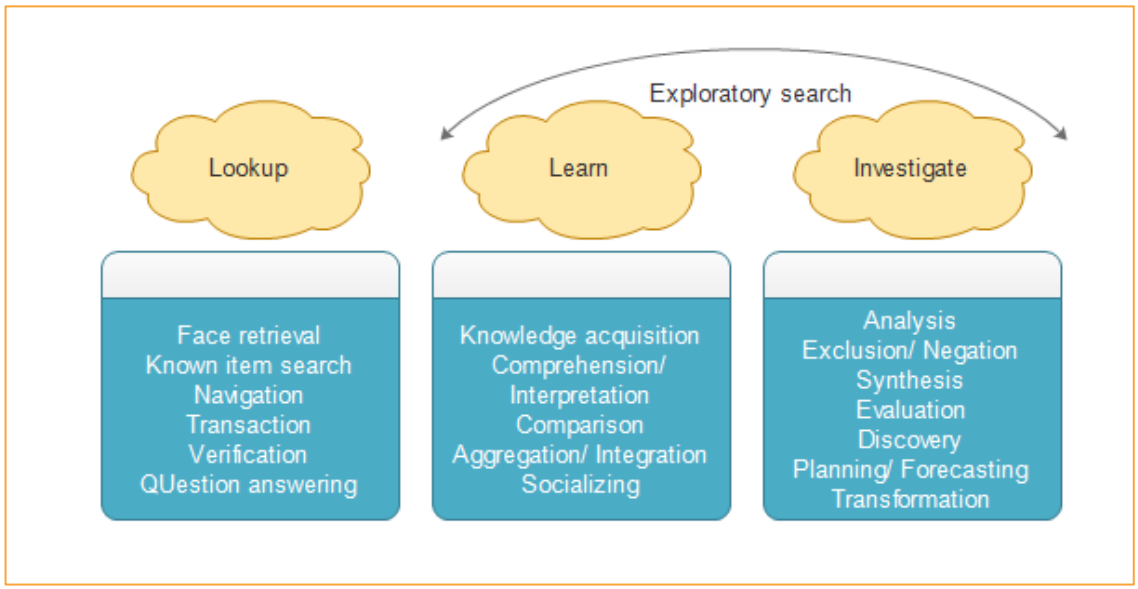


Figure 1-7: Exploratory search [22].

1.9. Information Retrieval strategies

So the relevant documents can be effectively retrieved, they are typically transformed into a suitable representation. Each retrieval strategy

incorporates a specific model for its document representation purposes. Retrieval effectiveness is dependent on the selected retrieval model [2]. A major focus of IR research has been the development of retrieval models that capture the relationship between a query and a document [10].

Indeed, IR models specify precisely how the topics and the documents are represented and how the indexing is performed [25]. IR models must also specify how these documents' surrogates are matched to the queries (the topic representations) [9, 15].

The retrieval strategies assign a measure of similarity between a query and a document [6]. These strategies are based on the common notion that the more often terms are found in both the document and the query, the more "relevant" the document is deemed to be to the query. Some of these strategies employ counter measures to alleviate problems that occur due to the ambiguities inherent in language:

- The same concept can often be described with many different terms (e.g., Informatics and Computer Sciences can refer to the same concept).
- The same term can have numerous semantic definitions (terms like bark and duck have very different meanings in their noun and verb forms).

The study of information retrieval models has a long history. And over the decades, many different types of retrieval models have been proposed and tested. In this dissertation, we will provide a basic description of the three most well-known IR models, namely: Boolean model, Vector space model, and the Probabilistic model.

1.10. A formal characterization of IR models

It is argued that the fundamental premises which form the basis for a ranking algorithm determine the IR model. Throughout this section, we will discuss different sets of such premises. However, before doing so, we should state clearly what exactly an IR model is.

An information retrieval model can be considered as a quadruple [D, Q, F, R (qi, dj)] where [4]:

- D is a set composed of logical views (or representations) for the documents in the collection.
- Q is a set composed of logical views (or representations) for the user information needs (i.e. queries).
- F is a framework that models document representations, queries, and their relationships.
- R (qi, dj) is a ranking function which associates a real number with a query qi Q and a document representation dj D. Such ranking defines an ordering among the documents with regard to the query qi.

To build a model, representations for the documents and queries should be first made. Given these representations, the framework in which they can be modelled is conceived. This framework should also provide a way to compare (map) these representations and then rank the obtained results.

The *Figure 1.8* shows various IR models, where, the components D, Q, F, and R (qi, dj) of each model are quite clear and can be easily inferred.

Belkin and Croft [26] classify retrieval models into two main branches, namely exact and partial matching models. Even though they have disadvantages to some extent, these two models are important to the history of IR and all popular models are forms of their extensions [11, 26].

Exact matching models are developed on the basis of Boolean algebra. For this type of model, queries are meticulously constructed with the help of Boolean operators. On the other hand, in order to allow partially matching documents to be listed on the results list, partial matching models such as Vector Space Models (VSM), probabilistic

retrieval models and more recently probabilistic language models have been developed.

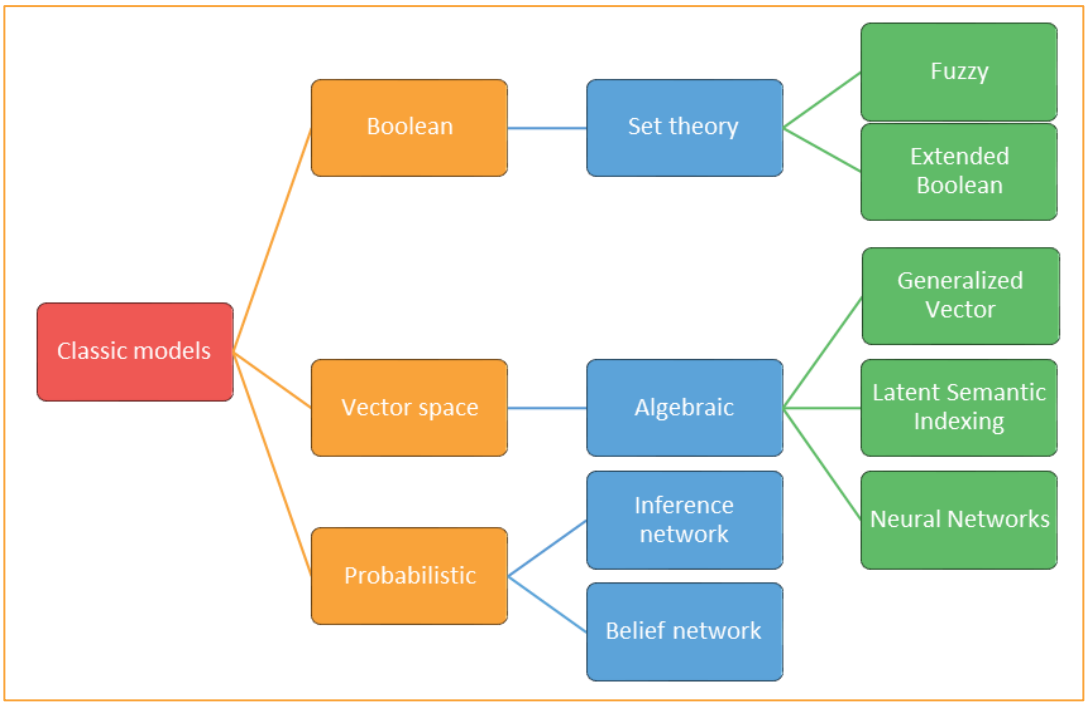


Figure 1-8: Taxonomy of IR models.

1.10.1. Boolean model

Boolean logic-based retrieval models only deem those documents which exactly match the Boolean query as relevant, for example; the results of the query "Information AND Retrieval", must include both keywords [11]. Indeed, the simplest model as well as the earliest in information retrieval is the Boolean model. In IR Boolean model, users are allowed to formulate queries in the form of logical clauses, and retrieve the set of documents that match the query [7, 25, and 27].

The Boolean Information Retrieval (BIR) is based on Boolean logic and classical set theory in that both the documents to be searched and the user's queries are conceived as sets of terms. Retrieval is based on whether or not the documents contain the query terms. Query terms are combined with three basic operators, the logical product AND, the logical sum OR and the logical difference NOT [9, 14] as seen in section 9.1.

Advantages and disadvantages of BIR can be summed up in *Table 1.2* [2, 8, and 14].

Table 1-2: Advantages and disadvantages of the Boolean Model.

Advantages
– Exact match: a document would match or do not match the query; – The Boolean model gives users a sense of control over the retrieval system; – The model is robust; – Results are predictable and relatively easy to explain
Disadvantages
– The notion of document ranking does not exist in a Boolean IR system. The retrieved documents are either scored 0 or 1; – Hard for untrained users to manipulate it correctly; – All terms are equally weighted; – The model is very strict and exact matching may retrieve too few or too many documents; – Complex queries are difficult to write; – Lack of knowledge about how to utilize its search possibilities; – The model views each document and query as just a set of words.

Despite decades of academic research on the advantages of ranked retrieval, systems implementing the Boolean retrieval model were the main or only search option provided by large commercial information providers for three decades until the early 1990s (approximately the date of arrival of the World Wide Web). Even if Boolean systems seem to be obsolete nowadays, still some specific domains like legal domain can require recall-oriented retrieval, which can be provided by Boolean systems.

Limitations of this model led to the development of weighting schemes that allowed users or systems to assign weights to individual terms indicating their importance.

1.10.2. Vector space model

The Vector Space Model (VSM) was first realized in Salton's Smart information retrieval system [28]. VSM represents both documents and queries as vectors in multidimensional space, whose dimensions consist of keywords. Every vector representing documents and queries can be built up with term weights [6, 11].

One of the most influential weighting schemes developed for the VSM is TF-IDF[4]; (TF: Term Frequency is the number of times the term occurs in the document and IDF: Inverse Document Frequency is the inverse of the number of documents in the collection in which the term occurs) [9-10, 25-27].

The similarity between a document and a query is computed (with the cosine[5] similarity measure for example), which gauges the angle between two vectors. Then the documents can be ranked according to the cosine values in a descending order. VSM is based on vector algebra, and is therefore mathematically founded, whereas its applicability in IR may be arguable from the justification point of view.

Advantages and disadvantages of VSM models can be summed up in *Table 1.3* [2, 5, 7-9, 14-15, 29].

Table 1-3: Advantages and disadvantages of the Vector Space Model.

Advantages
– In the vector space model, users largely use *free text queries*, that is, just typing one or more words rather than using a precise language with operators for building up query expressions (user-centred models); – The Vector Space model assigns non-binary weights to index terms in queries and in documents; – the Vector Space model is mathematically founded computes a continuous degree of similarity between queries and documents and supports partial matching; – The vector space model can best be characterized by its attempt to rank documents by the similarity between the query and each document; – This formulation prevents the retrieval system from favouring short documents over their longer counterparts; – Very simple similarity measures or term weighting schemes can be used; – Challenge is mostly finding good weighting scheme; – Tends to work quite well in practice despite obvious weaknesses.
Disadvantages
– The main disadvantage of the Vector Space model is that it does not define appropriate values to the vector components; – There is an assumption of term independence; – VSM often takes a lot of time to compute a high dimensional space in which a huge amount of different terms exists. Moreover, VSM ignores semantic relationships between terms and does not preserve any sequential order in a

[4] TF reflects the intuition that key terms conveying the meanings of a document tend to occur frequently within that document. IDF estimates how discriminative a term is.

[5] Other similarity measures include, but are not limited to: Inner product, Jaro, Dice, and Jaccard.

given document;
- Missing semantic information (e.g. word sense); Missing syntactic information (e.g. phrase structure, word order, proximity information);
- Does not consider the link structure of documents;
- Vector space models were the focus of most IR research in the 1960s and 1970s. They are typically less effective than modern alternatives but are still in use, in part for their simplicity and intuitive appeal.

The VSM procedure can be divided into three stages:

- The first stage is the document indexing where content bearing terms are extracted from the text-document;
- The second stage is the weighting of the indexed terms to enhance the retrieval of the relevant documents;
- In the last stage, the documents are ranked according to the value given by the chosen similarity measure.

In this dissertation, we propose a variant of Vector space models that takes into account semantic as well as contextual information and focus its ranking on an elaborated semantic similarity measure.

1.10.3. Probabilistic inference models

Probabilistic inference models apply concepts and techniques originating from areas such as logic and artificial intelligence [4]. Moreover, Probabilistic Retrieval Models (PRM) are based on probability theories, especially the Probability Ranking Principle (PRP), which means ranking by the decreasing probability of relevance of documents to a query. Documents can be ranked by the proportion of the probability of relevance and the probability of non-relevance [6, 11, 14, and 25].

In fact, Whereas Maron and Kuhns introduced ranking by the probability of relevance, it was Stephen Robertson in the late 1970s who turned the idea into a principle. He formulated the probability ranking principle, which he attributed to William Cooper as follows: ”Documents and queries are represented by binary vectors ~d and ~q, each vector element indicating whether a document attribute or term occurs in the document or query, or not.” [9].

Advantages and disadvantages of PRM can be summed up in *Table 1.4.* [7, 29].

Table 1-4: Advantages and Disadvantages of the Probabilistic Model.

Advantages
– Probabilistic methods have been shown to perform well on a variety of tasks, including Ad hoc retrieval, cross-lingual information retrieval, distributed IR, query difficulty prediction, passage retrieval, etc; – Highly competitive and widely used today due to their strong theoretical foundation in reasoning about uncertainty.
Disadvantages
– They do not comprehensively model the retrieval process.

1.10.4. Discussion

The above-mentioned models are the most studied IR models in the literature. However, a large number of other models have been investigated and used in prototypical IRS and fall under the name of "Alternative Models" they represent extensions of the classic ones as shown in the *Figure 1.8*. Some of these models are related to the so called *Soft Computing paradigm* or more specifically *Soft Information retrieval* [4].

Each of the above-mentioned models determine how a query is processed and how results are matched and ranked. They each have various strengths and weaknesses, a discussion of which is beyond the scope of this chapter. Modern search products typically blend elements of each model. For example, the popular open search platform, Lucene[6], uses both vector space and Boolean.

We can note that all of these models struggle with the inherently ambiguous aspects of human language, especially polysemy and synonymy[7]. In this dissertation, we propose a model that tries to overcome

[6] https://lucene.apache.org/core/

[7] Polysemic words have different meanings but are spelled and may be pronounced the same (e.g. "bank" as in "river bank" and "Federal bank"). Synonyms are different words with the same or similar meaning (e.g. example, "engine" and "motor"). Polysemy and synonymy undermine precision and recall of search results.

these linguistic issues by matching queries to documents based on meaning rather than keywords.

1.11. Conclusion

This Chapter reviewed concepts, techniques and classic models in IR most of which primarily tackle the field in its broadest meaning. The next Chapter considers the notion of Information Retrieval System, we will talk in detail about the main components of an IRS as well as the evaluation of IR models. The focus will be on text-based IR.

CHAPTER 2 TEXT-BASED INFORMATION RETRIEVAL SYSTEMS

2.1. Introduction

Since at the present time the majority of the information content is at rest existing in textual appearance, text is an important basis for information recovery [1]. Unfortunately, text carries a set of meaning, which still cannot completely be captured computationally. Consequently Information Retrieval (IR) methods are based on powerfully simplified methods of text processing, ignoring the majority of the grammatical formation of text and reducing texts fundamentally to the terms they include [1]. This approach is called full text retrieval and is an oversimplification that has verified to be very successful.

An information retrieval System (IRS) is designed to retrieve any documents or information required by the user community. It is primarily targeted to make the right information available to the right user at the right time [2]. IR is a discipline that deals with the retrieval of unstructured data or partially structured data, especially textual documents, in response to a set of query or topic statement(s), which may itself be unstructured.

From a schematic perspective, every IRS consists of three components [30]: (a) collection to be indexed, (b) the user's request (generally keywords), and (c) the matching algorithm (see *Figure 2.1*).

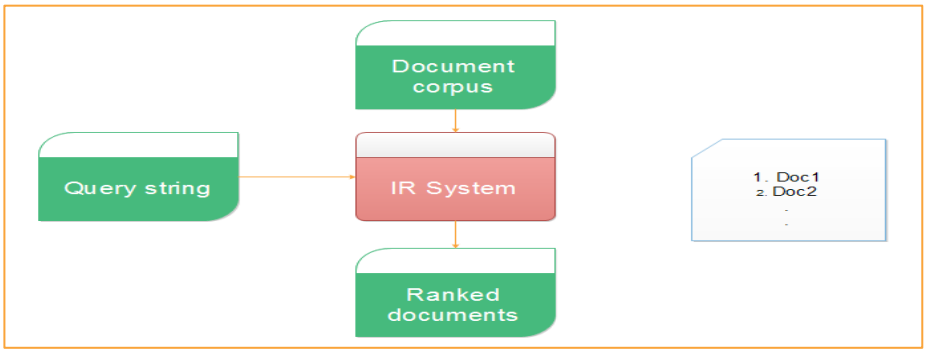


Figure 2-1: Basic view of the IRS [30].

2.2. Text-based information retrieval

"*Text Information Retrieval is a multidisciplinary field, involving information retrieval, information extraction, text analysis, clustering, visualization, categorization, machine learning, database technology and data Information Retrieval*" [31].

Text Information Retrieval is also known as text data mining or knowledge discovery from textual databases. It refers to the method of extracting interesting and non-trivial forms of information from text documents. Indeed, the normal form of accumulated information is text, text Information Retrieval is believed to include a viable potential higher than that of data Information Retrieval[8]. In truth, 85% of a company's information is contained in text documents [1].

Traditionally, there was a clear separation between structured data, typically stored and accessed via relational database management systems, and semi-structured data such as text, typically stored and accessed via IRSs [6]. Each processing system supported its own data storage files and access methods. Today, the distinction between structured and semi-structured data is quickly vanishing. In fact, we no longer are concerned with just structured and semi-structured data, but also images, audio files, and videos in the same storage repository.

[8] Unlike Information Retrieval, Data Retrieval deals with structured data (databases) with well-defined semantics.

The main problems of text-based Information retrieval systems are [30]:

- They may not retrieve relevant documents if they include synonymous terms (e.g. “restaurant” vs. “café” or “PRC” People’s Republic of China vs. “China”).
- They may retrieve irrelevant documents that include ambiguous terms (e.g. “bat” baseball vs. mammal or “bit” unit of data vs. act of eating, or “Apple” company vs. fruit).

2.3. Searching techniques

There are different searching techniques, including linear search, brute force search and binary search. These searching techniques are described as follows [25]:

- *Linear search technique*: It is a basic technique of finding a particular word or keyword from a list of words or array that checks presence of every element in list, one at a time and in a sequence. This search technique is the simplest search technique. Disadvantage of linear search is that its searching speed is very poor or slow especially in case of ordered list. This type of search is also called as *Sequential Search*.
- *Brute force search technique*: It is a very common problem-solving technique that consists of consistently itemize all possible participants for the solution and determine whether each participant gratify the problem’s statement.

  This searching technique is simple to apply and it will always return a solution if it exists.
- *Binary search technique*: It finds the position of a particular input value that is, “*the search key*” within an array sorted by some key value. For binary search technique, the given array should be arranged in some order that is, ascending or descending. In each step, this method examines the search key value as respect with the middle element key value of the given sorted array. If the value of both keys matches, then a matching item has been

found and it should be indexed. Differently, if the search key value is less/greater than the middle element's key value, then the method repeats its steps on the sub-array to the left/right of the middle element. If the leftover array to be searched and it is found empty, then the search key cannot be found in this empty array and a particular bit of string is returned that is, Not Found.

2.4. Usage area of IR systems

IRs were initially developed to improve and manage the large amount of data or information. Many private or government universities, corporate sector, and public libraries nowadays use IR systems to provide access to the different information like books, journals, and other documents. Now information retrieval is frequently used in so many applications and some common applications of information retrieval system are defined as follows [25]:

- *Web Search Engine:* One of the most practical applications of information retrieval system is a search engine and it is meant for retrieving relevant information from a large or big size text collections. They are best-known examples of IR system, but various searches exist, like: Desktop search, Enterprise search, Unify search, Mobile search, and Social search.
- *Multimedia search:* This type of Search can be applied by multi-modal search interfaces means this include some other type of media also for getting information which is different from textual search for example an image retrieval system is a multimedia search system in a computer for browsing, searching and retrieving images from a huge collection of digital images.
- *Digital Library:* A digital library is a type of library in which collections are stored in digital formats and these formats are accessible by computer systems. The digital information may be stored locally, or accessed remotely via computer network systems.
- *Information Filtering:* Information filtering system consists of many tools that help user to retrieve the most relevant and valuable

information. Information filters are also used to manage and structure information in a right and intelligible way, in favour to cluster messages on the mail addressed.

An information retrieval system (or search engine) is like a box which receives a query (which represents user's information need) and returns a list of closest documents. This box includes generally four components: indexing scheme, similarity measure, threshold, and corpus. Indexing scheme is a representation model of relations between terms and documents. This scheme allows the search engine to represent documents of the corpus in a way that facilitates the research process [32].

2.5. Reference model for search

The starting point for the search process is a user and an information need. The need is expressed in a query. The search engine interprets or processes the query and selects candidate responses from its index. The responses or results are ranked and returned to the user. The user evaluates the results and if satisfactory, the process stops here. Otherwise, the user may elect to reformulate the query and assess another set of results (this process is depicted in *Figure 2.2*).

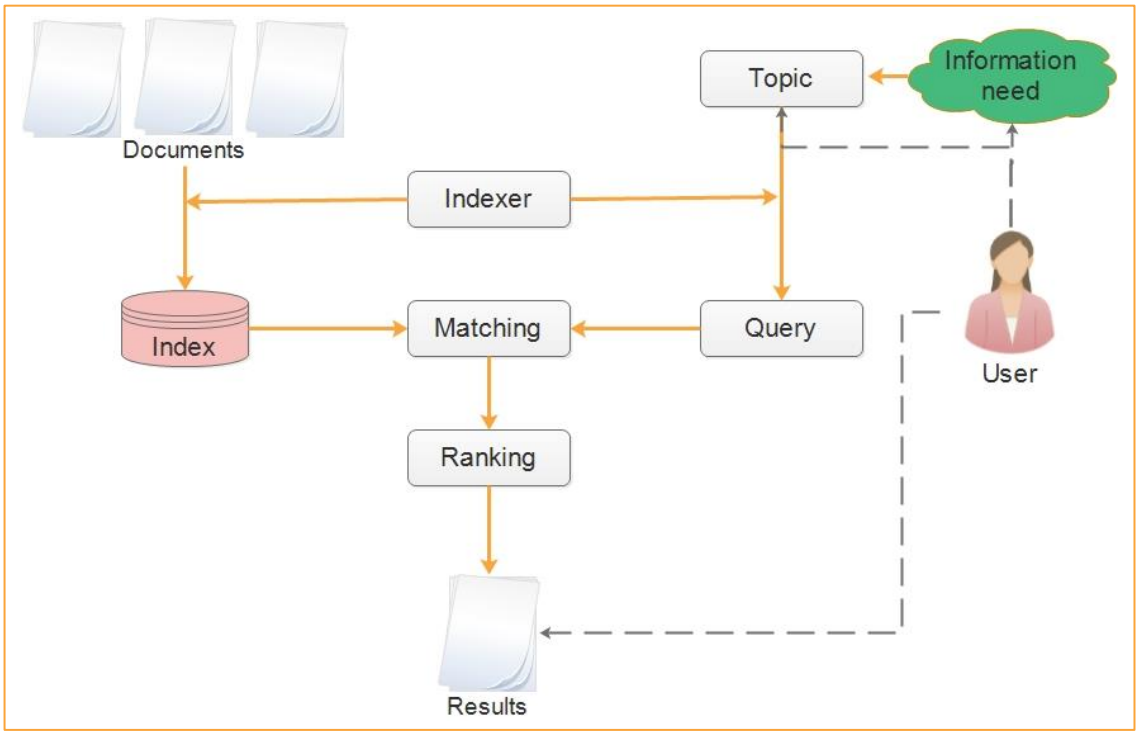


Figure 2-2: An elaborated view of the IRS [15].

There are three primary processes an IR model has to support [9, 25, and 30]: (a) the representation of the information of the documents, (b) the interpretation of the user's information need, and (c) the comparison of these two representations. Representing the documents in a summarized way is usually called the "*indexing process"*. Indexing process implemented off-line, means; client of the information retrieval system is not directly involved in this process. Indexing process result in a representation of the document. Users do not search irrelevant information; they have a need for only relevant information. The process of representing relevant information need to the given user is called as the "*query formulation process*". The resulting representation is called "*query"*. Comparing the two different representations is called as the "*matching process"* and retrieval of relevant documents is the result of this process. *Figure 2.3* hereafter, shows the overall IR process.

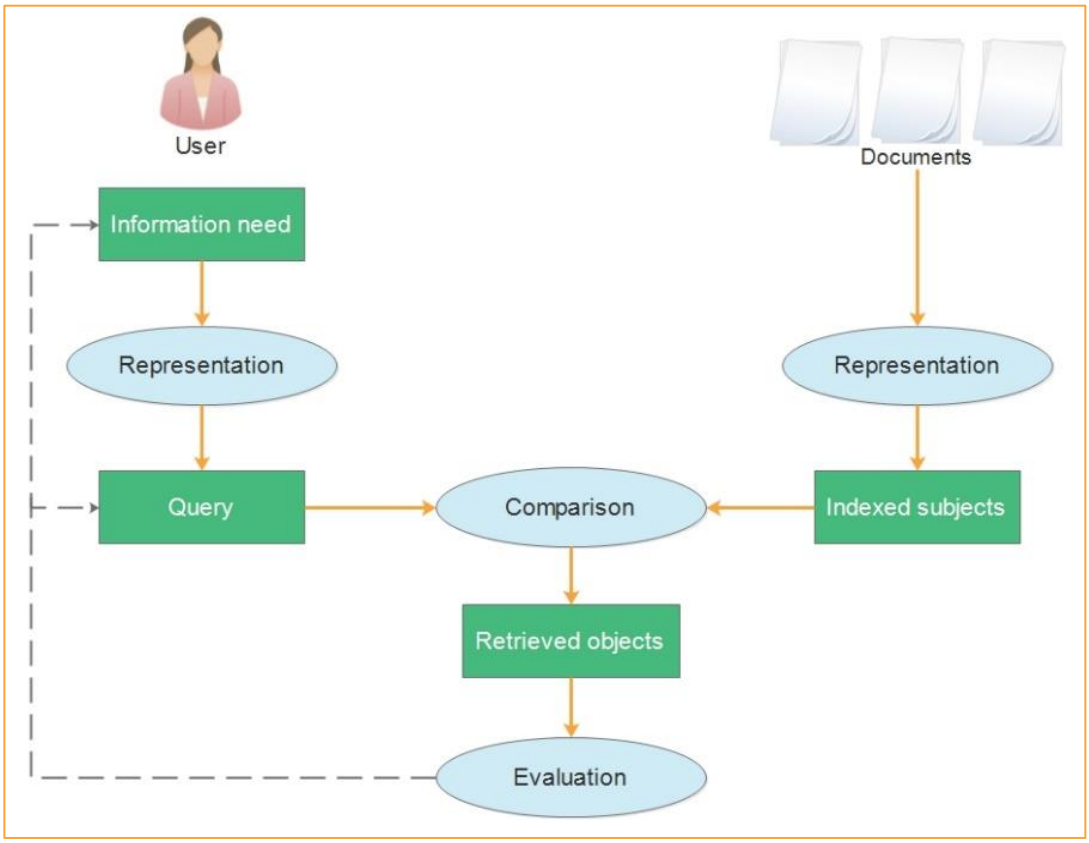


Figure 2-3: Basic Information Retrieval Process.

We devote the remaining of this chapter to talk about these important concepts for an IRS.

2.5.1. Natural language processing

Text processing methods can be leveraged prior to search in order to predict or select multiword terms that better represent an information need. However, much research in IR focuses on understanding the behaviour of IR systems, rather than pre-processing techniques that are independent of IR systems [29].

The noise originates from several sources, these include imperfect text recognition (6% word error rate), spelling variation, non-standardized grammar, in addition to user-side confusion due to her/his limited knowledge of the underlying language or the searched text. Manual correction or normalization are very time-consuming and resource-demanding tasks and are thus; not the ideal solutions [15].

There are two main steps in text-processing: Tokenisation and Normalisation. Two other steps have been proved to increase efficiency; those steps are: Stop-words removing and words stemming or lemmatizing.

2.5.2. Tokenization

Tokenization is the process of chopping on whitespaces and characters streams into tokens and deleting punctuation characters [12].

Example:

Input: "And verily the Hereafter will be better for thee than the present." Chapter (93) surat l-duha (The Forenoon), Verse (4).

Output: |And| |verily| |the| |Hereafter| |will| |be| |better| |for| |thee| |than| |the| |present|.

A *token* is an instance of a character sequence in some particular document. A *type* however, is the class of all tokens containing the same character sequence and a *term* is a (normalized) type that is indexed in the IR system dictionary.

There are some issues to tokenization [8, 12]:

- Capitalization (retrieval vs. Retrieval);
- Apostrophe (e.g. "aren't" to aren't, arent, are|n't, aren|t);
- Hyphenation (e.g. "over-eager" to overeager, over|eager);
- White spaces (e.g. "Tizi Ouzou" to 'Tizi Ouzaou' or 'Tizi|Ouzou');
- Compounds (e.g. "Computerlinguistics");
- Tokenization is language specific (i.e. there is a need for language identification);
- Tokenization should recognize specific strings (e.g. email addresses, URLs, etc.);
- The same tokenization needs to be performed on documents and queries.

2.5.3. Stop-words removing

Stop words are the extremely common and semantically non-selective words. They are excluded from the dictionary entirely [12]. The general strategy is as follows: (a) To sort terms by frequency; and then (b) To add the most frequent terms to the stop list.

Stop words are highly frequent words that appear in many documents, irrespective of the topic of a text, e.g. {with, the, do, if, he}. The length and content of stop-word lists vary, but a small list contains around 37 words. A more comprehensive list contains around 420 words [29]. The stop-list used in this dissertation contains 710 word; including 210 html tags. The advantages of stop-words removing techniques are that it is *effective* because these words will not be considered as relevant to be searched and it is also *efficient* in a way that it reduces the storage size and the time complexity. About the disadvantages of using a stop-words removing phase is that there is a problem with queries where the stop-words do have a higher impact (e.g. “To be or not to be”).

2.5.4. Normalization

Token normalization is the process of canonicalizing tokens so that matches occur despite superficial differences in the character. The most standard way of normalizing is to create equivalence classes, which are named after one member of the set [12].

Unfortunately, tokenization might cause unexpected results, for example: the acronym C.A.T. becomes cat the animal.

Normalization typically deals with accents and diacritics (might be critical in languages other than English), for example, résumé → resume or naïve → naive.

Normalization also deals with capitalization/case folding. A common strategy is to reduce to lower case. This might be critical also: “General Motors” → general motors

2.5.5. Stemming and lemmatization

Stemming and lemmatization can be considered as normalization techniques. The goal of stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form [12, 31].

Indeed, many morphological variations of words (with same or similar meaning) exist.

Examples:

Inflectional: 'bought', 'buying'→ 'buy' or 'am', 'are', 'is' → 'be'

Derivational: 'compute', 'computable' → 'computer' or 'car', 'cars', 'car's', 'cars' → 'car'

- *Stemming*: is a crude heuristic process that chops off the prefixes and suffixes of words in the hope of achieving this goal correctly most of the time. Stemming can be crucial for some languages, e.g., 5-10% improvement for English, up to 50% in Arabic [31].
- *Lemmatization*: is an accurate process that makes use of a dictionary and morphological analysis of words, normally aiming to return the base or dictionary form of a word. Lemmatization collapses the different inflectional forms of a lemma.

These two concepts will be further detailed in the *Chapter 6* of this thesis. But for now, let's just say that stemming increase recall and decrease precision. We aim to propose a semantically enriched version of a well-known stemming algorithm that increase recall without affecting the precision.

Note:

The Natural Language Processing pipeline should be applied to every document in the collection in order to help creating the corpus index (see *Figure 2.4*). Moreover, the same processing steps need to be applied to the documents and the queries indifferently.

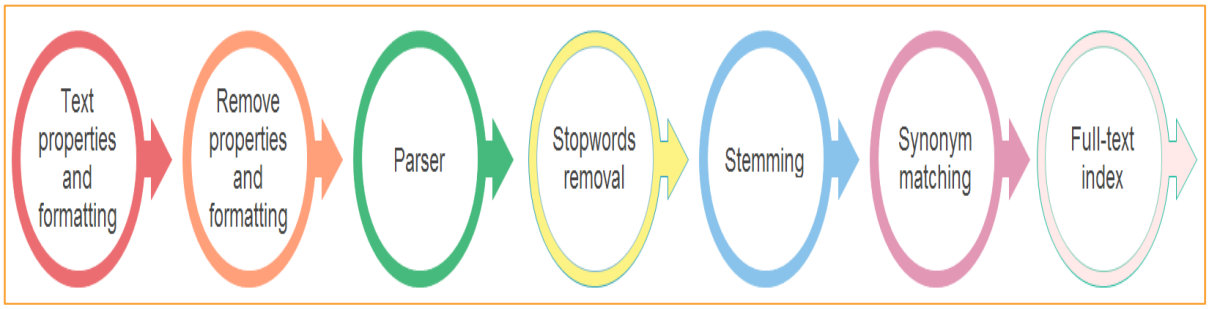


Figure 2-4: Natural Language processing steps for indexing [33].

2.5.6. Indexing process

Search systems rarely search document collections directly. Instead an index is built of the documents in the collection. Undoubtedly, the most difficult, most onerous task in IR is Indexing. To quote Fairthorne[9]: *"Indexing is the basic problem, as well as the costliest bottleneck of information retrieval"*.

Much of the fundamental research in IR indexing was carried out by Gerald Salton, Professor of Computer Science at Cornell, and his graduate students [5].

All practical descriptor languages have common ground in their necessity to have [34]: (a) an alphabetical arrangement and (b) an arrangement showing relationship between terms.

Hereafter, we introduce the definition of some primordial concepts to the comprehension of this sub-section [15, 31, and 35].

- *Document*: A document may constitute a paragraph extracted from a novel, a book chapter, a newspaper article or simply a scientific article's abstract. It can even be as short as a line of text which might represent a poetical verse, an image caption, or the contents of a cell in a table.
- *Corpus*: The corpus is the collection of material indexed by the search engine. The main corpus for search engines like Google[10]

[9] Robert Arthur Fairthorne (1900-1982) is a Mathematician and one of the pioneers of the field of information science.

[10] http://www.google.com

and Bing[11] is the web, although they are beginning to index information stored in mobile apps. Specialized search engines limit themselves to particular collections of content. An example of a specialized search engine is Scopus[12], a classic abstracting and indexing database covering a corpus of peer reviewed scientific and scholarly literature.

- *Index*: The IRS performance worsens when queries get more complicated and the documents' size or number grows. This is where indexing comes in to provide an efficient solution. The corpus should thus be indexed as a first step after which an inverted file of this index is produced. The index is a data structure that represents and stores items drawn from a content collection or corpus. It can be considered as a list of concepts with pointers to documents that discuss them. Most modern search engines use an “inverted index” which lists all the terms occurring in the corpus along with their locations. Designing an efficient, scalable, resilient and robust index is a source of competitive advantage and differentiation for search engines, especially in the case of social web search engines which must index petabytes of data and handle hundreds of thousands of queries per second with millisecond response time.
- *Inverted index:* An inverted file is analogous to a book index. A list of terms (such as words and acronyms) that are sorted alphabetically, each associated with the page numbers containing that term. This concept applies exactly to the inverted file, wherein indexing terms point to documents in which they appear (instead of pages). Documents are thus represented by their indexing terms.

Figure 2.5 represents the indexing architecture of an IRS.

[11] http://www.bing.com

[12] http://www.scopus.com

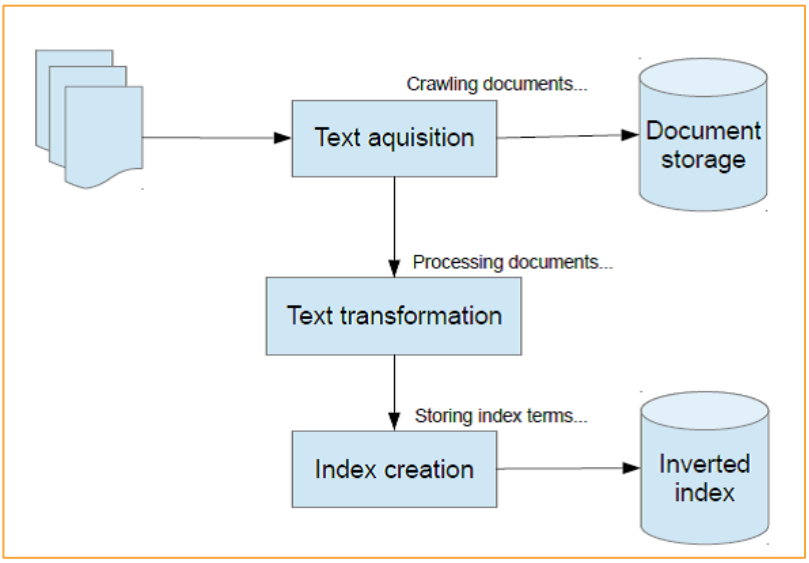


Figure 2-5: The indexing architecture [33].

Indexing helps increasing the efficiency of the retrieval engine and can have different strategies in keeping the relevant information about a given document: Index-term with document ID, Index-term with document ID and frequency, and finally Index-term with document ID and position [8].

2.5.6.1. Documents' representation

Document Representation means deciding what concepts should go in the index [31]: (a) Option 1 (controlled vocabulary): a set of a manually constructed concepts that describe the major topics covered in the collection and (b) Option 2 (free-text indexing): a set of individual terms that occur in the collection.

The second option of indexing is the most common one nowadays. This technique is somehow delicate in a way that it is mandatory to locate the important information in the documents. Thus, there is a need for a deep natural language understanding.

2.5.6.2. Indexing techniques

There are various common IR indexing techniques, including signature files and inverted index [9, 25]:

- *Signature file:* In signature file indexing technique each document return a bit of string, (that is, signature) using hashing method on its text and superimposed coding. The final output of document signatures are stored in a special way, that is sequentially in a separate file and this file is called as signature file. The signature file is much smaller than the original file, and it can provide high search rate.
- *Inverted index:* Each document can be represented by a list of some reference words called keywords which depict the contents of the document for retrieval purpose. Fast retrieval can be obtained if we invert on those keywords. All the reference words are stored alphabetically in a file called index file. For each keyword we keep a list of pointers to the characterized documents in the postings file. This method is mostly used by all the commercial systems.

In our thesis, we present an IRS that allows the creation of signature files using free-text indexing.

2.5.7. Querying process

The traditional (and most common way) to describe the user's search needs is via text. Other query types also exist, such as query by example (image) and query by humming (singing) [15]. These non-textual querying methods are not covered here as they lie beyond the scope of this dissertation which is solely concerned with searching texts.

We note also, that the term document in this dissertation only refers to text documents whenever mentioned from now on.

A simple vision of the querying architecture is depicted in *Figure 2.6*.

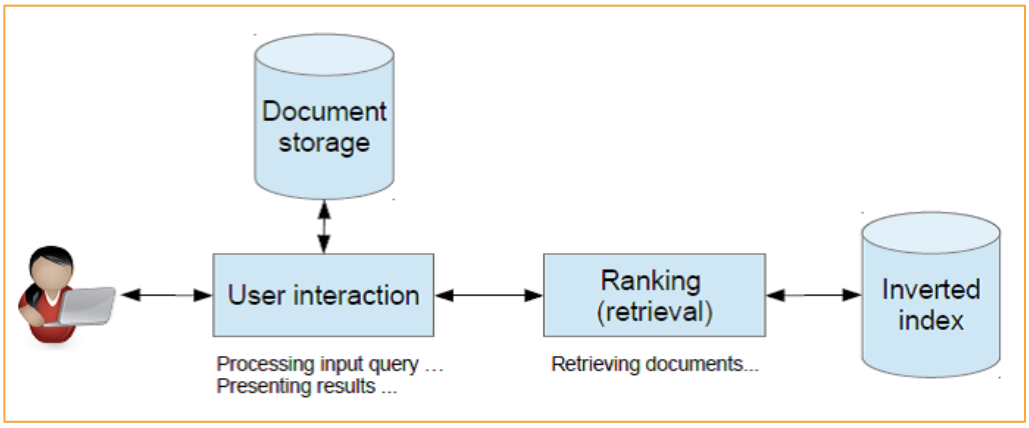


Figure 2-6: Querying Architecture [33].

Hereafter, we introduce the definition of some primordial concepts to the comprehension of this sub-section [5, 15, and 35].

- *Information need:* The information need reflects a problem or question that the user is trying to resolve. The possibilities are virtually limitless, ranging from the prosaic and utilitarian ("should I wear a coat today"?) to more complex and abstract needs ("what position should I take on global warming"?). In the former case, a simple, fact-based answer may be what is desired ("Yes, wear a coat, it is below freezing outside") while in the latter case the need may reflect a desire for a quick education on the subject. With the rise of the web and now smartphone apps, users expect more and more of their information needs to be satisfied through a search.
- *Context:* An information need exists in a specific user context. Examples of context include the user's search and purchase history as well as any personal preferences voluntarily submitted to the search engine. With mobile search an entirely new set of contextual information can be made available to a search engine including location in time and space, locally installed apps, bodily vital sign indicators and more. Capturing a rich user context is an area of significant innovation and competitive differentiation in modern search engines, allowing personalization and tailoring of results that match the characteristics of the user.

- *Query:* The user converts or adapts their information need into a query in order to search. Query input has evolved from intricate, syntactically sensitive Boolean interfaces usable only by trained searchers to looser keyword and phrase-based interfaces accessible and usable by all. Search engines do significant processing of queries, ranging from simple word stemming, to more advanced probabilistic expansion and interpretation techniques. The actual entry of the query is still largely based on textual input of keywords or phrases via physical or virtual keyboard interfaces. Query input is evolving rapidly though, and leading web search engines and emerging intelligent personal assistants are making progress in speech and conversational query interfaces.

  The formulation of a good query can make the difference between a successful and unsuccessful search. Although modern search engines have made great strides in interpreting queries and in handling quasi-normal phrases and questions, the query input is still far from being a natural or intuitive way to ask a question.

2.5.7.1. Query taxonomy

A distinction is usually made in IR literature between natural language queries and questions, even though the former often take the form of a question in the linguistic sense [29].

A question expresses an information need in question-answering (QA) tasks and seeks a specific, small chunk of information such as found in a sentence or phrase. In contrast, a query aims to retrieve relevant documents that contain desired information, or are *'on topic'*. For this reason, natural language queries are typically referred to as verbose queries, or description topics.

There have been many attempts to create a systematic categorization of queries [36]. An influential classification was developed by Broder [37] based on web log analyses. Although limited to the web, it still provides a useful basis for discussion of search queries in general.

Queries fall under three types: Informational, Navigational, and Transactional [5, 15, and 35].

- *Informational*: The intent is to acquire some information assumed to be present on one or more web pages. These may also be called “discovery” queries.
- *Navigational*: The immediate intent is to reach a particular site. A user seeking a specific site, document or person, somewhat like using a directory; may also be called a “known item search”.
- *Transactional*: The intent is to perform some web-mediated activity (purchase an item, download a file, etc.).

2.5.7.2. Query processing

In query processing, similar techniques used in text processing for documents are applied (i.e. tokenization, stop words filtering, stemming). However, some additional steps can be introduced [8]:

- *Spell checking:* consists of correcting the query in case of spelling errors;
- *Query suggestion:* consists of providing alternative words to the original query (based on query logs for example);
- *Query expansion and relevance feedback:* consists of modifying the original query with additional terms (e.g. “the best book for natural language processing” “the best [book | volume] for [natural language processing |NLP]”).

2.5.8. Evaluation process

In the information search process, users seek information with some purpose in mind. They then evaluate the obtained information from many different aspects, and if they acquire that information, that user’s own knowledge structure will change. For this reason, it is difficult to map out in advance the information retrieval process of a user seeking information [22].

Effectiveness and Efficiency are two basic parameters for measuring the performance of system. By effectiveness it means the level up to which the given system attained its objectives. Thus in information retrieval system effectiveness may be measure of how far it can retrieve relevant information while with-holding non-relevant information. Efficiency means how economically the system is achieving its objectives. In an information retrieval system efficiency can be measured by factors such as cost. The cost factors are to be calculated indirectly. They include factors such as response time, time taken by the system to provide an answer. User effort, the amount of time and effort needed by a user to interact with the system and analysed the output retrieved in order to get the correct information [5].

The search engine processes the query and looks for matches against the index. Results are selected, ranked and presented to the user. The most widely used ranking technique in traditional search engines was pioneered by Spark Jones and Salton in the 1970s. The intuition was that terms that occur frequently in a document, but infrequently in the overall collection, are the best discriminators for relevance. The concept is mathematically expressed in the formula: TF (Term Frequency)* IDF (Inverse Document Frequency) [15, 35].

The simplest notion of relevance is that the query string appears verbatim in the document. A slightly less strict notion is that the words in the query appear frequently in the document, in any order - bag of words representation [30].

2.5.8.1. Purpose of information retrieval

The central challenge in textual document retrieval is to rank a collection of documents according to their respective relevance to a query. In textual document retrieval, such a query usually consists of one or several search terms. When querying a document collection with a limited, controlled meta-data vocabulary, it is usually suitable to compute a binary relevance, such that a document is relevant to a query if it contains all

search tokens and irrelevant otherwise. However, when considering full-text retrieval (over natural language documents such as web-sites, news articles, etc.), computing a continuous relevance score is important to return a ranked list of documents to the information seeker.

One of the first and most straight-forward approaches to compute the relevance of a token to a document, is to count the (relative) number of occurrences of that token within the document. This is referred to as the term frequency. To further improve this relevance judgment, it is often useful to normalize it using an inverse-document frequency. As the name implies, this is the relative number of documents within a given corpus, which contain the token at least once. This is intuitive, because even though a token occurs often in a given document, it might still be quite irrelevant if it occurs in almost all of the documents [38].

A collection of *n* documents can be represented in the vector space model by a term-document matrix. An entry in the matrix corresponds to the "weight" w_{ij} of a term T_i in the document D_j; zero means the term has no significance in the document or it simply does not exist in the document [11].

$$\begin{pmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & w_{11} & w_{21} & \dots & w_{t1} \\ D_2 & w_{12} & w_{22} & \dots & w_{t2} \\ : & : & : & \dots & : \\ D_n & w_{1n} & w_{2n} & \dots & w_{tn} \end{pmatrix}$$

2.5.8.2. Term frequency

More frequent terms in a document are more important, i.e. more indicative of the topic. Let *Fij* be the frequency of term *i* in document *j. This frequency is called the Term Frequency (TF)*.

$$TF_{ij} = \frac{F_{ij}}{\max_i \{F_{ij}\}} \quad (1)$$

2.5.8.3. Inverse document frequency

Terms that appear in many different documents are less indicative of the overall topic. Let *DFi* be the document frequency of the term T_i (i.e. the number of documents containing term T_i. *IDFi* will be the inverse document frequency of the term).

$$IDF_i = Log_2(N = D_{fi}) \quad (2)$$

Where *N* is the total number of documents.

Equation (3) is used to compute a term importance by the tf-idf weighting technique:

$$W_{ij} = TF_{ij} \times IDF_i = TF_{ij} \log_2(N = DF_i) \quad (3)$$

A term that occurs frequently in the document but rarely in the rest of the collection is given a high weight.

2.5.8.4. Evaluation in information retrieval

The empirical nature of IR requires realistic datasets, these are, however, not widely available. For this reason, standardized test-collections were built by initiatives like, for instance, TREC[13] (Text REtrieval Conference) and CLEF[14] (Conference and Labs of the Evaluation Forum - formerly known as Cross-Language Evaluation Forum). These initiatives offer different tasks and test-collections to support research in IR. The alternative (yet expensive and resource-demanding) approach is to build test-collections that comply with the standards yet meet the specific research needs on hand [15].

2.5.8.5. Evaluation Criteria (Effectiveness VS Efficiency)

The problem of searching document collections to find relevant documents has been addressed for more than forty years. However, until the advent of the Text REtrieval Conference (TREC) in 1990 (which is hosted by the National Institute of Standards and Technology), there was no standard test bed to judge

[13] http://trec.nist.gov

[14] http://www.clef-initiative.eu

information retrieval algorithms. Without the existence of a standard test data collection and a standard set of queries, there was no effective mechanism by which to objectively compare the algorithms. Many of these algorithms were run against only a few megabytes of text. It was hoped that the performance of these would scale to larger document collections. A seminal paper showed that some approaches that perform well on small document collections did not perform as well on large collections [6].

At first, Cleverdon et al. (1966) set up an experimental environment for IR experiments, in which documents were indexed by content features and retrieved via queries, and then evaluated in a batch mode. This experimental setup is better known as Cranfield IR evaluation, sometimes also called the laboratory IR, which can also be described as system-centred/oriented IR. However, while system-oriented IR focuses on performance and effectiveness, designing a good IR system depends not only on system-oriented performance issues but also on understanding the users who interact with the system [6].

In 1985, Blair and Maron [39] authored a seminal paper that demonstrated what was suspected earlier: performance measurements obtained using small datasets were not indicative for larger document collections.

In the early 1990's, the United States National Institute of Standards and Technology (NIST), using the text collection created by the United States Defense Advanced Research Project Agency (DARPA), initiated a conference to support the collaboration and technology transfer between academia, industry, and government in the area of text retrieval. The conference, named the Text REtrieval Conference (TREC) aimed to improve evaluation methods and measures in the information retrieval domain by increasing the research in information retrieval using relatively large test collections on a variety of datasets.

TREC is an annual event held in November at NIST [8]. Over the years, the number of participants has steadily increased and the types of tracks have greatly varied. In its most recent incarnation in 2017, TREC consists of eight tracks, namely Common Core, Complex Answer Retrieval, Dynamic Domain, Live Question Answering, Open Search, Precision Medicine, Real-Time

Summarization, and Tasks Track. The specifics of each track are not relevant since the tracks are continuously modified. Suffice to say that the type of data, queries, evaluation metrics, and interaction paradigms (with or without a user in the loop) vary greatly. The common theme of all tracks is to establish an evaluation corpus to be used in evaluating search systems.

Today, the types of data vary greatly, depending on the focus of the particular track. Likewise, the volumes of data vary also. Thus, within roughly a decade, the collection size has grown significantly. This growth of data might necessitate new evaluation metrics and approaches in the future.

Given this increased participation, more and more techniques are being developed and evaluated. The transfer of general ideas and crude experiments from TREC participants to commercial practice each demonstrates the success of TREC.

2.5.9. Ranking in Information Retrieval

Partly because of the central role of the ranking process in search engines, great attention has been paid to the research and development of ranking technologies. Note that ranking is also the central problem in many other information retrieval applications, such as collaborative filtering, question answering, multimedia retrieval text summarization, and online advertising. Sometimes we need to rank documents purely according to their relevance with regards to the query. In some other cases, we need to consider the relationships of similarity and diversity between documents in the ranking process. This is also referred to as relational ranking.

To tackle the problem of document retrieval, many heuristic ranking models have been proposed and used in the literature of information retrieval. Recently, given the amount of potential training data available, it has become possible to leverage machine learning technologies to build effective ranking models. Specifically, we call those methods that learn how to combine predefined features for ranking by means of discriminative learning “learning-to-rank” methods.

The notion of learning to rank techniques is out of the scope of our study, but valuable information may be found here [40].

From an IR point of view, documents that are highly similar to a given query are considered as relevant while from a user perspective this concept still applies but however relevancy gets purely subjective and situational and may change over time and knowledge acquisition. Therefore, in addition to its similarity and the associated scores, other factors also contribute to shaping the user's evaluation of relevance of a given document such as broadness, timeliness, and information novelty. We will talk about those factors in details in *Chapter 5.*

2.5.9.1. Relevance ranking models

The goal of a relevance ranking model is to produce a ranked list of documents according to the relevance between these documents and the query. Although not necessary, for ease of implementation, the relevance ranking model usually takes each individual document as an input, and computes a score measuring the matching between the document and the query. Then all the documents are sorted in descending order of their scores.

The early relevance ranking models retrieve documents based on the occurrences of the query terms in the documents. Examples include the *Boolean model*. Basically these models can predict whether a document is relevant to the query or not, but cannot predict the degree of relevance. To further model the relevance degree, the *Vector Space model* (VSM) was proposed [40].

Relevance is a subjective judgment and may include [30]:

- Being on the proper subject;
- Being timely (recent information);
- Being authoritative (from a trusted source);
- Satisfying the goals of the user and his/her intended use of the information (information need).

We believe, the last point is the most important relevance criterion an IRS should fulfil.

Several metrics are used in information retrieval to judge the performance of ranking algorithms. Here, we introduce the terminology and give a short summary of those metrics, which are also partly used for evaluation in the scope of this thesis (see *Chapter 6*).

There are two key statistics in the measurement of IR system's effectiveness which are calculated based on the results (documents) returned by the IR system for a certain query or a set of queries [15]. To calculate the precision and recall, all the relevant documents must be determined for each query beforehand (see *Figure 2.7*). When using empirical test-collections (e.g., TREC and CLEF) a list stating the relevant documents for each query (relevance assessments/judgment) is usually made available together with the document set (corpus) and the queries [15].

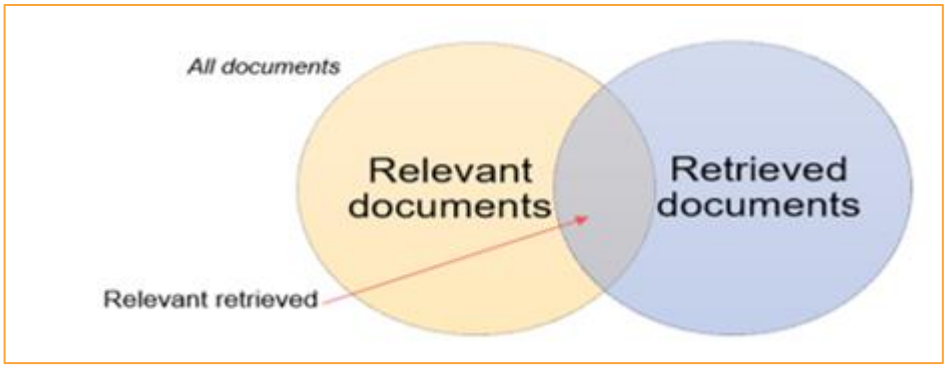


Figure 2-7: Nature of documents in a corpus.

Precision:

In the context of information retrieval, the higher the precision of a retrieval algorithm, the higher the ratio of relevant versus irrelevant documents in a result set. So precision is the fraction of retrieved documents that are relevant to the query [6, 9, 25, 30, and 32]. More formally, we notate precision as:

$$\Pr ecision = \frac{\left|\{relevant\_docs\} \cap \{retrieved\_docs\}\right|}{\left|\{retrieved\_docs\}\right|} \quad (4)$$

In the context of classification, precision is also known as the positive predictive value (PPV) and denoted as:

$$\Pr ecision = \frac{\left|\{true\_positives\}\right|}{\left|\{true\_positives\}\right| + \left|\{false\_positive\}\right|} \quad (5)$$

Recall:

Recall is the fraction of relevant documents that are retrieved. So in the context of information retrieval, the higher the recall, the higher the ratio of relevant documents in the result set versus relevant documents in the corpus [6, 9, 25, 20, and 32]. Recall is formalized as follows:

$$\mathrm{Re}call = \frac{\left|\{relevant\_docs\} \cap \{retrieved\_docs\}\right|}{\left|\{relevant\_docs\}\right|} \quad (6)$$

Recall is also used in classification where it is known as sensitivity:

$$\mathrm{Re}call = \frac{\left|\{true\_positives\}\right|}{\left|\{true\_positives\}\right| + \left|\{false\_negatives\}\right|} \quad (7)$$

2.5.9.2. Similarity measures

In short, a similarity measure is a function that determines the degree of resemblance between the query and each document. Thus, it will be possible to rank the retrieved documents in the order of presumed relevance using a threshold, which is a real number indicating how filters should be applied to ignore irrelevant documents [30, 32].

Many evaluation measures have been proposed and are currently used in IR. At the most basic level, the effectiveness of an IR system, or model, is measured in terms of precision and recall. However, the similarity measure depends on the nature of the documents and their presentation (i.e. model) as well as on the results presentation.

Both precision and recall as well as F-mesure and accuracy are defined on unordered sets of retrieved documents, and interpreted at different cut-off and balance points using interpolation, or via graphical

comparisons and definitions. These metrics are reported in this dissertation and computed using the Terrier platform on the WT2G TREC dataset web collection.

Text-based information retrieval faces some problems related to language processing, such Homonymy, Polysemy, and Synonymy which reduce the precision (i.e. relevant documents will be judged as irrelevant). Moreover, Synonymy and hyponymy reduce the recall (i.e. relevant document that do not contain the keyword, but its synonym instead will be ignored by the IRS).

In the case of a ranked set of documents however, some of the performance measures include [29, 33][15]: Precision-recall curve, Mean average precision, Precision at n, and Mean reciprocal rank.

- Average precision metric is calculated by averaging the precision after each relevant document is retrieved. It summarizes performance over all documents in a collection, and rewards systems that rank relevant documents earlier (high) in the retrieved set. It is thus well suited to the evaluation of open domain search, where users care almost exclusively about the top ranked results.
  However, for queries with many relevant documents, a long tail of lower ranked documents can have a substantial impact.
  The average precision is computed using the equation:

$$Average-precision = \frac{\sum_{k=1}^{k} P@k}{R} \quad (8)$$

  Where K is the rank of relevant documents in the retrieved list.
- The Mean average precision aims to reporting the mean of the average precisions over all queries in the query set. The Mean Average Precision (MAP) is calculated over all queries in a set, and demonstrates exceptionally good stability and discrimination between systems. However, it weights queries equally, so it does not reveal variation in performance for queries that have a great

[15] The descriptions introduced here are based on NIST standards

many, or only a few, relevant documents. MAP can vary widely across queries, and must be calculated on a fairly large and diverse test set in order to be meaningful.

- Precision at *K* evaluates results for the first *K* items of the retrieved documents calculating their precision at *K* and considering the top *K* documents only; ignoring the rest. This metric is suitable for evaluation of IR tasks where only a limited number of results are pertinent to a user. For example, in open domain search, ten results are presented per web page, so a user might be interested in precision at *K* = 10. Precision at the threshold of *K* documents is summed over all queries in a set, and divided by the number of queries. Precision at k is not a good measure for performance across a set of queries because queries with more relevant documents tend to have higher precision at *K*.
- Reciprocal Rank evaluates only one correct answer by inversing the score of the rank at which the first correct answer is returned.

$$\mathrm{Re}\,ciprocal - rank = {}^{1}\!/_{R} \quad (9)$$

Where *R* is the position of the first correct item in the ranked list.

- Mean Reciprocal Rank aims to reporting the mean of the reciprocal rank over all queries in the query set.

2.5.9.3. Discussion

Experimental evaluation is essential to the assessment of the effectiveness of IR systems. The traditional approach to measuring the effectiveness of diverse IR systems goes back to the Cranfield tests in the 1960s. However, neither user characteristics nor time are considered in the traditional evaluation process. In the Cranfield-type tests, still popular today, users are taken into account only marginally and their interests are represented in relevance assessments, evaluation metrics and topics to some extent. However, interaction with an IR system can be dissected more precisely and users' interaction during a search session can be divided further into subtasks. This in turn affects the evaluation process of

IR systems. Moreover, users' feedback during a search session, which may be of high or poor quality, can be exploited to improve the search results. This again influences the effectiveness of search systems. In the present thesis, we examine the effects of users' characteristics and the relevance with respect to the search effectiveness [11].

In information retrieval, relevance is used to determine whether search results satisfy the user's information needs. In system-centred information retrieval, relevance is regarded as an objective indicator determinable by a third party, and is used in judging search results [22]. This model assumes that a third party can objectively determine whether the search results are relevant or not. In contrast, in the user-centred approach to information retrieval, relevance can only be determined by users themselves. In system-centred information retrieval, whether the subjects of search results are relevant or not is determined objectively, and so relevance can be measured; whereas in user-centred information retrieval, whether search results are relevant to the context is determined subjectively, and so measuring relevance in this case is considered difficult. This problem will be further detailed in *Chapter 6*.

2.6. Conclusion

Foundational work on term selection for IR is inextricably linked to the development of IR models that go beyond a word independence assumption [29]. There have been some efforts in developing systems that interpret natural language queries and automatically perform the appropriate Information Retrieval operations. Text Information Retrieval tools appears nowadays in the form of intelligent personal assistants. Under the agent paradigm, a personal miner would learn a user's profile, conduct text information Retrieval process repeatedly, and forward information not including requiring an explicit request from the client [1].

Indeed, many problems are associated with the current IRS and such can be seen from the inability of the system to process request timely and to present inadequate results among others [2]. In view of these inadequacies, it is imperative to develop an IR system that will take

into account the context of the IR task. Effectively, identifying the meaning of words in context is not difficult for human interpreters, but remains a challenge for even the most advanced machines since a word has multiple senses indicating different meanings in different contexts [8].

Moreover, even though the actual IR evaluation efforts take the user into consideration by including predefined relevance judgments and diverse evaluation methods, they are limited in nature. As humans are diverse, so are IR system users. Accordingly, the interaction of the user with IR systems exhibits miscellaneous behaviour, which is lacking in the design and implementation of the pertinent systems. Classical studies assume an average user, who interacts with a retrieval system in a predictable and regular way. However, users are diverse and not always predictable [11].

Context is one of our major concerns and we will discuss it in detail, in *chapter 5*.

CHAPTER 3 WEB-BASED INFORMATION RETRIEVAL

3.1. Introduction

The Web can be considered as a large-scale document collection, for which classical text retrieval techniques can be applied. However, its unique features and structure offer new sources of evidence that can be used to enhance the effectiveness of IR systems. Generally, Web IR examines the combination of evidence from both the textual content of documents and the structure of the Web, as well as the search behaviour of users, and issues related to the evaluation of retrieval effectiveness. This chapter presents an overview of Web IR. It discusses the differences between classical IR and Web IR, a range of Web specific sources of evidence, and the combination of evidence in the context of Web IR. This chapter also provides a brief overview of work on the evaluation of Web IR systems, as well as on query classification and performance prediction.

3.2. Background of web Information Retrieval

The Web is unprecedented in many ways: unprecedented in scale, unprecedented in the almost-complete lack of coordination in its creation, and unprecedented in the diversity of backgrounds and motives of its participants [5].

Each of these contributes to making web search different – and generally far harder – than searching “traditional” documents.

The invention of hypertext, envisioned by Vannevar Bush in the 1940’s and first realized in working systems in the 1970’s, significantly precedes the formation of the WorldWide Web (which we will simply refer to as the web), in the 1990’s. Web usage has shown tremendous growth to the point where it now claims a good fraction of humanity as participants, by relying on a simple, open client-server design [5]: (1) the server communicates with the client via a protocol (the *http* or hypertext transfer protocol) HTTP that is lightweight and simple, asynchronously carrying a variety of payloads (text, images and – over time – richer media such as audio and video files) encoded in language called *HTML* (for

hypertext markup language); (2) the client – generally a *browser*, an application within a graphical user environment can ignore what it does not understand. Each of these seemingly innocuous features has contributed enormously to the growth of the Web, so it is worthwhile to examine them further.

The designers of the first browsers made it easy to view the HTML markup tags on the content of a URL. This simple convenience allowed new users to create their own HTML content without extensive training or experience; rather, they learned from example content that they liked. As they did so, a second feature of browsers supported the rapid proliferation of web content creation and usage: browsers ignored what they did not understand. This did not, as one might fear, lead to the creation of numerous incompatible dialects of HTML. What it did promote was amateur content creators who could freely experiment with and learn from their newly created web pages without fear that a simple syntax error would "bring the system down." Publishing on the Web became a mass activity that was not limited to a few trained programmers, but rather open to tens and eventually hundreds of millions of individuals. For most users and for most information needs, the Web quickly became the best way to supply and consume information on everything from rare ailments to subway schedules.

The mass publishing of information on the Web is essentially useless unless this wealth of information can be discovered and consumed by other users. Early attempts at making web information "discoverable" fell into two broad categories [5]: (1) full-text index search engines such as Altavista, Excite and Infoseek and (2) taxonomies populated with web pages in categories, such as Yahoo[16]! The former presented the user with a keyword search interface supported by inverted indexes and ranking mechanisms building on those introduced in earlier chapters. The latter allowed the user to browse through a hierarchical tree of category labels. While this is at first blush a convenient and intuitive metaphor for finding

[16] https://fr.yahoo.com

web pages, it has a number of drawbacks: first, accurately classifying web pages into taxonomy tree nodes is for the most part a manual editorial process, which is difficult to scale with the size of the Web. Arguably, we only need to have “high-quality” web pages in the taxonomy, with only the best web pages for each category.

However, just discovering these and classifying them accurately and consistently into the taxonomy entails significant human effort. Furthermore, in order for a user to effectively discover web pages classified into the nodes of the taxonomy tree, the user’s idea of what sub-tree(s) to seek for a particular topic should match that of the editors performing the classification. This quickly becomes challenging as the size of the taxonomy grows; the Yahoo! taxonomy tree surpassed 1000 distinct nodes fairly early on. Given these challenges, the popularity of taxonomies declined over time.

The first generation of web search engines transported classical search techniques such as those in the preceding chapters to the web domain, focusing on the challenge of scale. The earliest web search engines had to contend with indexes containing tens of millions of documents, which was a few orders of magnitude larger than any prior information retrieval system in the public domain. Indexing, query serving and ranking at this scale required the harnessing together of tens of machines to create highly available systems, again at scales not witnessed hitherto in a consumer-facing search application.

The first generation of web search engines was largely successful at solving these challenges while continually indexing a significant fraction of the Web, all the while serving queries with sub-second response times.

However, the quality and relevance of web search results left much to be desired owing to the idiosyncrasies of content creation on the Web. This necessitated the invention of new ranking and spam-fighting techniques in order to ensure the quality of the search results.

While classical information retrieval techniques (such as those covered earlier in this thesis) continue to be necessary for web search, they are not by any means sufficient.

3.3. Classical IR VS Web IR

Keyword-based IRSs often represent documents and queries as a bag-of-weighted-words or multi-words (phrase). This representation is obtained through document lexical analysis within collections that summarize document contents by a set of lexical units [41]. A keyword-based IRS relevance process may rely on an exact match, an approximate match, or a string distance between words within documents and query indexing. Hence, when a query is submitted, these systems will retrieve documents indexed by exact query keywords or some of their lexical variations (e.g. tumorous instead of tumour). Unfortunately, they miss documents having query keyword synonyms in their indexing (e.g. carcinoma instead of tumour) [42]. This so-called the synonymy problem is the most common shortcoming, but keyword-based IRSs also fail to consider various kinds of semantic relationship between words (hyponyms, hypernyms). They are hampered by polysemous problems due to language ambiguity [41-42]. Indeed, a word may have several meanings depending on the usage context (e.g. cancer as astrological sign or as illness).

A syntactic search engine will retrieve a document when its indexing contains a query keyword, even if the meaning of the word within the document differs from what the user had in mind. All of these issues account for the lack of precision of keyword-based information retrieval systems, which is a well-known problem [41].

Two solutions have been proposed to solve the above syntactic search limitations. Both of them involve improving indexing by introducing some semantics [41-42]:

- Structuring lexical units (e.g. noun phrases) extracted from documents using some kinds of relationship (synonymy,

subsomption, etc.). This is possible using natural language processing and machine learning techniques. This strategy may be seen as a first step towards interfacing ontologies and lexical resources since structuring of the latter involves ontological principles. This approach is still, nevertheless, syntactic since the semantics remain implicit.

- Use of conceptual resources to represent document content based on their meaning rather than their words. These resources may be arranged from less formal ones (thesaurus with strong lexical compounds: WordNet or UMLS) to more formal ones (e.g. Gene Ontology). They can also be general or domain specific.

Extraction techniques are needed to make use of such term meaning or concept for indexing purposes. These techniques may be manual or automatic, but this topic is beyond the scope of this chapter.

To sum up, two strategies lead to different indexing units characterized by their granularity: from a lower level (lexical units such as words, noun phrases) to a higher level (conceptual units). The next section reviews and discusses the foundations of conceptual based IRSs.

Moreover, we can say that Traditional IR differs from web-based IR because they have different requirements [42]. These requirements are discussed with respect to three aspects: the hypertext document model, the size and structure of the Web, the quality of information on the Web and the background of Web users [22, 43-44]:

3.3.1. Hypertext document model

The Web is based on a hypertext document model, where the documents are connected with directed hyperlinks. This results in a virtual network of documents. Hypertext was envisioned by Bush as a more natural way to organize, store and search for information, similar to the associative way in which the human mind works [45]. A reader approaches a text by reading and understanding small sections of it, while discovering the connections between the exposed concepts in the text.

The hypertext aids this process by making the connections between parts of the text explicit. In addition, it facilitates the reading of texts in non-linear ways, similarly to structures, Links are divided in two broad classes: internal substance links, and external commentary links. These two classes are further divided in subclasses, leading to an extensive taxonomy of link types. Similarly, there might be two main types of links, namely the organizational and the content-based links [41]. The former type of links was used to organize and help navigation among hypertext documents, while the latter type was used for pointing to documents on similar topics.

3.3.2. Structure of the Web

The Web is a vast repository of information, the size of which is increasing continuously. In November 1997, the size of the static Web was estimated to be approximately 200 million documents. Whereas, the indexable part of the Web was about 800 million documents in February 1999. Today, the indexable Web reaches more than 4.59 billion documents[17]. All these estimates refer to the publicly available part of the Web, which is indexed by search engines. However, it is estimated that even more information is stored in databases or access-restricted Web sites, composing the hidden Web, which cannot be easily indexed by search engines [42].

We can view the static Web consisting of static HTML pages together with the hyperlinks between them as a directed graph in which each web page is a node and each hyperlink a directed edge [5].

Figure 3.1 shows two nodes A and B from the web graph, each corresponding to a web page, with a hyperlink from A to B. We refer to the set of all such nodes and directed edges as the web graph.

[17] http://www.worldwidewebsize.com

Figure 3-1: Two nodes of the web graph joined by a link [5].

As one might suspect, this directed graph is not *strongly connected*: there are pairs of pages such that one cannot proceed from one page of the pair to the other by following LINKS and hyperlinks. We refer to the hyperlinks into a page as *in-links* and those out of a page as *out-links*. These notions are represented in Figure 3.2.

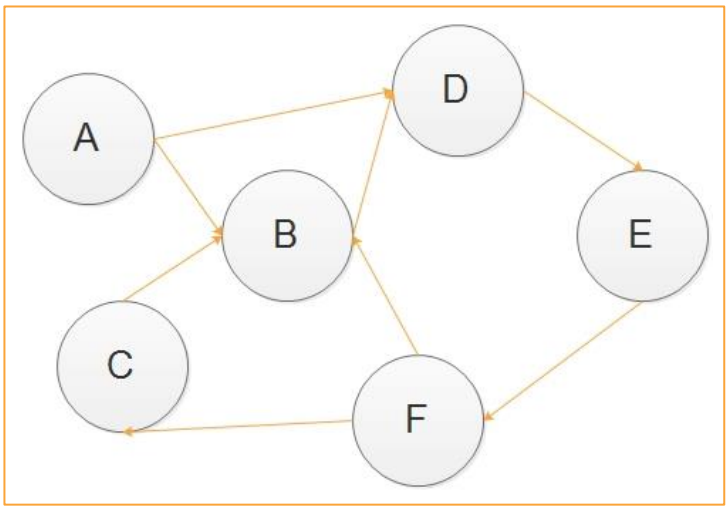


Figure 3-2: A sample small web graph [5].

Search engines have to collect, or crawl the documents from the Web by following hyperlinks, differently from classical IR systems, where the documents are often readily provided.

3.3.3. Quality of information on the Web

Classical IR systems have been often used in controlled environments, where documents contain reliable information that rarely changes. However, the Web is a quite different environment, where no assumption can be made about the quality of Web documents. The information available on the Web is very different from the information contained in either libraries or classical IR collections. A large amount of information on the Web is duplicated, and content is often mirrored across

many different sites [42]. This redundancy ensures that the information is always available, even when some of the mirrors are out of service. However, search engines and IR systems need to take into account the duplication of Web documents, in order to reduce the required resources for crawling Web documents and to avoid returning duplicate Web documents in the results presented to users.

In addition to duplication, the contents of Web pages are not guaranteed to be accurate. Indeed, Web pages may contain false or inaccurate information, due to unintentional errors by their authors, or due to intentional efforts to mislead users in visiting a particular website. Both the issues of duplication and quality of information are more significant in the case of the Web than in the case of classical systems.

3.3.4. Background of Web users

The Web is an open system accessible to anyone. Therefore, no assumption can be made about the users' expertise, experience or computer literacy. There are differences in the search behaviour of novice and experienced searchers in a classical IR setting. Studies of query logs from Web search engines showed that the majority of the users provide short queries, browse only the top ranked documents and do not reformulate the original query [42]. Moreover, users perform search tasks of varying types as seen in the previous chapters.

Empirical studies on Web user behaviour indicate that Web users are impatient and have a tendency to abort their requests within the first 20 seconds.

When a user feels a need for information, he searches online. That if his information needs are well-defined, then he will do a keyword search; but if his information needs are vague, then he will perform a category search. Browsing the list of search results, he clicks on a result relevant to his information needs. If he does not find any relevant results, he rethinks his information needs and changes the keywords, category or search

engine. Browsing the displayed web page, he acquires his desired information.

If his desired information is not available, he returns to the list of search results, and clicks on a separate result that matches his information needs. If he draws a blank again, he rethinks his actual information needs. Then, the acquired information leads to a new query, that is, it generates a new information need, and so the next information search begins (this phenomenon is known as the berry picking model).

In table 3.1, a summary of main differentiation between web IR and traditional IR approaches is presented.

Table 3-1: Web IR VS Traditional IR [42].

	Web IR	Traditional IR
Documents		
Languages	Documents in many different languages. Usually search engines use full text indexing: no additional subject analysis.	Databases usually cover only one language or indexing of documents written in different languages With the same vocabulary.
File Types	Several file types; some are hard to index because of a lack of textual information.	Usually all indexed documents have the same format (e. g. PDF) or only bibliographic information is provided.
Document length	Wide range from very short to very long. Longer documents are often divided into parts.	Document length varies, but not to such a high degree as With the WWW documents. Each indexed text is represented With one documentary unit.
Document structure	HTML documents are semi-structures.	Structured documents allow complex field searching.
Spam	Search engines have to decide which documents are suitable for indexing.	Suitable document types are defined in the process of database design.
Hyperlinks	Documents are connected heavily. Hyperlink structure can be used to determine quality.	Documents are usually not connected. Sometimes citation data is used to determine quality.
WWW Characteristics		
Amount of data, size of databases	The actual size of the WWW is unknown. Complete indexing of the Whole WWW is impossible.	Exact amount of data can be determined When using formal criteria.
Coverage	Unknown, only estimates are possible.	Complete coverage according to the defined sources.
Duplicates	Many documents exist in many copies or versions.	Duplicates are singled out While documents are put into the

		database. No versioning problems because there is usually a final version for each document.
User Behaviour		
User interests	Very heterogeneous interest.	Clearly defined user group with known information seeking behaviour.
Type of queries	Users have little knowledge how to search; very short queries (2-3 words).	Users know the retrieval language; longer, exact queries.
IR System		
User interface	Easy to use interfaces; suitable for laypersons.	Normally complex interfaces: practice needed to conduct searches.
Ranking	Due to the large amount of hits, relevance ranking is the norm.	Relevance ranking is often not needed because the users know how to constrain the amount of hits.
Search functions	Limited possibilities.	Complex query languages allow narrowing searches.

3.4. Web Search Engines

The growth of Internet has made search engines one of the most frequently used web applications over the past decades [46].

"*Search engines are programs that search documents for specified keywords and returns a list of the documents where the keywords were found. A search engine is really a general class of programs, however, the term is often used to specifically describe systems like Google, Bing and Yahoo! Search that enables users to search for documents on the World Wide Web."* [46].

Search engines are extensively important to help users to find relevant retrieval of information on the Web. In order to give the best according to the needs of users, a search engine must find and filter the most relevant information matching a user's query, and then present that information in a manner that makes the information most readily presentable to the user. Moreover, the task of information retrieval and presentation must be done in a scalable fashion to serve the hundreds of millions of user queries that are issued everyday [47].

Traditional search engines are typically keyword-based and have been quite successful by providing users simple search interface and

useful search results. However, since they cannot understand the semantics of a query or relations among queried concepts entered by a user, they often offer low recall and precision, with much of the related information being absent from the search results or too many irrelevant and ambiguous results being presented [48]. For example, the word “Chorba” (in Arabic, شربة meaning soup) may refer variant kinds of soups in Algerian country. When a user types the query “Chorba” into a traditional search engine, expecting some search results related to the corresponding soup in his region, the highly ranked search results are unfortunately all about Algiers’ one. In order to return desirable results (e.g., Oran’s) for this user, it is required that the search engine can clearly understand the ambiguous meaning of the word “Chorba.” And the context of the query (location of the user…etc). We will get to this point in the next chapter. In another example, suppose a computer science student wants to search for information related to a concept called “Folksonomy-based search” Unfortunately, the student cannot remember this terminology, though he knows that such a model relates to Information Retrieval and Social Bookmarking. Using semantic search, when the student types a pair of concepts “Information Retrieval” and “Social Bookmarking” within the domain of computer science courses, the reasoner should efficiently infer the concept of “Folksonomy-based search” and present it to the student for further querying. Different from traditional search engines, semantic search engines utilize semantic information of certain domain knowledge specified using ontology. With the support of domain knowledge, user queries are precisely and unambiguously analysed in order to provide better precision and recall rates for search results [48].

The current web search market is dominated by several search engines like Google, Bing, Baidu[18] etc. Though their architecture and used techniques may differ greatly in details, the basic workflow of search engines remains unchanged: crawling, indexing and searching. Indeed, a typical search engine architecture is shown in figure 3.3. As can be seen

[18] http://www.baidu.com

from the figure, there are in general six major components in a search engine [40]: crawler, parser, the crawler collects webpages and other documents from the Web, according to some prioritization strategies.

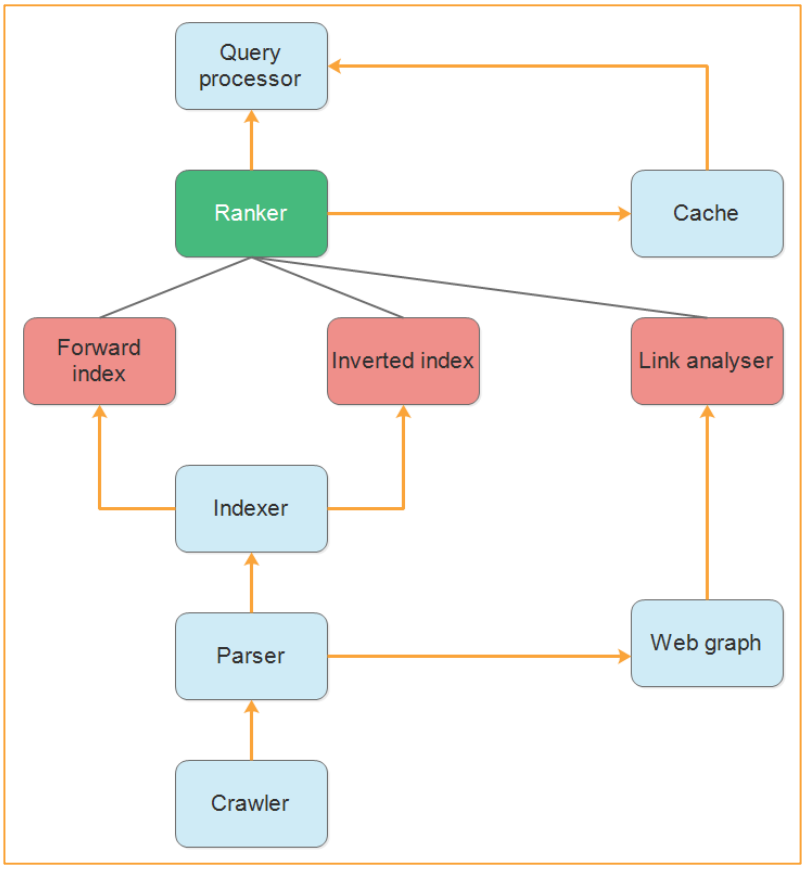


Figure 3-3: Typical search engine architecture [40].

The parser analyses these documents and generates index terms and a hyperlink graph for them. The indexer takes the output of the parser and creates the indexes or data structures that enable fast search of the documents. The link analyser takes the Web graph as input, and determines the importance of each page. This importance can be used to prioritize the re-crawling of a page and to serve as a feature for ranking. The query processor provides the interface between users and search engines. The input queries are processed (e.g., removing stop words, stemming, etc.) and transformed to index terms that are understandable by search engines. The ranker, which is a central component, is responsible for the matching between processed queries and indexed documents. The ranker can directly take the queries and documents as inputs and compute a matching score using some heuristic formulas, and

can also extract some features for each query-document pair and combine these features to produce the matching score.

Hereafter, we briefly talk about the workflow of a web search engine based on the anatomy of Google [49]. But before, we briefly introduce the most predominant web search engines and give some statistics.

3.4.1. Commercial web search examples

The Google is considered to be a world's largest and most comprehensive collection of web documents. It immediately finds the information that we need by using the following services [47]: Google Web Search is the search service offers more than 2 billion documents - 25 percent of which are other than English language web pages. Google Web Search offers users to search the numerous non-HTML files such as PDF, Microsoft Office, and Corel documents. Google's uses the powerful and scalable technology for searches, which is the comprehensive set of information and it delivers a list of relevant results with in less than half-a-second. Google Groups is a 20-year archive of Usenet conversations as is the largest powerful reference tool, offers the insight into the history and culture of the Internet. Google Groups have more than 700 million postings in more than 35,000 topical categories. Google Image Search Comprises of more than 330 million images, Google Image Search enables users to quickly and easily find electronic images relevant based on the variety of topics, including pictures (celebrities, popular travel destinations). The advanced features also include image size, format (JPEG and/or GIF), and coloration. It also restricts the searches to specific sites or domains. The Google Groups Usenet archive uses for the different contexts at Google: Spelling Correction and Query Classification.

3.4.2. Some statistics

The World Wide Web, also referred to as the web, has radically changed the way in which we produce and consume information. Notably, it contains billions of documents which makes it likely that some document will contain the answer or content a user is searching for. The web has been growing at a tremendous rate and is in a constant state of flux: some

documents change over time, some just disappear completely, and yet others are newly created. To give an idea of how the web has grown over the last decade [50]: in 1999, it was estimated that the web consisted of 800 million web pages and that no web search engine indexed more than 16% of the web; in 2005, the web was estimated at 11.5 billion pages; in 2008, Google announced the discovery of one trillion (1,000,000,000,000) unique URLs on the web; and in 2013, Google updated this number to 30 trillion (the same goes for the number of websites). The immense size of the web, its continuous growth, and its highly dynamic nature, make it challenging to build a web search engine that is effective, fast, and that can scale up to web proportions [51]. It is difficult to assess to what extent major search engines like Bing and Google actually keep up with the growth of the web; however, they often do manage to return satisfying results within just a few milliseconds [50].

According to the World Wide Web size[19], the evolution of websites these last two years is shown in figure 3.4.

[19] http://ww.worldwidewebsize.com

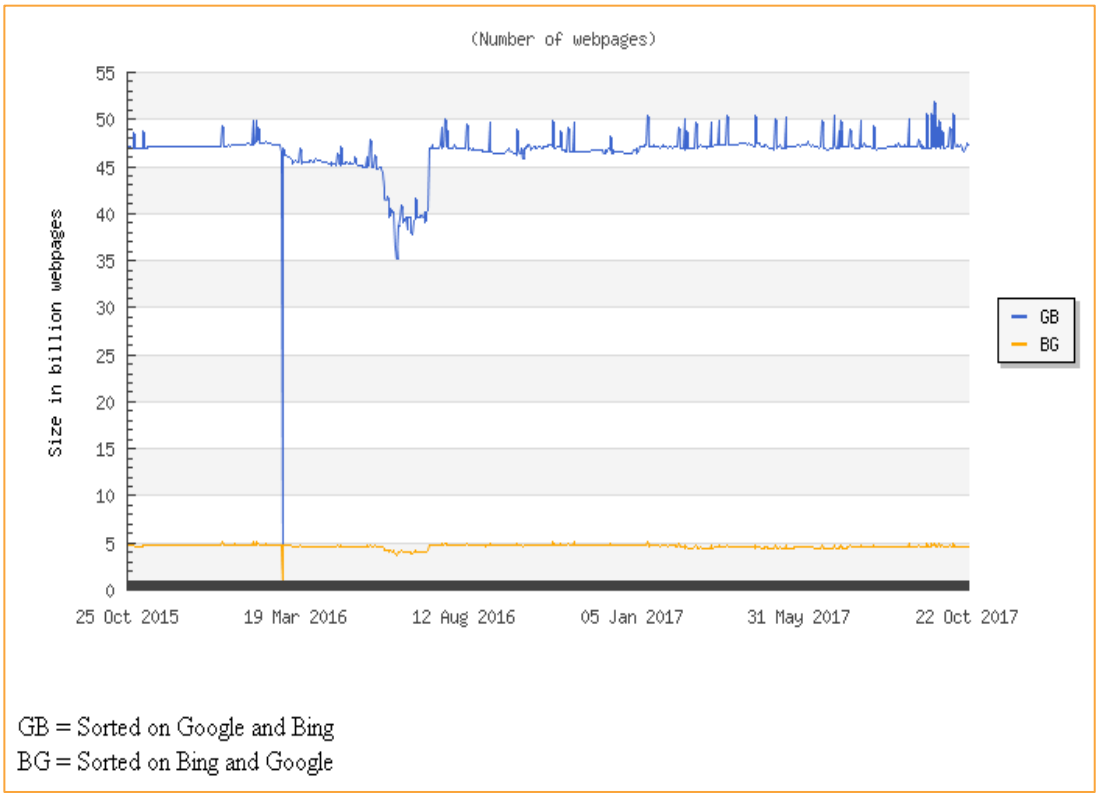


Figure 3-4: Evolution of the number of webpages these two years.

The extremely large size of the Web makes it generally impossible for common users to locate their desired information by browsing the Web [15]. As a consequence, efficient and effective information retrieval has become more important than ever, and the web search engines (or information retrieval system) have become essential tools for many people.

According to [46], the global marketing share percentage, in terms of the use of Search Engines heavily favours Google, with over 77%. This again reinforces the fact that Google are the market leaders, however it also highlights that the "Others" such as Yahoo, Bing and Baidu etc still hold a large audience and it would be silly to simply ignore them.

The number of people using internet search engines is increasing year on year and is almost unfathomable. At... 6,586,013,574 searches a day worldwide which in "word-terms" equates to six billion, five hundred eighty-six million, thirteen thousand, five hundred and seventy-four.

To put it into perspective there are on average around 500 million tweets per day, so 500 million X 13.

Figure 3.5 shows the number of daily Searches per Search Engine.

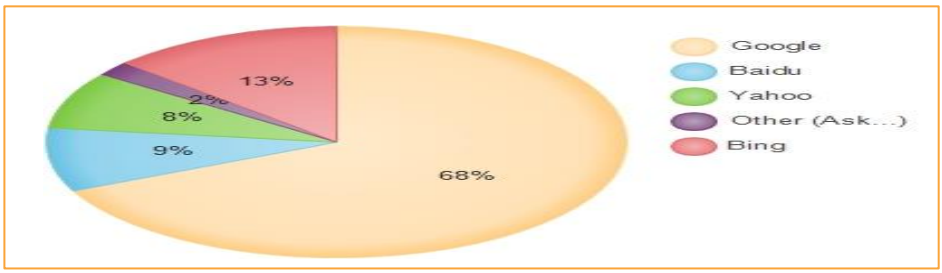


Figure 3-5: Number of daily searches by search engines in billions [46].

The final point concerns the device users choose to use in their daily search operations. It was several years ago now that Google announced that they had passed the tipping point whereby the number of Mobile searches had taken over that of desktop stating... “More Google searches take place on mobile devices than on computers in 10 countries including the US and Japan.”

The graph below (figure 3.6) highlights the rate at which Mobile has surpassed Desktop search.

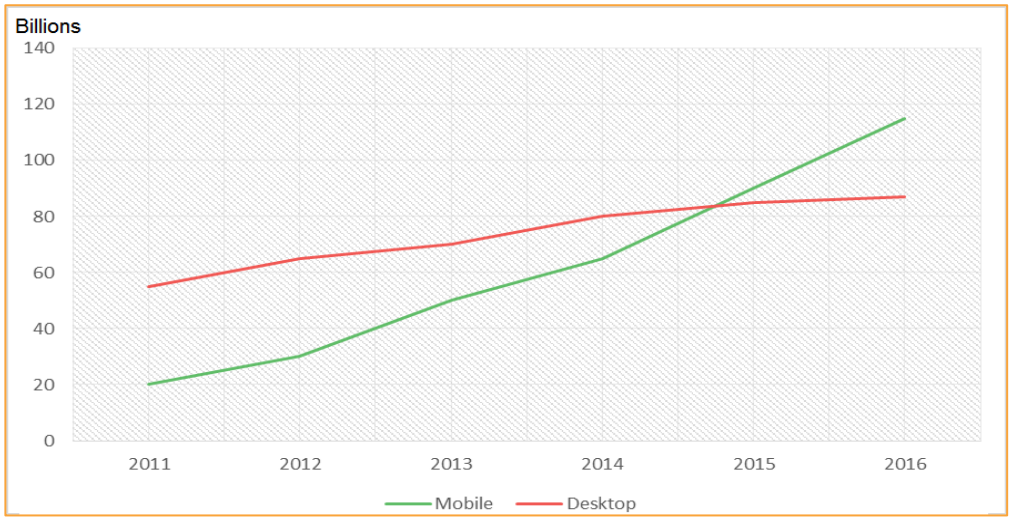


Figure 3-6: U.S. Local mobile search vs. Desktop search [46].

3.5. Web Information Retrieval System

Understanding the information needs of such a mass of users with varying interests and backgrounds, web search engines must also strive

to understand the information available on the Web. In particular, the decentralised nature of content publishing on the Web has led to the formation of an unprecedentedly large repository of information, comprising over 30 trillion uniquely addressable documents. While the lack of a central control is key for the democratisation of the Web, it also results in a substantial heterogeneity of the produced content, from its language and writing style, to its authoritativeness and trustworthiness [52].

Understanding the web graph is crucial for understanding the structure and dynamics of the Web itself, but it also plays a fundamental role in designing effective and efficient web search engines [37]. To cope with this challenge, web search engines are typically designed with three core components: crawler, indexer, and query processor. Figure 3.7 provides a schematic view of these components. In particular, a crawler browses the Web in order to collect documents into a local corpus. This corpus is processed by an indexer, which produces data structures for efficient access to the contents of the corpus. The resulting structures are then used by the query processor, in order to produce a ranking of documents that are likely to be relevant to a user's query. In the remainder of this section, we briefly describe each of these components [52].

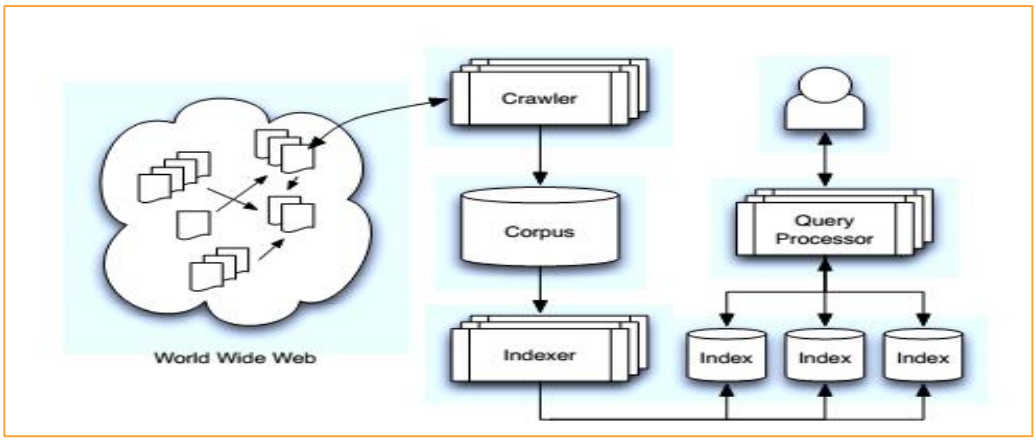


Figure 3-7: Schematic view of a web search engine [52].

3.5.1. Crawling

Crawling is the process by which search engines collect documents from the Web into a local corpus. Such a corpus can be then processed

by the search engine in order to allow users to efficiently locate information. The overall goal of crawling is to build a corpus as comprehensive as possible, in as little time as possible [52]. As shown in Figure 3.8, at all times, the crawler maintains a list of URLs to be visited, the so-called crawling frontier, which is initially filled with a few seed URLs. While the frontier is not empty, the next URL to be visited is removed from it and downloaded by a fetcher module, after a DNS resolver translates the URL domain into an IP address. The fetched document is processed by the crawl controller and the extracted contents are stored locally for indexing. The URLs extracted from this document—and the document's own URL, for continuous crawls—are inserted back into the frontier, so that they can be visited by the crawler at a later time.

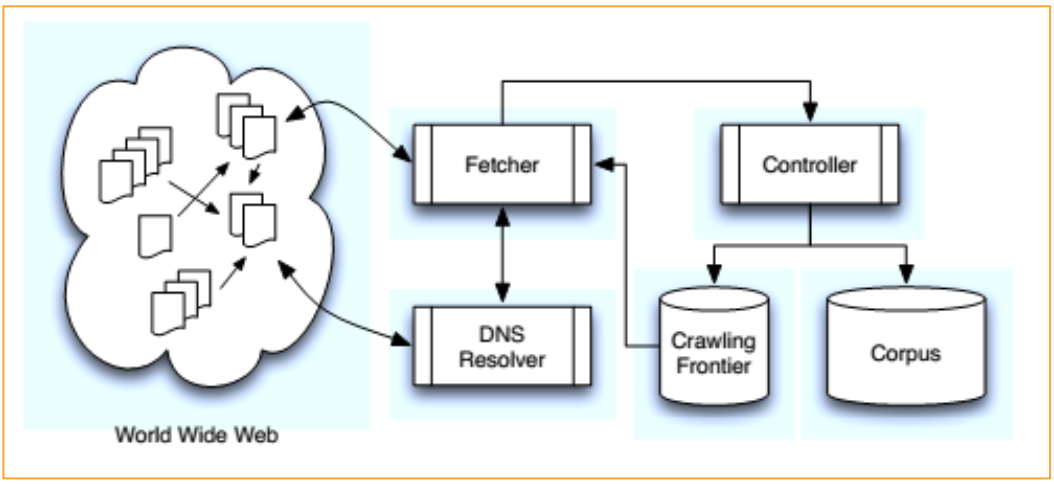


Figure 3-8: Schematic view of a crawler [52].

Not all content on the Web can be crawled directly. On the one hand, the surface Web comprises content that is reachable by following hyperlinks between documents in the web graph. On the other hand, the deep Web comprises content that is generated dynamically, typically in response to a user action (e.g., after submitting information through a form, or entering a password protected area). As a result, the deep Web is orders of magnitude larger than the surface Web and can only be sampled with special-purpose crawlers [49]. Nevertheless, the surface Web is itself massive, making crawling a challenging task. While new documents are created and existing ones are modified at a massive scale, the resources available for crawling—notably, storage and bandwidth—are limited. To

make crawling scalable, web crawlers must consider carefully which URLs to visit, and how often to revisit each URL.

There are different kinds of crawlers [49]:

- For a general purpose web search engine, given some initial links, the crawler iteratively downloads document contents and follows discovered links in pages.
- For a domain specific search engine, the crawler should only download documents that are about its interested topics.
- For an enterprise search, the crawler should not only download all internal documents by links, but also need to scan folders to find different files.

3.5.2. Indexing

Usually, downloaded documents are not plain text but has other formats, e.g. HTML, PDF, Word etc. These files have to be parsed to get their plain text and the content and metadata (e.g. title, keywords, and author) of processed files are stored in a file database [49].

The overall goal of indexing is to create a representation of the documents in the local corpus suitable for automatic processing by a search engine [53]. The devised document representations are then stored in appropriate data structures for efficient access by the query processor. Given a corpus of documents (e.g., crawled from the Web), each document is indexed following the general process illustrated in Figure 3.9. In this process, two main data structures are created [52]:

- The first of these is a lexicon, which stores information for all unique terms in the corpus, such as their total number of occurrences and the number of documents where they occur.
- The second structure is an inverted file, which stores, for each term in the lexicon, a posting list, comprising information on the occurrence of the term in different documents.

- Indexing may be performed in a single batch, in which case the whole corpus must be re-indexed when there is an update, or incrementally, through small atomic operations.

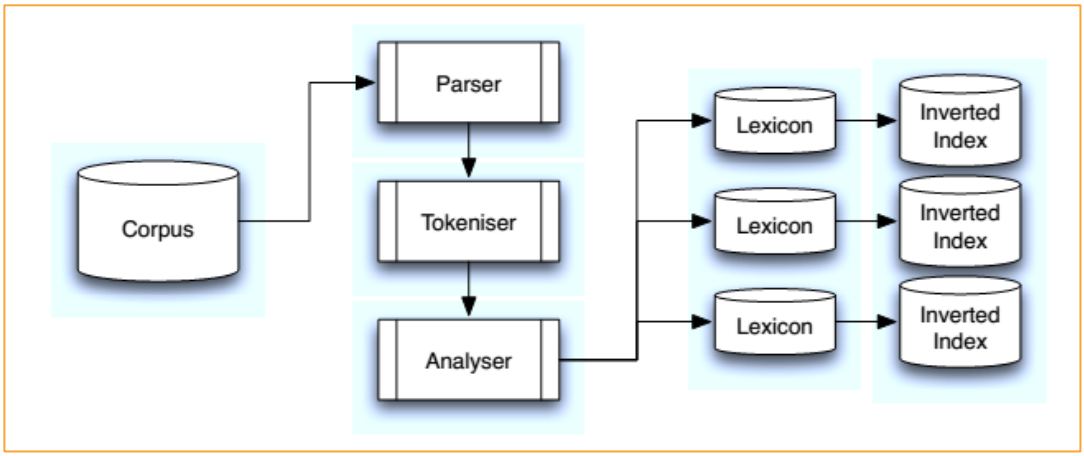


Figure 3-9: Schematic view of an indexer [52].

3.5.3. Query Processing

For several keywords from users, a search engine retrieves a list of relevant documents and ranks them according to specific metrics of relevance measurement [49].

Query processing is the component responsible for answering users' queries [52]. As illustrated in Figure 3.10, when a user poses a query, the search engine examines its index structures to locate the most relevant documents for this query. Given the size of the Web and the short length of typical web search queries, there may be billions of matching documents for a single query. In order to be effective, a search engine must be able to rank the returned documents, so that the most relevant documents are presented ahead of less relevant ones [53]. Query processing consists of three basic operations [52]: Initially, the search engine receives a query, as a typically short and often underspecified representation of the user's information need. This query may go through a series of query understanding operations, aimed to overcome the gap between the user's information need and the ill-defined representation of this need in the form of a query. This stage is important, since misinterpreting the user's information need implies that relevant documents may never be returned, regardless of how sophisticated the

subsequent retrieval is. Once a suitable representation of the user's query has been created, a matching process retrieves the indexed documents that contain the query terms. Lastly, to ensure that the user is presented with the most likely relevant documents for the query, the retrieved documents are scored and sorted by a ranking process. Query understanding aims to derive a representation of the user's query that is better suited for a search engine [52]. Typical query understanding operations include refinements of the original query, such as spelling correction, acronym expansion, stemming, etc.

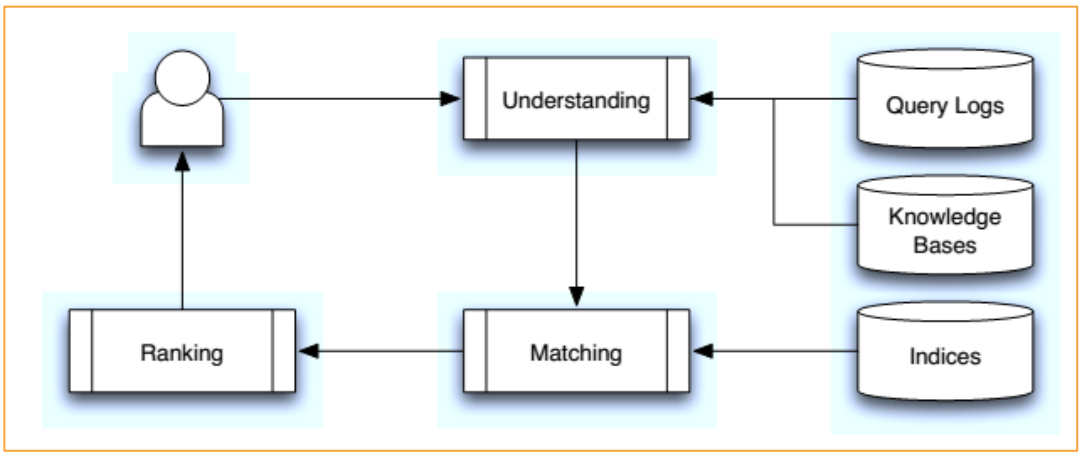


Figure 3-10: Schematic view of a query processor [52].

Other common query understanding operations are query topic classification, that aim to restrict the scope of the retrieved documents, and query expansion, to enhance the query representation with useful terms from the local corpus, or from external resources, such as a query log or a knowledge base such as Wikipedia [52].

The main challenge for semantic search approaches in the input query formulation is to identify and adopt the query format that provides the highest (balanced) level of expressiveness and ease of use [54].

- *Formal (Structural) approach*: The input query is expressed in one of the formal query languages for RDF (e.g. SPARQL or SeRQL) which are used to retrieve data from an RDF model.

- *Natural Language (NL) approach*: The input query is expressed using a natural language such as English (e.g. 'Where is the Saad Dahlab University located?').
- *Keywords-based approach*: The input query is a set of keywords of interest to the user (e.g. 'location University Saad Dahlab). Some of the systems employing this approach are Swoogle[20], Watson[21], and Sindice[22].
- *Graph-based approach*: The input query is formulated using a graph-based interface that explores the search space. This approach aid users in constructing their queries by visualising the data available and the possible ways of querying it.
- *Form-based approach*: This approach is similar to the graph-based approach in visualising the search space, while being different in using forms instead of graphs as the interface to build the query.
- *Hybrid approach*: This approach uses a combination of the previous approaches as the query format.

3.6. Web retrieval challenges

Web IR system are confronted with new challenges that traditional IR systems do not need to address. These challenges can be divided in two main classes, which are challenges related to the huge amount of data and challenges related to the user interface [42].

3.6.1. Challenges Related to the Data Amount

- Information Acquisition as the WWW is a widely distributed network of documents and information that is constantly changing, new methods and tools are necessary to gather all information available on the WWW. An even more challenging factor for information acquisition is the heterogeneity of the WWW. This heterogeneity requires robust (in term of failure tolerance and handling spam) and

[20] http://swoogle.umbc.edu/2006/

[21] http://watson.kmi.open.ac.uk/WatsonWUI/

[22] http://sindice.com/

polite (crawlers must respect web server policies for the crawl rate) crawlers.

- *Index Size*: The index size of a web IR system is of a few orders of magnitude larger than that of traditional IR system. It means that a large-scale distributed architecture is necessary to scale with the growing number of information sources.
- *Quality of Data*: Due to the open nature of the WWW, a problem arises with the quality of the information sources used in web IR sources. In most cases, there is no editorial process and therefore no control about the quality of web pages as it is for documents in traditional IR system. So data can be false, invalid, and poorly written or with many typos.
- *Up-to-dateness*: Web pages change very fast and thus search engines need to include changes as fast as possible. Because of its size and the different update intervals of each website, intelligent crawling strategies are needed.
- Spam Web users do not look solely for information, but also for items they would like to purchase. Hence, web content creators with commercial interests have a strong incentive to create web pages with a high ranking. This leads to spam web pages which are created to manipulate search results of a web IR system. Hence, a web IR system needs to identify spam pages from valuable web pages.
- *Near Duplicates and Shingling:* Almost 30% of web pages on the web are very similar or almost identical. These duplicates are to some extent legitimate copies in order to provide redundancy and increase accessibility. Nevertheless, they also increase storage usage and processing overheads. Therefore, feasible algorithms for duplication detection are necessary.

3.6.2. Challenges Related to the User Interface Problems

- *User Query Needs*: Traditional IR systems were typically used by information professionals who are trained to author well defined queries in order to get useful results from a system, which they

knew well. In contrast, users of web IR systems are not trained to author a well formed query, but rather to use only a few keywords.

- *Interpretation of User Query*: Because most users are untrained for the formulation of queries, it is likely that queries are short and simple, which leads to thousands of web pages in the search results. Therefore, an efficient ranking for search result items is necessary. Link analysis has been the most efficient ranking method for web pages in recent years. In addition, search engines use language settings in browsers and geo-location of IP addresses to filter and rank search engine results to provide a better search experience.

Moreover, the nature of the Web poses a number of challenges to classic IR systems. Several of these are outlined in this section [44].

3.6.2.1. Crawling

Web content is distributed across countless Web servers scattered across the Internet, therefore unlike IR collections it is a prerequisite to assemble a snapshot of the Web's content (a crawl) before constructing a representation of it through indexing. Typically snapshots are assembled by automated applications which engage in crawling; the process of recursively fetching documents using a pool of document locations (URLs) which is replenished with discoveries of new URLs referred to in the hyperlinks of fetched documents. Although implementing rudimentary crawlers is relatively straight forward, Google intimate that industry strength crawlers capable of assembling the large crawls typical of major search engines requires a great deal of engineering.

3.6.2.2. Diverse Search Requirements

In tandem with developments in Web technology and Web programming, the Web is increasingly functioning as a platform for a growing number of on-line services and Web applications such as Internet banking and Webmail. Changes in the use of the Web induce changes in the intent of Web searchers as seen before and it results in different types of queries. Although the category of informational searches is common to

both IR and Web IR, the abundance of content on the Web demands greater discrimination when returning results for broad-topic searches of this type.

3.6.2.3. Search Engine Persuasion

Search Engine Persuasion, coined SEP refers to deliberate manipulation of Web search engines in order to boost the ranking of documents in search results. SEP is far more common on the Web than in traditional IR contexts where there is relatively little competition for the attention of collection audiences. Due to the commercial motives of traffic hungry Web site owners, manipulation of this sort ranges from being deceptive to fraudulent. Understandably, efforts made by commercial search engines to maintain the integrity of their search results tend not to be made public.

3.2.2.4. Incorrect Content

Since there are generally no content controls on material published on the Web there is a higher chance that Web documents contain incorrect information than traditional IR collections. Web searchers tend to feel more assured by information that emanates from important sites. The challenge of retrieving correct content is therefore closely tied to that of retrieving authoritative content.

3.6.2.4. Duplication

Duplication of content is far more likely in the context of the Web than it is well controlled collections. Duplication poses a problem for both Search engines and searchers alike. Search engines are computationally burdened by the crawling, indexing and storage of duplicate content and Internet searchers find the presence of duplicates amongst retrieval lists a nuisance.

3.7. Conclusion

Traditional IRS are based on the well-known technique of “bag of words” (BOW) representation expressing the fact that both documents and queries are represented as bags of lexical entities, namely keywords.

A keyword may be a simple word (as in “computer“) or a compound word (as in “computer science”). Weights are associated with document or query keywords to express their importance in the considered material. The weighting scheme is generally based on variations of the well-known tf*idf formula.

A key characteristic of such systems is that the degree of document-query matching depends on the number of shared keywords. This leads to a “lexical focused” relevance estimation which is less effective than a “semantic focused” one [5]. Indeed, in such IRS, relevant documents are not retrieved if they do not share words with the query, and irrelevant documents that have common words with the query are retrieved even if these words have not the same meaning in the document and the query. The problems mainly stem from the richness in terms of expressive power, yet the synonymy and polysemy inherent in natural language.

In a digital world search is essential, pervasive, ubiquitous and ceaseless. Yet, search remains unsatisfying in many ways. Why is this the case? When we search, we are after an answer. But today’s search engines do not have enough intelligence to provide answers. They are also not advanced enough to interpret naturally phrased queries and understand the ambiguities of ordinary language. And so we type in keywords and get links or lists of results that are more or less relevant depending on the type of search, our query and the competence of the search engine. There is always a further step when we have to select an item from a result set and assess it. Hopefully, the answer eventually emerges, but there are no guarantees.

The ultimate search engine would directly answer our query. Perhaps it would even anticipate our query and present the answer before we even ask. To answer us, the search engine would need to understand our query, our context and the corpus of knowledge at its disposal. Such kind of search engine remains a futuristic vision for the most part. But search technology is advancing rapidly and recent innovations are beginning to overcome the limitations of the conventional keyword

query/result list model. Meaning in search is emerging along actionable meaning in the guise of intelligent personal assistants that provide deeply contextual search triggered through a natural, conversational interface.

CHAPTER 4 CONCEPT-BASED INFORMATION RETRIEVAL

4.1. Introduction

Semantic Web was an extension of the Web. The web depends on visual representation of information through HTML tags. This visual representation makes information clear for humans to understand but very difficult for machines to understand and process. For example to emphasize something it could be in a different font or colour. Some form of extraction is required to strip off the information part from the presentation part. Other techniques are used to infer meaning from this information; this leads to an increased complexity in the agents dealing with the web.

Another problem with the web was the fact that different terms were used to represent the same meaning, for example in a shopping site the shopping cart could be referred to as cart, while another would refer to it as shopping basket or basket for short, yet another site could refer to it as shopping bag. All these words refer to the same meaning or the same semantics, which is very obvious to humans while it is unknown to software agents. These agents have to be explicitly informed that the previous terms are all the same. Another example comes from the fact that the web is multi lingual; an English shopping website would use the word "*price*" to refer to an item's price, while a French website would use the word "*prix*", an Arabic site would use the word "ثمن". An agent that is looking for a product and comparing prices to retrieve a list of the cheapest sites would have to be familiar with these terms.

The Semantic Web targets solving these problems by providing not only the data but also metadata that describes explicitly what this data means. This form of data annotation makes an agent understand the semantics behind the data and thus allows for better interpretation and allows a better inter agent communication and collaboration. As stated by Berners-Lee [4], "this notion of being able to semantically link various

resources (documents, images, people, concepts, etc) is elementary. With this it can begin to move from the current Web of simple hyperlinks to a more expressive semantically rich Web, a Web where it can incrementally add meaning and express a whole new set of relationships (hasLocation, worksFor, isAuthorOf, hasSubjectOf, dependsOn, etc) among resources, making explicit the particular contextual relationships that are implicit in the Web. This will open new doors for effective information integration, management and automated services". The Semantic Web promises a solution in which the web becomes one big knowledge base and everyone has access to it. In order for this to happen there should be supporting technology that allows for such annotation in a formal and unified syntax, such annotation are RDF/RDFS and OWL which are standards set by the World wide web Consortium (W3C). Also reasoning on the Semantic Web promises for more intelligence in services provided by the web such as personalized Information retrieval, e-learning and many other applications where agents would pull the information and process it having a better understanding of its meaning.

4.2. Semantic web vision

After the invention of the World Wide Web, Tim Berners-Lee proposes the Semantic Web. The Semantic Web simply means the web of meaning. In the web, information is presented in natural human language which is not rich enough to convey formal meaning and therefore it is not machine processable. This current web contains millions and millions of resources such as HTML files, documents, images and graphics, and media files. These resources contain huge amounts of information scattered in various web pages and documents. The current web is a web of documents and understandable only to humans. This makes information retrieval processes very hard; humans alone cannot deal with this huge amount of resources on the web. Software agents or machines could help in this process but a difficulty arises from the fact that machines do not understand human language. Trying to make machines act as humans is a very complex task and needs a lot of training [5].

The basic idea of the Semantic Web is to give information a well-defined meaning, thus better enabling agents and people to work in cooperation. W3C states "The Semantic Web is about two things. It is about common formats for interchange of data, where on the original Web we only had interchange of documents. Also it is about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing".

The challenge of the Semantic Web was to provide a language that expresses both data and rules for reasoning about the data and that allows rules from any existing knowledge-representation system to be exported onto the Web. As stated by Berners-Lee "Making the language for the rules as expressive as needed to allow the Web to reason as widely as desired" [5].

Tim Berners-Lee, the inventor of the World Wide Web, proposed the concept 'The Semantic Web' and presented his vision in his book Weaving the Web [40] as follows: "*I have a dream for the Web [in which computers] become capable of analysing all the data on the Web - the content, links, and transactions between people and computers. A 'Semantic Web', which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines will finally materialize*" [49].

Tim Berners-Lee, James Hendler and Ora Lassila presented a vision for the next generation of the web in May 2001 with an article entitled "The Semantic Web" [55]. In the article, Berners-Lee and his co-authors vivified the idea of a semantic web through a detailed use case that imagined an intelligent web software agent capable of checking calendars, making appointments, finding trusted persons and places and more. The "semantic web agent" does all of this autonomously, drawing inferences on behalf of its human user. Berners-Lee tells us that all of this can be achieved without futuristic, sci-fi-like Artificial Intelligence (AI). It becomes

possible through the encoding of meaning or semantics into web pages by their authors. But it is not just web pages. Interestingly, the vision also included intelligent physical entities like home appliances that adjust their settings in concert with the needs of the household residents[23].

Inherent in the semantic web vision are three core concepts [56-57]:

- A web imbued with meaning expressed using Resource Description Framework (RDF) “triples” (subject/predicate/object).
- Ontologies identifying the things that exist, their definitions and their relations.
- Software agents (“semantic web agents”) capable of inference and autonomous action.

4.3. Semantic search definition

The word semantics is derived from the Greek word semantikos,"significant" from semaino, "to signify, to indicate" and that from sema,"sign, mark,". Linguistically, it is the study of interpretation of signs or symbols which are used for some specific contexts. Semantic analysis is the process of relating syntactic structures, from the levels of phrases, clauses, sentences and the whole text, to their language-independent meanings [56].

Semantic search provides insight into unstructured documents stored by extracting the relevant keywords and index statistics in the database. Then, it is also used to identify these keywords and index similar or related documents [57]. Moreover, Semantic search tries to improve search by understanding the contextual meaning of the terms and tries to provide the most accurate answer from a given document repository. The technologies behind semantic search are mostly used to access unstructured data [2].

In fact, no unified definition of semantic search exists. It has been used by different research communities including Information Retrieval,

[23] Nowadays, this concept is referred to as The Internet of things

Natural Language Processing and the Semantic Web to describe different approaches and strategies employed to improve search performance and user experience. However, they all share the broad goal, which is to better understand users' information needs (represented in their queries) and/or the Web/domain content; and to improve the matching required between performance and experience [54].

Figure 4.1 shows an abstract architecture for semantic search in which the basic steps in the search process are illustrated. The user inputs their query in a specific input format that is adopted by the system (e.g. as a NL sentence or using a view-based interface to construct the query). The query is then processed and transformed into a formal representation as required by the underlying query engine. The amount of transformation is influenced by the query input approach as shown in Figure 4.1. The formal query is then executed against the search space which either describes a single domain (closed-domain) or multiple ones (open-domain). Finally, results generated from this step – documents or data – are presented to the user in a format chosen by the system (e.g. ranked list of documents or NL answers).

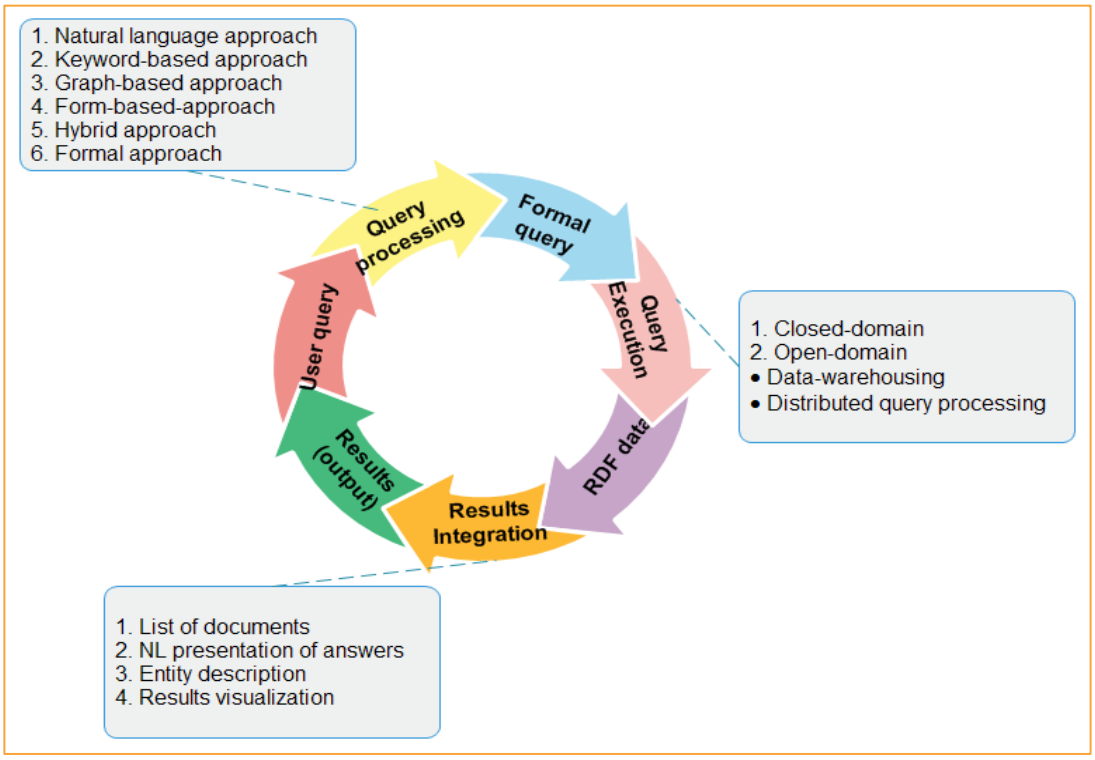


Figure 4-1: Abstract architecture of a semantic search [54].

4.4. The structure of semantic web

The Semantic Web combines several existing technologies to convert the World Wide Web from a web of document to a web of data. Tim Berners-Lee proposes a layered approach for achieving the Semantic Web [58].

Figure 4.2 shows the different layers of the Semantic Web. Logic or reasoning is one of the major important issues for Semantic Web and it is an important design issue when creating a Semantic Web agent.

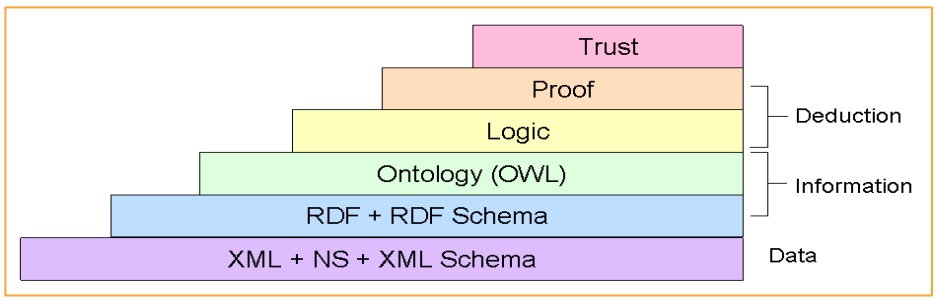


Figure 4-2: Semantic web layers [58].

The architecture of the Semantic Web is realised by the Semantic Web Stack which arranges layers of languages and technologies in a

hierarchy. Each layer of the hierarchy uses the capabilities of the layers below, whereas the architecture still evolves as its layers are materialised. The Universal Resource Identifier (URI) and Unicode language are found at the very bottom of the hierarchy for supporting the necessary unique identification of all web resources and for all natural languages. In what follows a brief introduction of each layer is given [1, 49, and 59]:

- *Extensible Markup Language (XML):* XML is designed for writing structured documents with self-defined vocabulary. It strong expressive power as users can add arbitrary structure, but it does not contain any meaning about these self-defined structure.
- *Resource Description Framework (RDF):* RDF is a basic data model for standardizing the definition and usage of web resources. Its basic element is triples in the format <Subject, Relation, Object>, which models the relation between objects. RDF Schema (RDFS) extends RDF with some hierarchical modelling primitives, e.g. class and property, subclass and sub-property.
- *Ontology Language*: RDFS can be viewed as a primitive language for writing ontologies. Ontology languages, like OWL, DAML+OIL, defines more complex relationships. The set of technologies (Web Ontology Language OWL and RDF schema) enable reasoning and facilitate knowledge sharing and reuse of semantic web information.
- *Logic*: Logic layer exploits the data and rules expressed by ontology language to infer hidden knowledge or find relevant information. This is what the application-level agents used to perform all kinds of tasks.
- Proof and Trust: Proof layer provides validation of the inferred knowledge; trust layer takes use of digital signature and other tools to guarantee users the authenticity of semantic information.

The Semantic Web Stack is still evolving and periodically revised to include additional layers that support the semantic web technology.

As Tim-berners Lee said, if the Semantic Web is properly designed and fully exploited, heterogeneous web resources can gain strong

capability and it can even assist in the evolution of human knowledge as a whole.

4.5. Searching the semantic web

The difficulty is that semantics is not necessarily accessible using surface syntax or word order. Words, and word associations, are interpreted by humans, and interpretation is influenced by memory, imagination, emotion, world knowledge, social and physical factors. By consequence, semantics is not always explicit in text and an information need is not always explicit in a query. This results in a gap between an information need and a query that is not addressed by linguistic processing [29].

Table 4-1: Conceptual perspective VS linguistic perspective.

Parameter	Conceptual perspective	Linguistic perspective
Relationship element	Models of relationships among objects	Models of relationships among words
Semantic backbone	Ontologies help to capture entities in the real word and their relationship	Taxonomies, thesauri, dictionaries, corpora for capturing entity names and relationships
Inference element	Inference along domain specific relations	Inference along linguistic relations, e.g. broader/narrower/ functionally related terms
Search basis	Knowledge based search	Natural language/ keyword based search
Search elements	Entities, relationships, documents	Document

Alternatively, a large corpus can also be used to fetch background information of the words in the form of their contexts in order to find the relatedness among the words.

As the Semantic Web grows in size there will be an increasing need of searching for information. Users will want to perform search queries and expect Semantic Web documents that best match their query as results, in analogy to web

search; only the expected precision of Semantic Web search should be better due to the better understanding of the search terms [58]. It must be noted that Semantic Web documents are different than conventional web documents, information viewed in Semantic Websites are what the developer wants to present but the semantics behind the presentation is what matters, while with web documents the presentation is simply the way of formatting the looks of the information. For example a computer shopping site in Dutch, Arabic, and English would share the terms from a computer shopping ontology, while the terms are presented in the three different languages but their semantics is the same because they have a common source of semantics. A search engine searching for certain computer specifications would perform its search and return the result based on the user preferred presentation.

There are several techniques to implement semantic search. It is evident to distinguish between structured and unstructured data. Unstructured data is usually a text which does not contain any machine readable semantic information. The best search or indexing method depends on the structure of your data [2].

One type of structured data are ontologies. An ontology formally describes available concepts in a specific domain. It describes relations between words like “dogs are enemies of cats” or “dog is an animal”. With this information it is possible for a search engine to “understand” a query by travelling an available ontology.

One possibility to query structured data is conceptual graph matching. In this method each query and the data are represented as trees of concepts (ontologies). The search engine compares the query with each tree in the database and finds the best matching tree.

One way to represent such a data structure is by using Resource Description Framework (RDF). RDF is a framework used to describe data. It can be used to describe data structures much like a domain class model. Using this net of connecting concepts, it is possible that the search engine understands the search context and can retrieve more accurate

search results. Using this conceptual network it is also possible to do word sense disambiguation. If the user searches for a word with multiple meanings, the search engine chooses the most probable meaning by examining the other words in the query and the available concepts.

The problem of the previous mentioned techniques is that they require structured data – they are not suitable to access unstructured data like text documents. To query unstructured data, the search engine has to index all documents, split it up into keywords and score them depending on statistical analysis. Most search engines index all the documents with a term extraction or a phrase extraction algorithm[24] [2].

The next step is to remove noise words or stop words. Some search engines do not remove stop words to better support exact phrase search.

Another important thing in semantic search is inflection and stemming. Inflection is the modification of a word to another form – for example another tense, case, etc. Transforming a word back to its stem is called stemming. Using word stemming the search engine only uses the stems of the words and therefore can match words even if the user searches with words in the wrong form. We will tackle this subject in details in chapter 6.

A search engine should also support synonyms and replacements. The synonymic noun database is used to match nouns with the same meaning whereas replacements are used to match abbreviations or wrong spelled words. Using a statistical database every keyword will get a score depending on its relevance in the language. The scores are pre-calculated from a big pool of random texts like books, magazines, web pages, etc. Using this database words with high appearances in a language are scored lower than rare words [2].

[24] Phrase extractions will bundle multiple words whereas a term extraction algorithm simply divides the text into single word chunks.

4.6. Ontologies in Information Retrieval

The term Ontology originates from the discourse of philosophy. In philosophy, the notion of concept refers to the fundamental nature of existence [26]. It should have a definitional structure derived from a list of features. A feature entailed by the definition of a concept must be both necessary and sufficient for the membership in the class of things covered by this concept [1].

In order to understand the basic structure of the world and the study of existence, the word ontology has been connected with a branch of metaphysics. The problem is that the philosophical definition of ontology is not easy to port to the scientific domain. ”An ontology is a detailed model/picture/schema of a slice of reality which is based on the facts that know about that reality. This model /picture /schema is a description of some of the things and some of the relationships between the things that are known about that reality”. Another definition of the term Ontology is the following: ”Ontology defines the terms used to describe and represent an area of knowledge”. These ontologies can be shared by different applications, people and databases within a domain [58].

A domain can be an area of knowledge, like medicine or a specific subject area. The definitions of ontologies are machine readable and they describe basic concepts in the domain and the relations between them. The knowledge, which is encoded in ontologies, is reusable due to the fact that the encoded knowledge can span different domains [58].

Ontologies are able to specify the following kinds of concepts, which enable the description of almost every knowledge [1].

A concept means a set or class of individual objects (individual in short). Those objects have the same types of features, called atomic (primitive) terms. Features of an individual should logically satisfy the definition of a concept, if the individual belongs to the type defined by that concept;

Relationships can be specified between concepts or individuals. The most representative relationship is the concept subsumption (⊒). A concept C subsumes a concept D (C ⊒ D) (or D is subsumed by C) if any individual in D is an individual of C. In this case, C is called a parent class of D; while D is called a subclass of C. If any individual of C is also in the meanwhile an individual of D, C and D are equivalent (C ≡ D). Besides, the non-hierarchical relationships between categories can be described by roles. A role R describes the value restriction (∀R.C) or the existential restriction (∃R.C) of a relation in the world. It is considered as a binary predicate and the concept C is called as the range of C or a role filler; while the concept being described is named as the domain of the role.

Another kind of role restrictions rules the cardinality of the range set, which is called cardinality restriction.

With their relationships, concepts can be organized in a structure, called ontology. In short, it is a formal, explicit specification of a shared conceptualisation [60].Formalizing the representation of knowledge and achieving knowledge interoperability are gaining increasing importance in many areas like Medicine, Artificial Intelligence and the World Wide Web. Ontology becomes the standard approach for achieving this [49].

There are many motivations for developing and using ontologies [58]:

- To enable reuse of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyse domain knowledge.

4.6.1. Ontology search

Traditional search is based on keyword matching, which can hardly catch the actual conceptualization associated with user needs and contents. There are research efforts trying to aid search with ontologies to improve the search accuracy. Several procedures exist in integrating

ontology search [49]: input analysis and translation, ontology annotation of documents and various ontology exploration techniques.

Ontology search means searching for documents with the support of underlying ontology entities [49]. We assume that the annotation of ontologies to documents are accurate and these ontologies describe the meaning of these documents.

Firstly, the ontology search engine processes a user query. The processing module disambiguates user’s query and transform the keyword-based query to an ontology query. For example, if a user searches for "Rock musicians in Africa", by querying the ontology base, the search engine should learn that "Rock" does not mean a kind of stone but a music genre, as the existence of musician implies.

Therefore, an ontology query like <rock, ?, musicians>, <musicians, ?, Algeria> will be produced. Then, the document retrieval process can be performed in several ways. A classical way is to use the vector space IR model, which represents both query and document ontology as a vector and compares their similarity.

Ideally, with the support of ontology, the accuracy of search result should be highly improved. However, a severe problem is that the ontology knowledge base should be complete enough for the thesaurus of common search and documents. Otherwise, if the search engine only relies on ontology search, the search accuracy may even drop due to frequent missing in matching ontology. Therefore, a search engine that combines text search and ontology search techniques may give better results in the long term [49].

Indeed, complete conceptual indexing is hard to achieve in realistic collections. The reasons are twofold: firstly, domain ontologies may be hampered by weak coverage of all content aspects of the documents and secondly, high quality indexing requires human expertise and is thus a tedious task. This is known as the semantic gap issue. Indeed, automatic or semi-automatic indexing techniques cannot always extract all

significant document concepts. In order to increase ontology coverage and improve both document and user query indexing within conceptual based IRSs, lexical components could be added to the ontology [41].

4.6.2. Hybrid Ontology based IRS

Hybrid IRSs have been designed to take both keyword based and conceptual based indexing units into account. A hybrid ontology based information retrieval system is defined as follows [41]:

“An ontology based information retrieval system is called hybrid when it manages document indexes of different granularities (ontology based and keyword based) and levels (document level and passage level descriptions) during indexing and matching processes and/or during the result presentation stage.”

In hybrid IRSs, the two granularity document descriptions are considered separately since they do not describe the same viewpoint on the document.

This assumption is based on the fact that a document keyword is used as an indexing unit only when information extraction tools failed at connecting it to a concept within the ontology. This independence assumption leads to hybrid IRSs that propose relevance models using two kinds of document/query suitability assessment: conceptual or semantic based and keyword based. A merged strategy of these two outputs is then applied. Three kinds of query are thus possible in such hybrid relevance models [41]:

- Fully semantic or conceptual queries (using only ontology concepts or relations).
- Fully keyword queries (no semantic description of documents is available).
- Mixed queries (both keyword and conceptual queries are available).

There are several ontology description languages for encoding domains knowledge: KL-ONE, KQML, DAML+OIL17, OWL (OWL2), RDF-

S, etc… Their study is out of the scope of this thesis, but readers may find more details here [1].

4.6.3. Some open issues related to ontology based search systems

- Maintaining ontologies will be a real challenge. Since new words are constantly being created as well as new senses are assigned to existing words. To update ontologies regularly and consistently will be a real challenge for maintaining the completeness of the ontology.
- Differently published ontologies pertaining to the same domain; to estimate the reliability of ontology and its publisher will be required before using it.
- Issue of Ontological terms mapping will also crop up like mapping of class ―car‖ in one ontology with ―automobile‖ in the other.
- In heterogeneous Web environment, there is a need for the system to move between ontologies without any need for domain specific reconfiguration, again a big challenge.
- Evaluation benchmarks have not been standardized as yet in case of conceptual perspective of semantic search systems in terms of ontology based semantic search.

4.7. Conceptual IRS

As seen above, conceptual resources such as ontologies are used within the IR community to overcome some keyword-based system limitations. Conceptual IRSs are based on the assumption that document contents are better described by conceptual abstractions of real word entities than by lexical relationships that may be found within it or dictionaries [41]. A cognitive view of the world is thus considered in such systems. The emergence of domain ontologies, boosted by the development of the Semantic Web (in its infrastructure and content), has led to an increase in conceptual IRSs. In these systems, ontology based concepts are used as pivot language for indexing documents and expressing queries. Such conceptual description of the word may also be used as a semantic guide while visualizing documents or data.

Ontology also provides conceptual space in which metrics (semantic similarities or distances) can be deployed to implement the relevance calculus process in IRSs [41].

4.7.1. Conceptual indexing

Concept-based indexing represents both documents and queries using semantic entities, the concepts, instead of (or in addition to) lexical entities, the keywords. Retrieval is then performed in this conceptual space. Concept-based indexing approaches hold the promise that representing documents and queries (or enhancing their BOW representation) using concepts will result in a retrieval model that is less dependent on the index terms [61]. Indeed, in such a model, documents could be retrieved even when the same concept is described by different terms in the query and the documents, thus alleviating the synonymy problem and increasing recall. Similarly, if the correct concepts are chosen for ambiguous words appearing in the query and in the documents, non-relevant documents would not be retrieved, thus alleviating the polysemy problem and increasing precision.

Concept-based indexing relies on concepts identified from the content of the document and the queries based on linguistic knowledge resources (such as dictionaries, thesauri, ontologies, etc.) [61].

It is necessary to distinguish between conceptual and semantic indexing. Conceptual indexing comes from the IR community and relies on concept hierarchy or domain ontology (e.g. the ontology for biomedical investigation: MeSH), where documents are associated with a bag-of-concepts describing their contents. Semantic indexing comes from the Semantic Web community, where metadata are added to a knowledge database to characterize documents (resources). Semantic indexing is also called annotation within the Semantic Web community [41].

4.7.2. Concept identification

Concept identification aims at assigning documents terms to the corresponding entries in the ontology (or any other linguistic resource)

[61]. For this aim, representative keywords are first identified in each document, using classical indexing techniques (tokenization, lemmatization, stop words elimination, etc.). More complex processes can also be integrated to recognize multiword features (nominal phrases, collocations ...). These terms are then mapped onto the ontology in order to identify the corresponding concepts (or senses). An ambiguous (polysemic) term may correspond to several entries (senses) in the ontology, it must be disambiguated. To disambiguate a term, Word Sense Disambiguation (WSD) approaches generally exploit local context and definitions from the ontology. The underlying idea is to estimate the "semantic relatedness" between each sense associated with the target term and the other senses from its local context. WSD is a very challenging technique that disambiguates word senses in a given context [9]. Unlike humans that determine the meaning of words in context without much difficulty, machines may encounter a problem in identifying the meaning of words because words often have more than one meaning. Many efforts have been made to tackle this problem using topic models [9].

4.8. The future of search

4.8.1. The present

The Semantic Web, envisioned fifteen years ago, now exists, but plays a rather marginal role in semantic search so far. It is employed in some very useful basic services, like an e-commerce site telling a search robot about the basic features of its products in a structured way. But the Semantic Web is now here near its envisioned potential (of providing explicit semantic information for a representative portion of the Web) [62].

Web search has improved dramatically over the last fifteen years. We see three major reasons for this. First, the user experience in web search is mainly a matter of high precision. Second, web search engines have steadily picked up and engineered to perfection the standard techniques over the years (including basic techniques like error correction, but also advanced techniques like learning from click through data, which

especially helps popular queries). Third, a rather trivial but major contributing factor is the vastly increased amount of content. The number of web pages indexed by Google has increased from 1 billion in 2000 to an estimated 50 billion in 2015 (selected from over 1 trillion URLs). For many questions that humans have, there is now a website with an answer to that question or a slight variant of it, for example: Stack Overflow (programming) or Quora (general questions about life). Social platforms like Twitter or Facebook provide enormous amounts of informative contents, too.

4.8.2. The near future

Over the next years, semantic search will mature further. The already large amount of text will grow steadily. The amount of data in knowledge bases will grow a lot compared to now.

Knowledge bases will be fed more and more with structured data extracted from the ever-growing amount of text. The basic techniques will be essentially those described in this chapter, but elaborated further, applied more intelligently, and on more and more data with faster and faster machines. This extraction will be driven by learning-based methods, based on the basic NLP methods. Data from the Semantic Web might provide important training information (either directly or via distant supervision). The combination of information from text and from knowledge bases will become more important. The current state of the art in systems like Watson or Google Search is that the text and the knowledge base are processed in separate subsystems (often with the knowledge base being the junior partner), which are then combined post hoc in a rather simple way. The two data types, and hence also the systems using them, will grow together more and more [62].

4.8.3. The not-so future

The development as described so far is bound to hit a barrier. That barrier is an actual understanding of the meaning of the information that is being sought. We said that semantic search is search with meaning. But somewhat ironically, all the techniques that are in use today merely

simulate an understanding of this meaning, and they simulate it rather primitively. One might hope that with a more and more refined such “simulation”, systems based on such techniques might converge towards something that could be called real understanding. But that is not how progress has turned out in other application areas, notably: speech recognition (given the raw audio signal, decode the words that were uttered), image classification (given the raw pixels of an image, recognize the objects in it), and game play (beat Lee Sedol, a grand master of Go). Past research in all these areas was characterized by approaches that more or less explicitly “simulate” human strategy, and in all these approaches eventually major progress was made by deep neural networks that learned good “strategies” themselves, using only low-level features, a large number of training examples, and an even larger number of self-generated training examples (via distant supervision on huge amounts of unlabelled data or some sort of “self-play”). Natural language understanding is just so much more multifaceted than the problems above (speech recognition, image classification, and gameplay). In particular natural language is much more complex and requires a profound knowledge about the world on many levels [62].

4.9. Conclusion

Right now the semantic web techniques cannot replace a human as he still must validate all the results that a computer generates. Still the human is the one to formally define concepts, things, and events, real live and present them in a machine-understandable form.

In addition, even though Semantic Web technologies and ontologies are now widespread and accepted, they are hampered by the fact that they cover few aspects that a document deals with. This is known as the “*semantic gap issue*”.

In short, seventeen years later, we are not close to realizing the vision. There aren’t any software agents roaming an open, semantically enriched web, drawing inferences from reliable factual information and completing tasks for users. On the other hand, bits and pieces of the

vision are blossoming although they are taking shape in ways unanticipated back in 2001. For example [35]:

- The major search engines are increasingly extracting meaning from the web, leveraging semantically tagged pages and large structured knowledge bases.
- Intelligent personal assistants like Siri, Google Now, and Cortana have emerged which resemble the predicted “semantic web agent”. However, so far they lack inferential ability. In addition, they navigate a hybrid digital space composed of the open web and “closed” smartphone apps. This significantly diverges from the more completely open vision sketched by Berners-Lee.

CHAPTER 5 CONTEXT-BASED INFORMATION RETRIEVAL

5.1. Introduction

Nowadays, context-aware systems cover various domains such as smart homes and offices, meeting rooms, health and elderly assistance, and museum guides. In this chapter, we investigate the significance of the inclusion of a context dimension in the overall process of an Information Retrieval (IR) task. Nevertheless, the remarks and results obtained can apply to other domains where the use of context is becoming crucial, yet possible given the technological advance.

Context refers to the circumstances in which an event (an IR computing task in our case) takes place [63]. In fact, context is multi-layered; it extends beyond users or systems. It is not self-revealing, nor it is self-evident, but searchers do integrate context which they understand intuitively in IR theory and practice [17]. In other words, context includes all the intrinsic and extrinsic factors, which are related to a given search task and whose the direct or indirect inclusion in the IR process leads to enhance, whether implicitly or explicitly its effectiveness to convey the right information to the searcher [3].

According to Lombardi [63], seeing the difficulties in most context-aware applications, observations have been made about the nature of context information in pervasive computing systems. Thus, context characteristics are [63]:

- Context must be abstracted to make sense;
- The sensors of which context may be acquired from can be distributed and heterogeneous;
- Context has many alternative representations;

- Context is dynamic, which means that time and place can change the acquired context;

- Context information is imperfect and uncertain.

Different user devices need semantically rich descriptive context models to provide shared understanding and handle environments changes. Therefore, a context-aware system should automatically recognize the situation using various sensors. For example, if a user is typing a query and having the following GPS coordinates 22.7850° N, 5.5228° E, in April at 10AM, then he or she is probably assisting to the traditional Spring celebration 'Tasfit' in the oasis city of Tamanrasset, Algeria [3]. We talk about transforming numeric and discrete data into logical comprehensive ones. Semantic representation of the user's context is the core of most nowadays Contextual Information Retrieval (CIR) works. The model must fit the search task and responds to the very various and dynamic user's needs of information [65]. Likewise, a categorization of context types helps application designers uncover the pieces of context that will most likely be useful in their applications [64]. Indeed, according to Mcheick [65], in order to model the context of an application, first of all, one has to look for different elements that affect the application. So, before processing context, we must have that kind of information [65].

Context modelling techniques provide a crucial support to the delivery of the right information at the right moment. Moreover, it allows adaption, personalization, and also anticipation of the results to be returned by the Information Retrieval System (IRS) [68]. Effectively, context modelling is a step towards decoupling context management tasks from their application. This process involves several open research issues. To this aim, while modelling and designing the context, we should - regardless of the model - take into account some requirements.

In this section, we began by presenting a synthetic overview of the notion of context and its significance in the IR process as well as the

motivations and the issues surrounding IR activities. Then, we highlight the modelling requirements, to the purpose of finding correlations with the issues overviewed in the previous section. After that, an evaluation the importance of various context criteria and factors and their correlations is presented. Indeed, we performed a Friedman test evaluation together with a Kendall's W normalization upon a data sample from a survey about the search habits of 434 anonymous internet users [3]. The obtained results support the overall idea that, given the technological advance, a standard contextual model is today conceivable.

5.2. Context significance in Information Retrieval

According to Kehinde et al [67], context refers to the circumstances in which an event (an IR computing task in our case) takes place. In fact, context is multi-layered; it extends beyond users or systems. It is not self-revealing, nor it is self-evident, but searchers do integrate context, which, they understand intuitively, in IR theory and practice [17]. In addition, IR task's context is any information whose change modifies the task's outcome [17]. Thus, an application is believed to be "context-sensitive" or "context-aware" if its structure and behaviour change depending on the context so as to provide relevant information and services for a given user. Research activities on context-aware IR have increased remarkably in recent years and many approaches have been developed to automatically provide users with information and services based on their current situation [19]. But unfortunately, they remain greatly dependent on the field of application (smart-spaces, weather-forecast, tour guides…). In fact, there are no standards.

Context-aware computing was introduced for the first time by Schilit, Adams, and Want who state: "*One challenge of mobile distributed computing is to exploit the changing environment with a new class of applications that are aware of the context in which they are run*". After that, there have been many definitions about the notion of context in IR. One of the most approved definitions is the one given by Dey (2001): "*Context is any information that can be used to characterize the situation*

of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves. And by extension, the environment, the application and the user are embedded in". In short, we can say that Context includes all the intrinsic and extrinsic factors, which are related to a given search task and whose the direct or indirect inclusion in the IR process leads to enhance, whether implicitly or explicitly its effectiveness to convey the right information to the searcher.

Throughout years and with the advance of technology, search task became more flexible, allowing a wider range of choices between different sources of information, devices, and search categories. Moreover, the perspective of an eventual collaboration became possible, regardless of the location of the different searchers.

Motivations behind the ascent of context in IR can be grouped as follows [3, 16, 68 – 69]:

- *User (searcher) aspects*: people need help around their activities. Thus, context may be used in: personalizing and customizing services and information to the user, executing automatically some services for a user, tagging some Information to support latter retrieval, and enhancing the efficiency of IR.

- *Environmental aspects*: The search can either be self-initiated or external. In addition, the user's goal may not be specific enough and can be changed several times during the search process. Thus, fuzziness and variability lead to a need of adaption especially in terms of interaction between the user and the systems which are not well defined factors.

- *Technology*: The large amount of data leads to the rise of new applications: *user's preferences learning*, *context computing*, and *social-networking services*. Likewise, high technology improvements have occurred: tactile, 3G (4G, 5G…) connections, GPS... Especially, the generalization of the use of mobile phones, and the

emergence of ultra-books, tablets, and smartphones… which open up a new world whither user can interact with more people in a greater number of locations.

5.3. Issues of Information Retrieval

Besides the great benefit from the use of context, this latter can have many counterparts. More precisely, it is not the inclusion itself which generates problems, but the bad exploitation of the contextual features in the global IRS whether before, during, or after search. Here after, we synthesize the features of Information Retrieval tasks and the issues they might cause (see figure 5.1).

- *Proactivity*: Nowadays, technology allows us to be simultaneously active in a multiplicity of spaces. For example: reading a book or watching a movie, while receiving an SMS or sending it [70]. This would lead to disruption and distraction. We talk about the problem of activity spaces' mixing (i.e. several directions at once), which is hardly manageable. In fact, the goal of the user may not be specific enough and due to those distractions, it can be changed several times during a search session [68]. Moreover, the locality where the search of information is focused may continuously change due to the portability of mobile devices. Thus, users' interests may also change as their location changes [19].

- *Empowerment*: Different search results are relevant to different persons; a first solution was to empower the searcher [71]. Thus, users were involved to express constraints or preferences in an intuitive manner resulting in the desired information to be returned among the first results [19]. Consequently, they became overwhelmed. Indeed, in old practices, the users were the masters

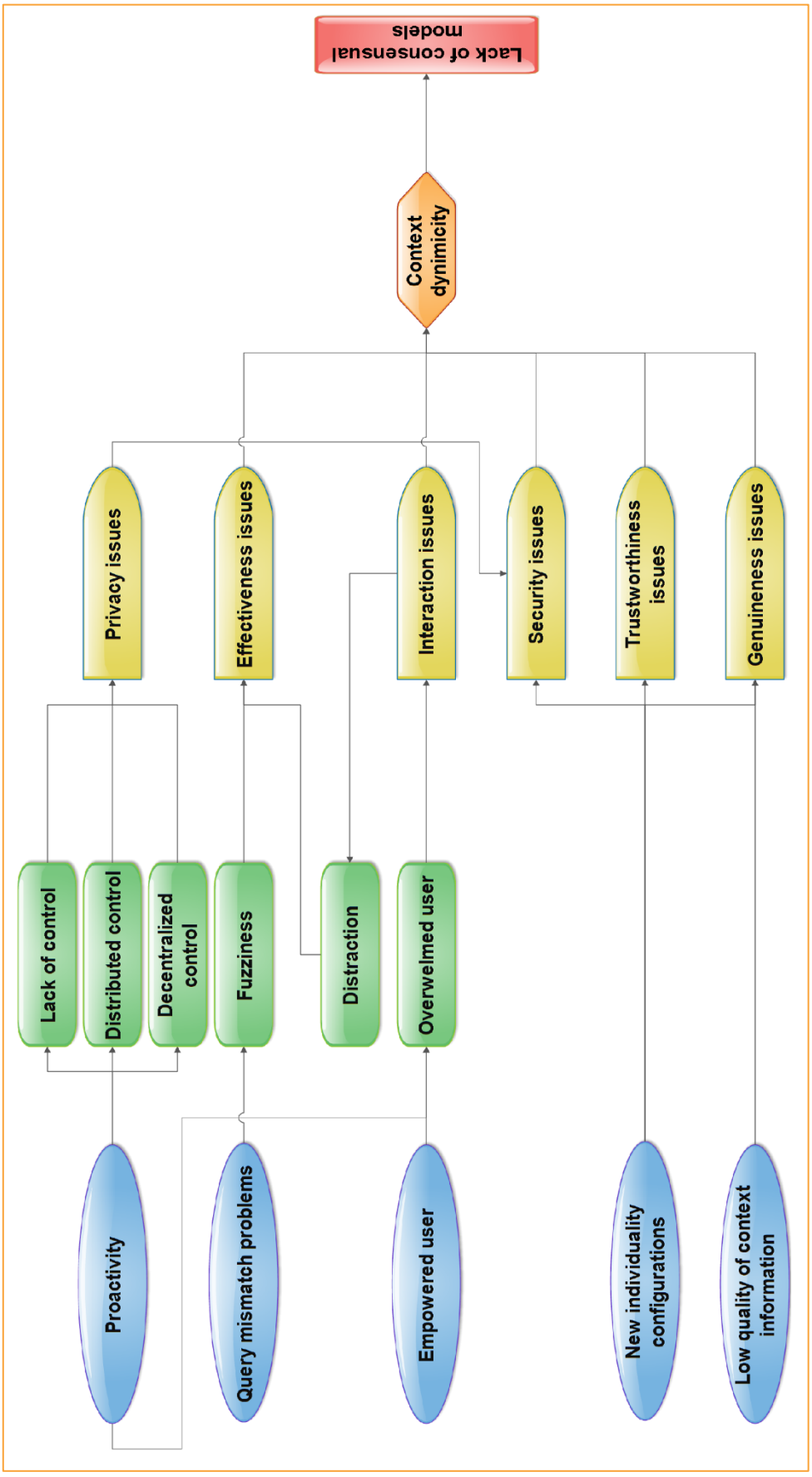


Figure 5-1: The issues surrounding Information Retrieval.

of applications' reactions. They interacted with mouse, keyboard… etc. Nowadays, users have a higher degree of dynamicity (smoother experience), but paradoxically they lose control as the flexibility increases. According to Kapor [72], users have no idea about when, what, why, and from whom they get the information and to whom they send it. In fact, the Internet allowed them to have decentralized and distributed control instead [73]. As an outcome, privacy theft dangers occurred in this new era of IR, where everyone is over-connected.

- *New individuality configurations:* Sometimes virtual partners become more important than the physical persons beside us [69]. Indeed, first, there were friends and family cycles… now the sphere is being globalized; especially because of social media that offers the possibility to interact publicly. New excitements about self-expressing and self-publishing occurred [73] (e.g. social networks, blogs, forums…). Public has become more active and more participative in new media and the power of media shifted to the power of people. Since Internet cultivates new configurations of individuality [73], internet users are turning to world citizen with a meaningful role to play. Then, security issues might result if those roles stay unmanageable.

  In fact, the results we obtained in our survey show that the limit between 'Personal preferences' and 'Social network preferences' is shrinking. That is to say people do take into account the view of their (physical and virtual) social network proportionally to their own *Personal preferences*. They are indeed influenced by their friends, collaborators, as well as by their social network. This is why the opinion of these latter is as important as their own; yet most people do prefer performing their research alone, which is paradoxical.

- *Query mismatch problems:* Our study shows, analogically to the study of Broder [37], that people do perform informational (thematic) search more than navigational one (fuzzy, unknown, or poorly

defined needs). Effectively, the demands of everyday life like establishing contacts, shopping, traveling, entertainment, and news consumption are generally well covered in the Internet. But when it comes to thematic queries, the user will feel like navigating without compass [70]. Moreover, mobile users utilize limited number of keywords per query, which causes query mismatch problems [74]. Undeniably, the fewer keywords, the searcher uses, the harder it is for the IRS to please their need of information. Contrariwise, our survey's results resemble barely to the study conducted by Kamvar and Baluja (2006). Indeed, the two sample results (i.e. Smartphone and non-smartphone users) were nearly similar and this is due to the technological advances that made smartphones as powerful as some laptops nowadays.

- *Low quality of context information:* Context is nowadays used whether implicitly or explicitly in most search engines. Thus, IR can also have issues with genuineness. In fact, low quality context information can be a consequence of sensors' technical limitations and context reasoning algorithms or privacy policies of the entities which benefit from the contextual information [75].

- Besides, *context data are imperfect:* Incorrect; if they fail to reflect the true state of the world they model, Inconsistent; if they contain contradictory information, and Incomplete; if some aspects of the context are unknown. As a result, decisions are based on erroneous context data, which can generate genuineness issues. This may increase the cost of reasoning since the context is uncertain or does not represent accurately the reality. Thus, quality of context models has been proposed to quantify this inaccuracy [75].

5.3.1. Discussion

The context is dynamic and moving. This is why a focus on context management aspects is required so that context can be handled in real time. Besides, although context aware devices and applications offer more customized services and provide a richer experience, there are no known

standard models that fit a large scale of devices, neither theoretical basis, nor rigorous definition of its usability and usefulness [76-78]. In short, there is a lack of consensual models.

Figure 5.1 recapitulates the afore-mentioned issues found in IR and their correlations. For instance, we think that the proactivity of the user can cause her lack of control or the fact that the user may have a distributed or decentralized control, which will generate in a higher level some privacy issues. In addition, security issues and flexibility evolve disproportionately. Thus, an empowered user is an overwhelmed user who may have some interaction issues. Furthermore, the distraction of the user may substantiate the fuzziness of her queries and by transition, the effectiveness of the obtained results. Besides, new individuality configuration and the low quality of context information can lead to security, trustworthiness, and genuineness issues.

5.4. Context's components

Further to researches in the field of CIR, we can observe that each search task is unique and comes under a certain configuration of contextual factors. However, some correlations can be found among a set of search activities of the same user, between two similar users, or between two disjoint users performing a search task in a similar configuration of contextual factors. According to Jilei (2010), context fully describes the searcher, her device, and her surroundings using a wide range of sensed and historic information which forms the backbone for a completely new class of services. There is a real need for categorizing context's types or components in order to spot the most useful ones according to a given application. Effectively, nowadays, context is more targeted than ever.

As Han, Wang, M., Wang, J. [16], we agree that task is the driving force that constitutes IR and real information behaviour. In order to find if there may be other contextual components, we choose sixteen valuable works that made use of context for different purposes. Our goal was to deepen our comprehension of the notion of context according to different

use cases and to come out with a categorization of the context factors. What can be noticed is that the use of contextual factors differs from one application to another. Thus, the related works are given just as valuable resources to enrich future researchers with leading theories, models, and results in the area of Contextual IR (CIR).

We find that the IR task is usually interlaced with seven contextual components (*Table 5.1*), namely: user, queries, device, time, location, environment, and documents. We restricted our focus to those seven contextual factors and to test their coverage, we conducted a short survey about search habits.

Table 5-1: Most Important contextual factors in an IR task.

Components	Example	Sources	Related works
Search task	Personal calendars can be used to discover user's current task	Forms, events in the calendars, query logs, feedback	[17, 24, 78 - 83]
User	Sana usually browses technology news when waiting the subway in working days morning.	Profiling, user mining, forms and feedbacks, search logs, personal data and content, contact list, social network.	[17-18, 24, 65, 78 - 82, 84 -87]
Queries	-	-	[18, 65, 81, 86]
Device	The doctor uses her tablet in a hospital to search about the suitable diagnosis.	Composite Capabilities/ Preference Profile (CC/PP) proposes an infrastructure to describe device capabilities and user preferences. Used for content presentation [63].	[24, 78, 82, 85 – 86, 88]
Time	According to a time where a user search for a restaurant we can deduce the type of food he is searching for "break-fast", "lunch", etc.	System clock, calendars,	[65, 78-79, 82, 85-86, 89]
Location	City guides, weather forecasting, products and services marketing	We can use infrared, Bluetooth and WIFI signal strength to determine indoor locations and GPS for outdoor locations.	[65, 78-79, 82, 85 – 86,88-89]
Environment	Find all the participants for a meeting saved as an event in the calendar.	Environment sensors, device pervasiveness (Bluetooth, accelerometers…)	[17, 65, 78-79, 82, 85 - 89]
Documents	-	Web, intranet, or personal texts, images, videos, etc.	[17, 24, 81]

5.5. Context modelling

Pasi [90] remarked that, in recent years, a great deal of research has addressed the problem of personalizing search, to the aim of taking into consideration the user context in the process of assessing relevance to user's queries. Context-awareness is one of the drivers of the ubiquitous computing paradigm, whereas a well-designed model is a key accessor to the context in any context-aware system [91]; independently from the field of application, yet firmly dependent on the application itself. In fact, a variety of context models have been proposed to properly handle the key aspects of the context, while focusing on scenario-based acquisition, management, and representation of context [92]. Whereas the majority of works and research in this field provide context models that make use of context features in a particular application, the challenge of the community these last years has been to come out with a prospective standardization of context models.

5.5.1. Definition

According to Go & Sohn [91], the meaning of modelling context is to make context interpretation knowledge. Indeed, according to Mcheick [96], in order to model the context of an application, first of all, one has to look for the different elements that affect the application. So, before processing context, we must have that kind of information. This point will be tackled in the 4th section.

Furthermore, to address the issues surrounding the IR task, there is a need for context models that foster context reuse and support the ease of retrieving the right kind of information by providing appropriate abstractions of contextual information [94].

The typical approach considers a number of special requirements and conditions. So, in this section, we tackle those requirements and see if correlations can be found regarding the aforementioned IR issues, but before, we began by defining the notion of context modelling.

According to Mcheick [96], context-awareness is no longer limited to desktop, web, or mobile applications. In other terms, context management has become an essential functionality in software systems [38]. A data life cycle shows how data moves from phase to phase in software systems like applications or middleware, i.e. it explains where data are generated and where they are consumed. An appropriate context lifecycle consists of four phases, namely: Context Acquisition, context Modelling, Context Reasoning and Context Dissemination. In the remainder of this sub-section, we will focus more on the Modelling phase. For more detailed information, reader may refer to the papers [63 and 95] where a good definition about context architectures is given; tackling the sensed sources, context-acquisition, pre-processing, storage management, distribution, representation, then fusion and reasoning.

Content is usually delivered together with contextual information to users as well as the context does surround the request for information initiated by this same user. Content is the main information whereas, context is used to improve the quality of service and user's experience [76]. In this regard, Abowd et al. [80] state that, context modelling techniques are cornerstones in the delivery of the right information at the right moment; providing a crucial support to enable effective reasoning, adaption, personalization, and also anticipation of the results.

A context model formally describes and expresses informative knowledge about the relevant aspects of the real world that are used for an application [96 - 97]. It abstracts from the technical details of context sensing and allows coupling the real world to the technical view of context adaptive applications. Therefore, context models play an important role for building applications that can react on real world events and one of the challenges associated to this research is to construct a model that can be used for different context-aware systems [96]. Thus, according to Ryu et al. [86], in order to fully benefit from the context, we have to follow a process (*Figure 5.2*).

Figure 5-2: Context integration steps.

As a matter of fact, context modelling allows independency between the application and its context. Effectively, contextual information space is characterized by the state of the different elements that constitute it (i.e. the set of the observations performed in a given time). Lombardi [60] gave examples:

- Energy can be considered as context in the research area of smart energy.
- Occupancy, weather, time and location play an important role in smart heating.
- Physical activity recognition which is important in context recognition can for example be achieved through smart glasses.

Research in context modelling is not new. Likewise, in recent years, six leading context models have been introduced; namely: *Key-value models*, *Mark-up Scheme Models*, *Graphical models*, *Object Oriented models*, *Logic based models*, and *Ontology based models*. In addition, a possible hybridization can be considered in certain cases. The detailed study of those models is out of the scope of this chapter, however, valuable information can be found in [91, 95, and 98]. Moreover, in [97 and 99] an interesting overview of context representation types and the different usages of context models during the operation of a context-aware application is given.

5.5.2. Modelling requirements

According to Bhargava, Krishnamoorthy, & Agrawala [100], an ideal context model is one which serves efficiently in any domain and will be abstract enough to manage all the dimensions of context such as location, time, and user profile. It will be versatile enough to have a rich set of representation features such as flexibility, context granularity and constraints. It will also be advanced enough to incorporate a variety of context usage functionalities. Thus, a context-aware system, that incorporates the most useful of these features and characteristics aforementioned, will focus on the context problem as a whole, and will be abstract and generic enough to be applicable in any domain or environment.

Context modelling is a step towards decoupling context management tasks from their application. This process involves several open research issues. Likewise, while modelling and designing the context, we should -regardless of the model- take into account some requirements. A review of some related work in the literature [66, 90-91, 95-99,101] reveals over 50 different requirements. *Table 5.2* summarizes these requirements; grouped in a categorization adapted from [99 - 102].

According to Bolchini et al. [102], defining the requirements covers the focus of the model, its representation and the way context data are used; the result is a rich set of features, emphasizing that context modelling is a complex problem. Depending on the specific purpose it is designed for, each model may include several of the listed features.

Table 5-2: Context models' requirements.

Categories	*Description*	*Features*
Information capture	Context information has to be used as explicit query to the community information system. A context should basically be recognized automatically; however, the system should allow users to explicitly provide context information at the same time.	1. Context detection, 2. Context Inference, 3. Context construction

Representation features	Explicit representation concerns previous knowledge about the environment. Thus, the system has to consider all partially matching contexts and merge them into a coherent presentation of the information. Moreover, it may be important that additional services and requirements can be integrated in the model at run-time. Moreover, people who are not the initial designers carry out the final design and the maintenance of context-aware systems, usually. Thus, the adaption to specific domains should be easy and concise.	4. Representation Standards, 5. Uniform Context Representation, 6. Context dimensions, 7. Structuration of the information space (flat, tree, graph), 8. Relationships and dependencies, 9. Compatibility and usability of modelling formalisms, 10. Context Fusion, 11. Evolutionary development and flexibility, 12. Balance simplicity and ease of use, 13. Genericity (domain independent), 14. Reusability and extensibility, 15. Consistency – no contradictions, 16. Readability and understandability (intuitive relations and terms), 17. Richness and detail, 18. Distribution of the model, 19. Usability and Feasibility of context exploitation in the final application, 20. Interoperability: It should enable syntactic and semantic interoperability between different applications and services, 21. Completeness, redundancy: it should cover the whole domain, but do not redefine explicit/implicit knowledge necessarily, 22. Variable context granularity: the ability of the model to represent the characteristics of the context at different levels of detail, 23. Valid context constraints: the possibility to reduce the number of admissible contexts by imposing semantic constraints that the contexts must satisfy for a given target application, 24. Multi-Context Modelling: the possibility to represent in a single instance of the model all the possible contexts of the

		target application, as opposite to a model where each instance represents a context.
Reasoning features	A context model should have the ability of inferring good/ bad behaviours that have to be adapted/ avoided based on background knowledge of the current state. Likewise, in case the system perceives ambiguous, incoherent or incomplete context information, it should be able to interpolate and mediate somehow the context information and construct a reasonable current context. Furthermore, both physical world and our measurements of it are prone to uncertainty. Hence, one of the key requirements of context-awareness is capturing and making sense of imprecise, and sometimes conflicting data, while, being aware about the limits to user's trust and not to cross them.	25. Richness and quality of information, 26. Heterogeneity and mobility, 27. Applicability, 28. Comparability, 29. Activity Recognition, 30. Goal Recognition, 31. Expressiveness and Reasoning, 32. The selection of appropriate level of automation, 33. Contextual ambiguity and incompleteness management, 34. Avoidance of unnecessary interruptions as well as information overflow, 35. Partial Validation: Context information and contextual interrelationships are complex. Development of validation mechanisms is particularly desirable, 36. Inference: Most of context information is not directly acquired; the gathered information (low-level context) may be processed to obtain high-level context information by composition, abstraction or inference techniques, 37. Satisfiability (constraint modelling): restrictions and constraints on acceptable values.
Context management and usage	The context model should support inference of higher level context from low level sensed context. Moreover, it should allow applications to behave differently in different contextual situation.	38. Context Caching and Update Scheme, 39. Maintenance and evolution of the context model; 40. Selection of the appropriate visibility level of system status; 41. Context adaptation, the ability to implement or modify services by automatic context changes, 42. Context scalability, the ability to obtain new information from the context through existing information and use resources related to the current context.

Other features	The modelling effort for designing and maintaining context models should clearly pay off in terms of improved access to information and increased working efficiency. Moreover, one of the goals of a context modelling approach is to give context-related relevant information to the user while he or she is in that context. This means, that the recognition of the current user's context and the retrieval of information relevant to that context has to be done in reasonable time.	43. Timeliness, 44. Traceability, 45. History logging, 46. Insurance of user control (the user must feel in charge of the situation), 47. Definition of a security level to ensure user privacy.

Moreover, Bettini et al. [98] noticed that the new approaches of context modelling and reasoning address many of the requirements found in the literature; however, none of them fulfils all the requirements for a generic context information modelling and reasoning approach.

In addition, as long as the integration of a contextual dimension and the concept of context awareness remains independent from the business side of the application, we can find correlations with other fields related to IR (like Cloud computing, Big-data, etc.).

Furthermore, we have remarked that all the aforementioned requirements are related to the issues we previously outlined. Therefore, it is of most importance to analyse deeply those requirements in order to find the most suitable way to overcome the issues.

5.5.3. Discussion

Najar et al. [99] remarked that the observed context elements (i.e. relevant information) as well as their use differ from a system to another, and consequently from a model to another, and it is often difficult to evaluate them.

In fact, there are various issues and open research challenges that need to be addressed. In this section, some of the challenges have been highlighted for the purpose of achieving the correct implementation of

context-aware systems and we observed that the cited challenges do not only match the requirements and issues of context modeling, but also those of the information retrieval task.

As the authors Khattak et al. [95], we agree that before proceeding to the reasoning phase, context aware components and their related information have to be fused and merged, but how? In which extent? And on what basis? In the remaining of this chapter, we will try to solve these questions; focusin on the context modeling requirements.

5.6. Evaluation

According to Pasi [90], evaluation is a quite important issue that deserves special attention, and which still needs important efforts to be applied to context-based IR applications. To evaluate a model means to assess its quality properties, such as accuracy... Effectively, the quality criteria of a context model are [98 – 108]:

- *Accuracy*: how exactly the provided context data mirrors the reality;
- *Precision*: how detailed a measurement is stated;
- Probability of correctness: probability that a piece of context data is correct;
- *Trust-worthiness*: how likely it is that the provided data is correct;
- *Resolution*: granularity of information;
- *Up-to-dateness/freshness*: age of context information.

5.7. Survey about nowadays search habits

In order to understand the trends and users' intents in IR and come out with significant patterns for our upcoming research in CIR, we conducted a short survey among 434 anonymous online users (mostly Facebook and Linkedin users).

5.7.1. Sample Data

Based on the afore-mentioned influential context factors that can be found in the literature, ten leading questions have been formulated and formatted. Then, we broadcasted the Google Form link through some social network groups and also provided a printed version to students (about 12% of the participants). The participants were from 27 nationalities which contributed to enrich the study, but unfortunately, since the study was carried out online many socio-demographic categories have been excluded. Furthermore, despite the fact that many similar surveys have been conducted already, our main focus was to understand the habits and preferences behind actual daily search tasks knowing that several technological advances occurred this last decade. We took special care to formulate the study in the simplest possible form in order to provide researchers in the field of CIR with a clear view about contemporary search preoccupations.

We note that context information is input when delivering a service. This information can be segregated into categories. A categorization of context types helps application designers uncover the pieces of context that will most likely be useful in their applications.

Likewise, the survey motivated the respondents for information surrounding seven context dimensions found in the literature namely: *search task*, *user, queries*, *device*, *time*, *location*, *environment*, *documents*. Within this context, the six questions mentioned bellow, were formulated in the simplest possible form:

- While searching the internet, what do you use (source of information)?

  Famous search engines, Social Networks, Forums, Mobile apps, Other (specify)…

- While searching the internet, what do you use (device)?

  Desktop, Laptop, Tablet, Smartphone, Mobile phone.

- What are your favourite search categories?

  Local services, Technology, Travels, Entertainment, Society & communication, Sport, Health & food, Games & hobbies, News & events, Science, Industry, Other (specify)…

- How many keywords do you usually use?

  1 – 3, 4 – 6, 6+.

- What are the most influent factors in a search activity?

  Accuracy, Location, Time, Personal preferences, Social network preferences, Results & content personalization, Other (specify)…

- How do you prefer performing a search activity?

  Alone, Over social networks, With real friends or relatives, Other (specify)…

Furthermore, users (see *Table 5.*3) were invited to provide background information about their gender, age, activity, and whether they own a Smartphone or not. These information allowed us to deepen the analysis. Hereafter, we will focus on the fifth question since the answers may be considered as being quality criteria in CIR. Indeed, we believe that defining the quality criteria of a context model, may help to merge the different context items (i.e. elements) wisely; by developing a formula of prioritization of those elements in order to increase the degree of precision and reach the desired grade of relevance.

Table 5-3: Socio demographic categories of the respondents' sample.

Gender		Education	19,6%
Female	52,8%	Research	30,4%
Male	47,2%	Industry	11,1%
Age		Commerce	3,2%
Under 18	0,3%	Unemployed	3,9%
18 – 29	44,2%	Retired	0,9%
30 – 49	32,7%	Other	8,1%
50+	22,8%	**Smartphone owner**	
Activity		Yes	71,9%
Student	22,8%	No	28,1%

In fact, we found that the most important context factors that prompt information retrieval are; beginning by the most important: *accuracy*, *freshness* (time), *location*, *personal preferences*, *social-network preferences*, but also *trustworthiness* of the context' sources, *results ranking, presentation of the information*, and *display speed* according to respondents' suggestions. Besides, it is important to note that depending on the current situation and goals, only a few of a very large number of context items may be relevant. This defines the relevant context. Thus, the relevant context is a subset of the overall context, and is likely to change as the situation changes and even as additional information becomes available [100].

5.7.2. Results and discussion

In the following, we will try to analyse the results. But before, it is important to mention that for a deep understanding of them, we performed a cross tabulation analysis, which shows -mostly- very harmonious results regardless to the different types of demographic categories.

5.7.2.1. Favourite source of information

About 62.2% of the searchers concede preferring 'Famous search engines' to perform their search activities whereas, 19.7% choose 'Social networks' and 10.9% 'Forums'. Besides, a minority of searchers 5.1% and 2.1% admit using, respectively, 'Mobile applications' and 'other sources of information' like dedicated web portals, less known and more targeted search engines, digital libraries, internal society or university databases, library catalogues, faceted search engines, personal content (documents, emails, and bookmarked web pages), and finally, computational knowledge engines such as wolfram. Unfortunately, these results (*Figure*

5.3) gave rise to our apprehension about the preference of researchers towards famous search engines, which, are agreed to provide powerful search results for trivial queries. But, they lose out personalization and customization of the results according to internet surfers' needs and purpose. And consequently, they miss effectiveness if the needs are unknown, dealing with thematic search activities for example.

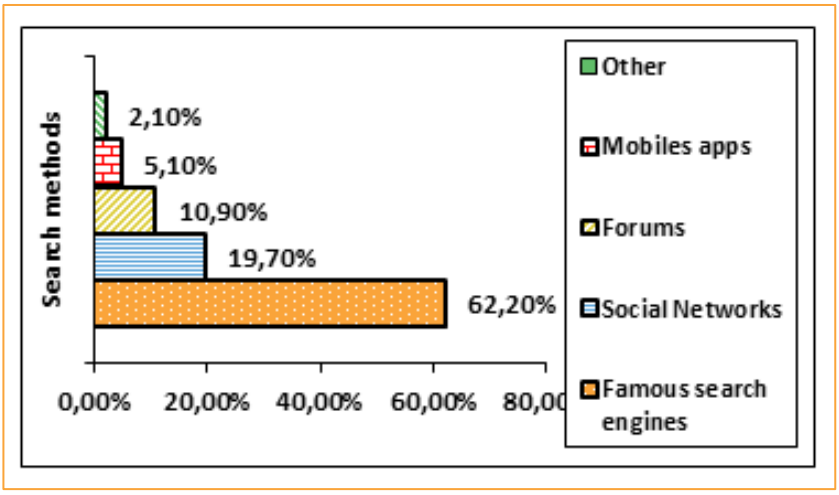


Figure 5-3: Search methods statistics.

5.7.2.2. Favourite devices used for search activities

Results show (*Figure 5.4*) that 39% of searchers use mostly 'Laptops' while searching, whereas 23.1% still prefer 'Desktops', 21% 'Smartphone', 15% 'Tablets', and only 1.9% of searchers use their 'Mobile phones' in daily search tasks.

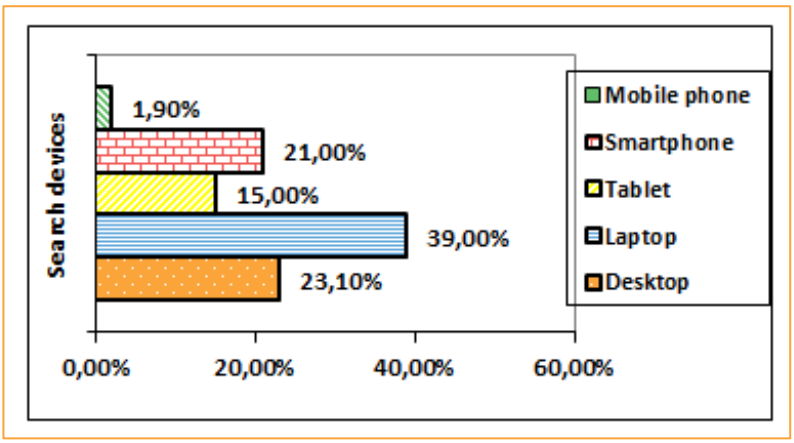


Figure 5-4: Search devices statistics.

It means that despite the spread of mobile technologies, people still make use of desktops when it comes to perform their daily search

activities. Moreover, we can notice that among all the mobile devices, laptops are the favourite, and it is quite understandable because of their ease of use in terms of interaction fluency, query typing, and clarity of results presentation.

5.7.2.3. Favourite search categories

Concerning favourite search categories, unlike the study of Kamvar and Baluja [83], we found that 'Technology' outclasses the other categories with 19.69%, nearly followed by 'News and events' with 16.41%, and 'Science' 14.86%. The remaining proposed categories obtained the scores showed in *Table 5.4*, beginning with the highest.

Table 5-4: Results of search categories.

Categories	Responses	Categories	Responses
Technology	19.69%	Society & communication	6.11%
News & events	16,41%	Local services	5.56%
Science	14,86%	Sport	4.83%
Entertainment	8.75%	Games & hobbies	4.10%
Health & food	8.57%	Industry	3.01%
Travels	6.65%	Others	1.46%

Despite the differences between the mentioned search categories, we wanted to find some patterns concerning the types of needs behind the queries. The survey results show, analogically to the study of Broder (2002), that the respondents were most willing to perform informational (thematic) search than navigational one.

5.7.2.4. Number of keywords per query

The 44.7% of respondents admitted using from one to three keywords, 44.5% from four to six, and 10.8% more than six keywords. Undeniably, the less keyword, the searcher uses, the harder it is for the IRS, to please their need of information. For instance, we have noticed that 45.51% of Smartphone users utilize from one to three keywords, whereas 43.26% use from four to six. Contrariwise, this trend was reversed for respondents without Smartphones with, respectively, 42.62% and 47.54% as shown in *Figure 5.5*.

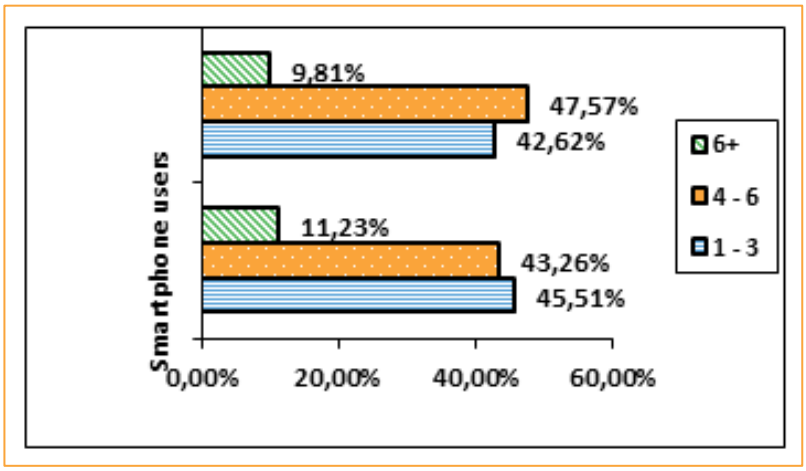


Figure 5-5: Smartphone and non-Smartphone users results.

These results resemble barely to the study conducted by Kamvar and Baluja [83], who reported that mobile users' queries are shorter and therefore more ambiguous. Indeed, we remark that the two sample results (i.e. smartphone and non-smartphone users) are nearly similar and this is due to the technological advance concerning smartphones that are nowadays as powerful as some laptops. Nevertheless, the results obtained in this section about keywords, indicates the need to rely on the context factors surrounding the search activity.

5.7.2.5. Most important contextual factors

We noticed that the most important contextual factors (*Figure 5.6*) are: '*Accuracy*' with 38.29%, then '*Time*' (freshness of the information) with 23.9%, followed by '*Results and content personalization*' with 12.13%, '*Personal preferences*' with 11.3%, '*Location*' with 11.18%, and '*Social network preferences*' with 2.85%. Finally, 0.36% of respondents chose the option '*Other*', and gave some suggestions. We retain trustworthiness and genuineness of the information sources, the results' ranking and referencing, and website speed. This question was somehow the core of our study, since our main focus was about the importance of extrinsic and intrinsic contextual factors in any search activity.

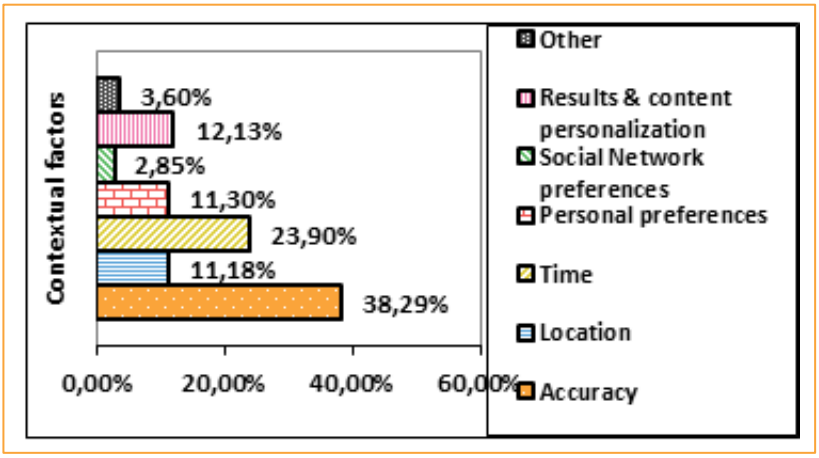


Figure 5-6: Statistics about the most important contextual factors.

The two most interesting outcomes are:

- The accuracy and freshness of the information are more important than their relation to the notion of location. This differs from the perspective of Ryu et al [86], who classified contextual factors that prompt information needs as follows-beginning with the most influent: location, time, conversation, and activity, and also Kamvar and Baluja [83] who classified them as follow: activity, location, time, and conversation. Instead, this confirms the trend concerning the interaction at a very large scale (allowed by social networks mostly), where, everyone is a world citizen without known boundaries, nor territorial limitations of knowledge.

- The limit between 'Personal preferences' and 'Social network preferences' is small. That is to say people do take into account the view of their (physical and virtual) social network proportionally to their own 'Personal preferences'. According to Evans and Chi (2008), external environment (i.e. people) may be valuable information resources for one's information search process. In their paper, Evans and Chi (2008) state that recently, searchers have observed direct user cooperation during web-based information seeking. Active collaboration may occur under some circumstances, where users interact together remotely, asynchronously, and even involuntarily and implicitly. They are indeed, influenced by their friends, collaborators, as well as by their social network. This is why

the opinion of this latter is as important as their own yet most people do prefer performing their research alone as found in the question concerning the collaboration in research.

5.7.2.6. Collaboration in search activity

84.33% of respondents concede that they rather perform a search activity alone. Whereas, 11.06% prefer being surrounded by real (physical) friends, and 4.15% choose to rely on their social network circles. Moreover, 0.46% of respondents gave suggestions that support overall that most searches are performed independently, but at times can be conducted collaboratively. This does depend on the need. These results (*Figure 5.7*) support that effectively, the IR task can either be external or self-initiated.

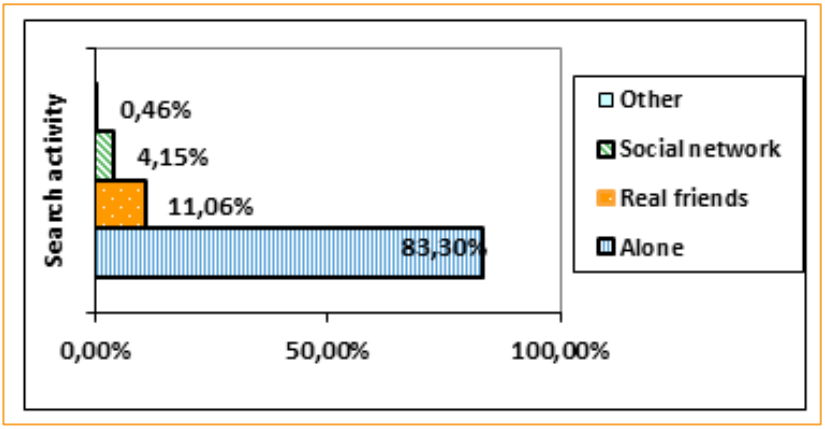


Figure 5-7: Search activity statistics.

5.7.2.7. Discussion

Throughout years and with the advance of technology, search task became more flexible, allowing a wider range of choices between different sources of information, devices, and search categories. Moreover, the perspective of an eventual collaboration became possible, regardless of the location of the different searchers. In this chapter, the significance of the inclusion of a contextual dimension was discussed. Moreover, we inquired about actual search trends taking into account the technological advances. Thus, we introduced our short survey with its detailed results and analysis, which we expect will provide future researchers with valuable information. We retain the inclination of users towards: (a) social

network preferences proportionally to their own personal preferences, also (b) users concern about accuracy and time, and finally (c) shorter and thus more ambiguous queries. Consequently, our upcoming work will consist on the formalization and testing of a CIR model centred on the IR task.

In this section, we put forward, the correlation between the different demographic categories outlined in the survey regarding "*Accuracy*" and "*Time*" as well as other context criteria. To reach this goal, we opted for a *Friedman* test evaluation together with *Kendall's W* (Kendall's coefficient of concordance) which is a normalization of the Friedman statistic.

5.8. Case study

Developed by the U.S. economist Milton Friedman, the Friedman test is a non-parametric alternative to ANOVA with repeated measures that can be performed on ordinal (ranked) data. In other words, the Friedman test is used for one-way repeated measures analysis of variance by ranks. No normality assumption is required. It is used to detect differences in treatments across multiple test attempts. The procedure involves ranking each row (or block) together, then considering the values of ranks by columns. For more details, see Corder and Foreman's paper [109].

Kendall's coefficient of concordance *(W)* is a measure of the agreement among several K judges (or subjects) who are assessing a given set of N objects (treatments) [110]. Depending on the application field, the "judges" can be variables, characters, and so on. Kendall's *W* ranges from 0 or 0% (no agreement) to 1 or 100% (complete agreement).

There is a close relationship between Friedman's two-way analysis of variance without replication by ranks and Kendall's coefficient of concordance. They address hypotheses concerning the same data table and they use the same $\chi 2$ statistic for testing. They differ only in the formulation of their respective null hypothesis. Considering a sample data as a table, in Friedman's test, the null hypothesis *(H0)* is that there is no

real difference among the N objects, which are the rows of the data table. Under *H0*, they should have received random ranks from the various judges, so that their sums of ranks should be approximately equal. Kendall's test focuses on the *K* judges instead.

- *Friedman's H0*: The *n* objects are drawn from the same statistical population (there is no difference between the treatments).

- *Kendall's H0*: The *k* judges produced independent rankings of the objects (there is no correlations between the subjects).

For our evaluation, we use a subset of the survey response data. Thus, we focus on the question concerning context factors and criteria to the aim to deepen the analysis considering the different background information (*gender, age, activity, possession of smartphone*). In this regard, our case study resembles to one of the classic Friedman's examples of use: "*n* welders each use *K* welding torches, and the ensuing welds were rated on quality. Do any of the torches produce consistently better or worse welds?" Consequently, we consider *N* categories (subjects, lines…); each judges the most important context criteria among *K* different factors (treatments, columns…). Which are the most important context factors (Friedman test)? Is there a concordance (i.e. a dependence) between the rankings produced by the different categories (Kendall's *W*)?

Computations were made by an open source tool from "Anastats"[25]. The tool allows to:

- State the significance level α (in our case $a = 5\%$)
- Calculate the degree of freedom

$$nu = K - 1 \quad (10)$$

[25]http://www.anastats.fr/index.htm the tool can be downloaded here: http://www.anastats.fr/stats/Telechargement.htm#friedman

- Calculate the critical value $q(nu,a)$ (using the *ch²* distribution table[26])
- State the test statistic (i.e. decision rule) as follows:

- If $x^2 > q$, we reject Friedman's H0 hypothesis (i.e. there is a coherence and an agreement among the categories or judges). Where ***X²*** is computed using the formula:

$$x_r^2 = \frac{12}{NK(K+1)} \sum_{j=1}^{K} R_j^2 - 3N(K+1) \quad (11)$$

- Where *k* is the number of groups (treatments), *n* is the number of subjects, R_j is the sum of the ranks for the j^{th} group.

- Calculate the *Kendall W* coefficient of concordance

$$W = \frac{Chi^2}{(N(K-1))} \quad (12)$$

5.8.1. Results and discussion

In this section, we present the obtained results. As a reminder, we used the Friedman test and the Kendall's W normalization of it in order to find if there are correlations between the perception and the assessment of context criteria by different demographic categories. Our aim was to find out if a possible standardization can be conceivable.

So, the *Tables 5.5, 5.6, 5.7,* and 5.8 represent, respectively, the evaluation's data sample and results according to *Gender*, *Age*, *Activity*, and *Possession of smartphone*.

[26] http://sites.stat.psu.edu/~mga/401/tables/Chi-square-table.pdf (for example if *df(nu)* = 6 and α = 5% (0.05), then the risk of error = 12,59).

Table 5-5: Evaluation according to gender.

	Accuracy	Location	Time	Personal preferences	Social network preferences	Results & content personalization	Other
Male	35,580	16,830	20,190	12,260	2,880	11,540	0,720
Female	37,910	12,200	25,490	9,590	2,610	11,760	0,400
Results	nu = 6, q = 12.59, x^2 = 11.79 (q > x^2; Friedman's H0 true), W = 98% (Kendall's H0 rejected).						

Table 5-6: Evaluation according to age.

	Accuracy	Location	Time	Personal preferences	Social network preferences	Results & content personalization	Other
Under 18	48,900	0,440	0,440	48,900	0,440	0,440	0,440
18 – 29	37,280	11,500	27,530	8,010	3,140	12,200	0,350
30 -49	40,450	11,990	22,100	10,490	1,870	12,360	0,750
50+	46,020	11,360	22,160	6,250	1,140	12,500	0,570
Results	nu = 6, q = 12.59, x^2 = 18.41 (q < x^2; Friedman's H0 rejected), W = 77% (Kendall's H0 rejected).						

Table 5-7: Evaluation according to activity.

	Accuracy	Location	Time	Personal preferences	Social network preferences	Results & content personalization	Other
Student	34,919	11,111	24,867	13,227	4,233	11,640	0,005
Education	41,667	8,929	21,429	11,310	1,786	14,286	0,595
Research	39,922	11,628	24,806	10,078	2,326	10,465	0,775
Industry	40,217	7,609	22,826	13,043	1,087	13,043	2,174
Commerce	24,989	14,993	29,987	9,996	4,998	14,993	0,045

Unemployed	31,105	15,552	22,218	11,109	6,665	13,331	0,020
Retired	42,692	0,128	14,231	14,231	0,128	28,462	0,128
Other	33,333	15,476	20,238	13,095	5,952	10,714	1,190
Results	nu = 6, q = 12.59, x^2 = 42,30 (q < x^2; Friedman's H0 rejected), W = 88% (Kendall's H0 rejected).						

Table 5-8: Evaluation according to possession of smartphone.

	Accuracy	Location	Time	Personal preferences	Social network preferences	Results & content personalization	Other
Smartphone users	40,200	11,040	23,390	11,530	2,640	10,540	0,660
Non-smartphone users	33,050	11,440	25,000	10,590	3,390	16,100	0,420
Results	nu = 6, q = 12.59, x^2 = 11.14 (q > x^2; Friedman's H0 true), W = 93% (Kendall's H0 rejected).						

Two observations can be made from the bellow tables:

- Since the Friedman's *H0* is rejected in the cases "*Activity*" and "*Age*" evaluation, there is a difference between the treatments. It means that the different categories gave different appreciations to the context criteria. This observation is reversed in the case of "*Gender*" and "*Smartphone possession*", where the Friedman's *H0* was true (i.e. the *n* objects are drawn from the same statistical population). Thus, because of this righteous divergence it is better to rely on the global survey's results in order to differentiate the appreciations of the different criteria. In other words, we can say that there is no clear correlation between the criteria as each criterion is unique, derives from different factors, and implies the consideration of different context features. Nevertheless, the fuzziness concerning the boundaries of context criteria can be overcome by inference techniques. Thereby, one modelling criteria can be abstracted,

inferred, and handled (or managed) from another one. For example: *Personal preferences*, *Social network preferences*, and *Results & content adaption* can be used to elicit information about *accuracy*, *time*, or *location*.

- However, concerning the Kendall's evaluation, the obtained results were very encouraging. Such as the Kendall's *H0* hypothesis can be rejected in all the performed tests. It means that there is a strong correlation (i.e. concordance) and harmony between the different subjects (categories). In other words, the different criteria were appreciated almost alike regardless of the categories in the different tests. Thus, both "*Men*" and "*Woman*" have, approximately, the same exigencies in terms of context criteria as well as the different "*Age*", or "*Activity*" categories do have close appreciations. Moreover, the concordance between smartphone users and non-smartphone users in *Table 5.8* supports the idea that smartphones are becoming almost as powerful as laptops or desktops. So, users do have the same concerns regardless of the device they are using.

The most interesting outcome is that, given the technological advance, a prospective standardization of context models can be conceivable if we take into account the human factor (user context dimension from which, information about the other dimensions can easily be inferred). But as there are many other context factors (six in the case of IR), each context dimension should be analysed independently in order to evaluate the feasibility of a standard model resulting from their fusion.

5.9. Conclusion

Context-aware systems can, nowadays, dynamically adapt to different user situations to provide smart services and relevant information. In general, context refers to the information that can be used to characterize a given situation and context models are employed to formalize the acquisition, reasoning, and dissemination or consumption of the contextual information surrounding

context-aware systems. However, context modelling and the inclusion of context in the global IR process still have some open research issues and challenges especially the lack of consensual models.

In this chapter, the significance of context in the field of Information Retrieval was discussed together with the issues that might occur in search activities and their correlations. Moreover, a detailed study of context modelling and more precisely context modelling requirements was introduced. Thus, a categorization of these latter was proposed aiming to draw potential solutions to the outlined IR issues.

Assuming that a context model in a context-aware system has to allow the smart fusion of context information and elements before proceeding to the reasoning phase, we evaluated the appreciations of context quality criteria according to different demographic categories using the Kendall's W coefficient of concordance. The obtained results are very encouraging, and corroborate the harmony between the judgments (appreciations) of the different demographic categories indicating that an eventual standardization of context models is possible, at least from the Human (user dimension) point of view.

After reviewing the notion of context in IR, and studying the importance of the inclusion of a context dimension in the overall IR process, we describe in the two last chapters of this thesis our proposed context-aware IRS. More precisely, we will talk about our stemming algorithm, indexing method, and query-document mapping technique.

CHAPTER 6 PROPOSITION OF A SEMANTICALLY ENRICHED

CONTEXT-AWARE STEMMING ALGORITHM

6.1. Introduction

In this chapter, we talk about our modified version of the Context-aware Stemming algorithm, itself based on the well-known Porter stemmer in an effort to maximizing the proportion of the meaningful stems and thus, the search effectiveness without compromising the other performance measures. Several stemmers are presented and a synergetic hybrid solution is proposed. Indeed, the Semantically Enriched Context-Aware Stemming algorithm combines features from algorithmic stemmers and dictionary stemmers with respect to conceptual indexing techniques in order to improve retrieval performance; proposing root words much comparable to lemma. Moreover, a new query-document mapping technique is proposed based on a previous work and the experimental results conducted with the WT2G dataset show that our algorithm is noticeably more efficient; enhancing precision (up to 300%) as well as recall (up to 700%) as compared to Porter and CAS algorithms.

According to Bouhriz et al. [111]:"*In the context of Information Retrieval System (IRS), semantic coherence between text and the terms chosen to represent them, enhance the precision of the returned results*". Therefore, it is important to design and implement semantic text processing methods to facilitate the selection of the most relevant terms. This would improve the capacity of these systems [111].

In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. Consequently, document indexing will also be

more meaningful if semantically related root words are used instead of stems.

Stemming algorithms have been developed in order to compress the size of documents and their index files up to 40% or 50% sometimes; by reducing the words in the document to as many common base forms as possible (i.e. storage space and processing time are also reduced). This procedure has had a direct impact in increasing the recall and thus, the search effectiveness. Indeed, the idea consists of increasing the number of relevant documents that are successfully retrieved in response to a query that would include the base forms of the words and not their different variants [112].

Although there is a difference between "*stemming*" and "*lemmatizing*" as in *stemming*, a set of rules is applied to form the final base forms without taking into account the textual context whereas in *lemmatizing*, the understanding of the Part of Speech (POS) and the context of the words in a sentence is very important before the reduction of the word forms can be performed, the basic function of both the methods, is to reduce a word variant to its *stem* in the case of *stemming* and *lemma* in the case of *lemmatizing [113 – 114]*. In this chapter, we talk about our proposed efficient stemming algorithm that also integrates the advantages of lemmatizing algorithms. Our main objective is to improve recall as well as precision in an IRS.

Despite the fact that context-aware stemmers provide more meaningful stems and rule-based stemmers take advantage of some language phenomenon which can be easily expressed by simple rules, they both are time-consuming tasks. In this regard, we developed an algorithm that has the advantages of a stemmer that uses the syntactical as well as the semantic-knowledge to reduce stemming errors. Indeed, the new hybrid stemming method is based on a combination of affix stripping (based on Porter Stemming algorithm), context-aware techniques (based on the Context-Aware Stemming "*CAS*" algorithm), and

corpus based techniques for English language (based on Wordnet). The Semantically Enriched Context-Aware Stemming Algorithm (SECAS) proposed can be effectively used in pre-processing stages of text summarization and classification systems in the context of Information Retrieval (IR) and its main goal is to provide meaningful stems in order to enhance recall without decreasing precision.

In general terms, the proposed method is divided into two main parts. The first part is a pre-processing phase; where stop-words, plural, and special characters are removed. While in the second part, the document is stemmed using an improved version of the *CAS* algorithm. We tested the performance of the proposed scheme with the *WT2G* dataset using the Terrier Platform. In order to achieve this, we implemented an evaluation platform that allow the indexing of *WT2G* dataset using Porter, *CAS*, and *SECAS* algorithms. The tests are then performed by producing result files with different threshold for query-document similarity and comparing them to *qrels.wt2g* file using the Terrier platform. The encouraging results indicate the superior performance of the proposed method compared with Porter and *CAS* algorithms as it provides in 99% of the cases meaningful stems with a precision up to 95% and a recall of 81%.

The remainder of this chapter is divided into four sections. First, the concept of stemming and an overview of its related problems and existing types are given. Then, we present our proposed Semantically Enriched Context-Aware Stemming Algorithm and our evaluation platform. In this regard, a discussion and an analysis of the promising results is provided. Finally, we outline the outcomes of the proposed stemming method and our future work.

6.2. Related work

Stemming, is an important pre-processing phase in most of Text-Mining and Natural Language processing applications [112, 114 – 116]. Moreover, it has been proved that the use of stemming in IRS can

improve many other related tasks, such as Machine Translation and Sentiment Analysis.

“Stemming is used to enable matching of queries and documents in keyword-based information retrieval systems.” [64]. In this regard, the purpose of stemming is to reduce the different grammatical variants of a word (noun, adverb, adjective, verb, etc.) to a common base form (root) called “stem”; supposing that the words that share the same stem, do have the same meaning. We talk about conflating (i.e. bringing together) all those variants to facilitate their retrieval by the IRS with the aim of improving the recall performance [117].

This section views the definition of stemming algorithms and the problems they might raise as well as their different approaches and classes.

6.2.1. The concept of stemming

In traditional IRS, a first approach to give back results in response to a query consisted of fetching all the corpus’ documents word by word and then, ranking the documents according to the number of their common words with the input query. This approach was very time consuming and induced to a loss in terms of accuracy. To overcome those drawbacks and increase the search results’ accuracy and relevance, stemming has been introduced and widely used over the last forty years. The idea was to reduce the words to their linguistic roots; by omitting their prefixes and suffixes (i.e. affixes) [118 - 120].

If, for example, a searcher enters the term freedom as part of a query, it is likely that he or she will also be interested in such variants as freeness, free, or freest. *Figure 6.1* synthesizes a very good example given by Rajput & Khare (2015) [116], where many syntactic variants can be found for the word “*Correct*”. That is to say this word can substitute very well the variety of its related words in the context of Information Retrieval Systems.

The words conveying the same meaning, should inevitably be stemmed to the same root even if they differ in the way they are written [113 - 114, 121]. Moreover, most languages of the world are inflected, meaning that they have different forms. This change can express differences in: number, tense, gender, aspect, or mood. This is why stemming is defined for individual languages because the stemming rules depend on how the language expands the root term. For a list of the languages that have a default stemming algorithm (Arabic, Dutch, French, English…), words tend to be constant at the front, and to vary at the end. However, for some of the world's languages, Chinese for example, the concept of stemming is not applicable.

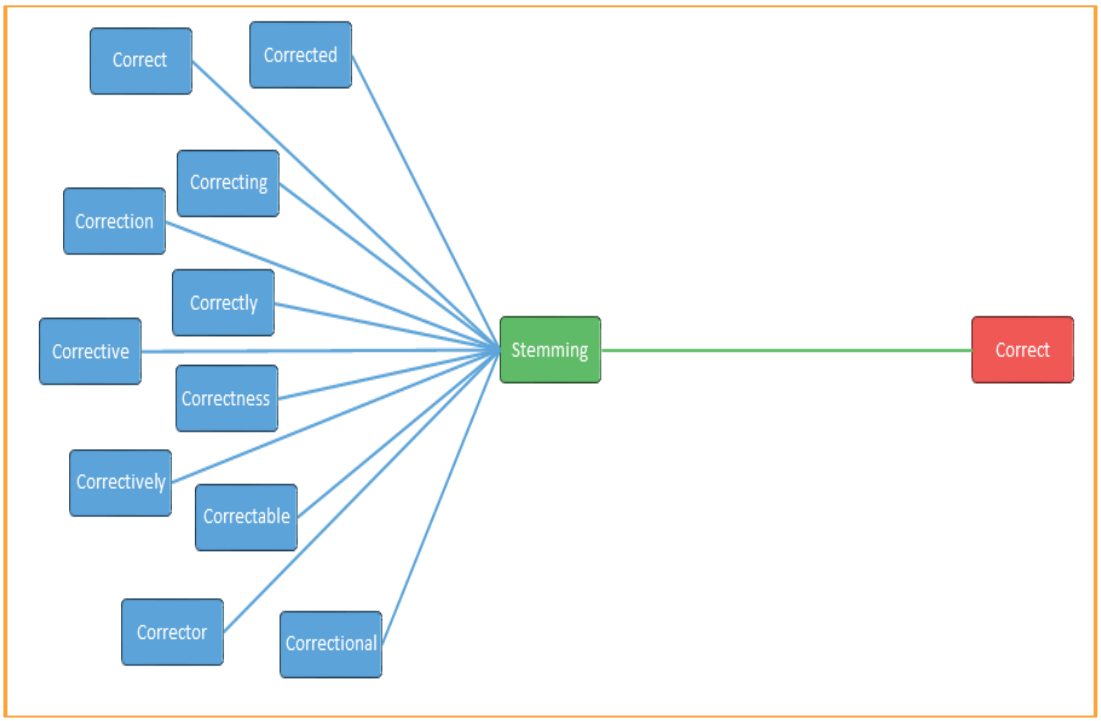


Figure 6-1: Stemming principle.

6.2.2. Stemming techniques

Vijayarani et al. [121] and Ruba Rani et al. [122] group stemming techniques into three main categories, namely [121-122]: truncating techniques, statistical techniques, and hybrid techniques as depicted in *Figure 6.2*. These techniques differ in the way stems are found.

Stemming can either be achieved manually using procedures and regular expression or automatically [116]. Automatic techniques can further be sub-divided into four techniques, namely: Affix removal, successor variety, and n-gram [115, 123-124].

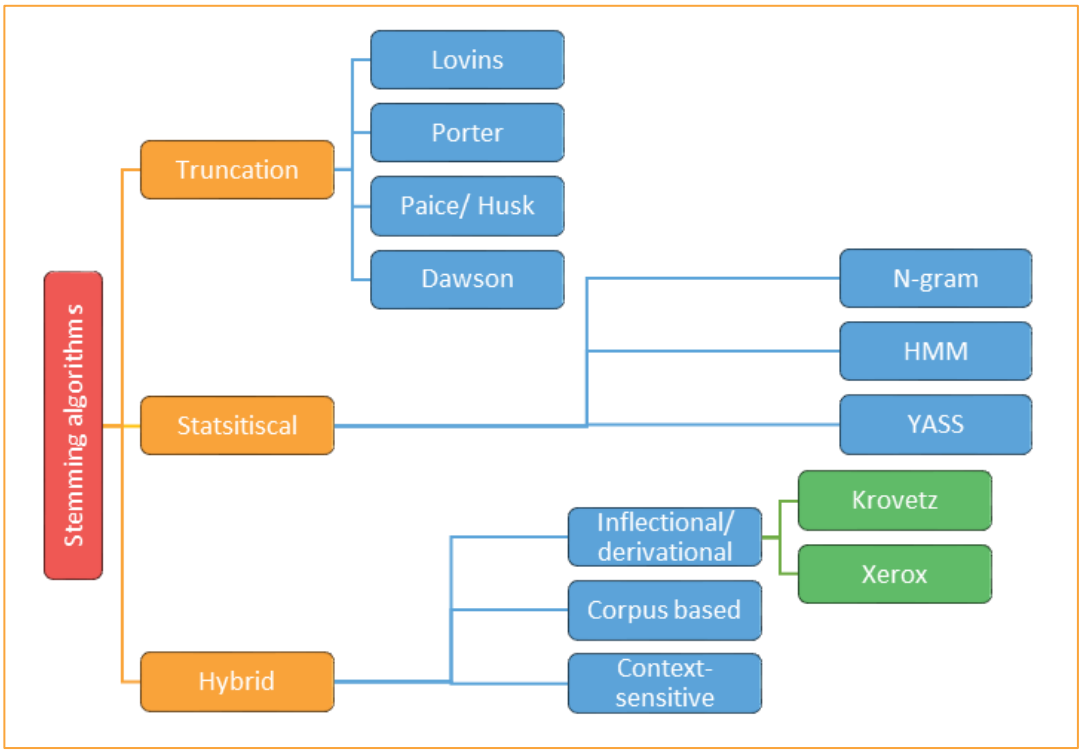


Figure 6-2: Stemming classes.

Finally, there is two kinds of Affix removal techniques "Longest match" and "Simple removal". *Figure 6.3* summarizes these different conflations approaches.

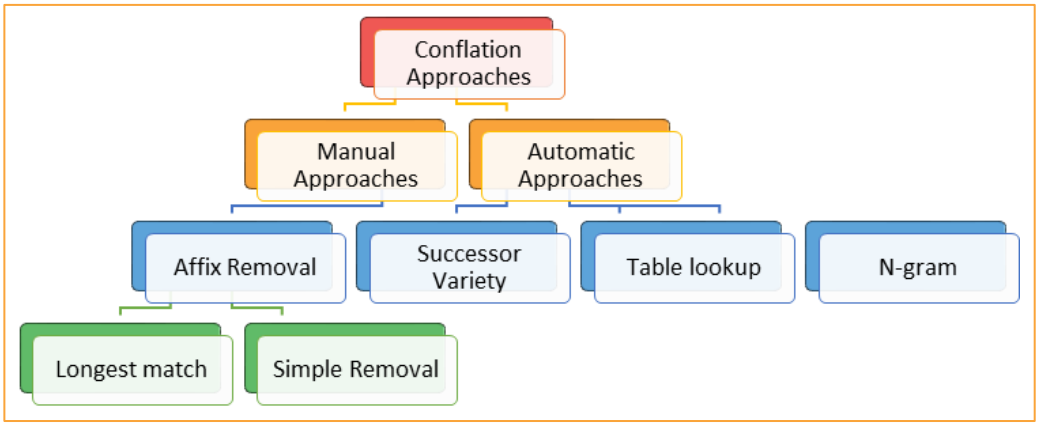


Figure 6-3: Conflation approaches.

The name "Stemming Algorithm" refers to the Affix removal techniques (the most common techniques) that remove suffixes or prefixes from words to form common stems [67, 115-117, 119]. While, Successor variety stemmers are based on the computation of frequencies of letter sequences in the text, N-gram methods conflate the words according to the number of shared di-grams or n-grams.

Gormley and Tong [125] classify the approaches above-mentioned (see summary in *Table 6.1*) into two main classes:

- *Algorithmic stemmers:* These are the rule-based stemmers. These algorithms are fast, use little memory, and give good results with regular words. However, they do not work very well with irregular words.
- *Dictionary stemmers:* Following the principle of table-lookup, these algorithms simply search for the stems in a dictionary. Dictionary stemmers are able to solve the polysemy and synonymy problems present in any language.

The study of all the aforementioned algorithms is out of the scope of our work, but we tried to describe a brief overview of the state-of-the-art in the area of stemming algorithms and got to make the following observations [117, 119, 122, and 126]:

- The majority of stemming's impacts on retrieval performance have been positive.

- Stemming increases significantly the retrieval performance regardless of the type or category of the stemmer.
- Rule based algorithms provide the highest accuracy among all the existing algorithms.
- The different stemming algorithms are quite similar in their objectives, but none of them give 100% output.

Table 6-1: Advantages and disadvantages of stemming algorithms.

Algorithm	Advantages	Disadvantages
Manual stemmers	The risks of incorrect conflations is reduced.	Time consuming; The impact on efficiency and effectiveness is insignificant.
Affix-removal techniques	Improvement in terms of retrieval efficiency and effectiveness as well as compression.	Errors can be made in the conflating process. Likewise, the plural of a word maybe conflated to a different stem then its singular form, e.g., "Flies" and "*Fly*" are stemmed to "*Fli*" and "*Fly*" respectively; These algorithms do not make use of any lexicon and do not understand the Part of Speech (POS) and the context of the words in a sentence.
Successor variety	These techniques are completely automatic; They are language independent; They give good results in the context of multi-lingual applications.	The corpus used for the cmputations must not be small.
Table lookup	This techniques is the simplest and fastest and their error rate is low.	The inflected forms that are not included in the table, can never be retrieved; Table may become large with time.
N-gram	These techniques are language independent.	The storage of indexes requires large storage capacity.

6.2.3. Stemming problems

A words' stem has been defined as follows: *"A word's stem is its most elementary form which may or may not have a semantic*

interpretation." [127]. Unfortunately, no stemming algorithm is perfect because in English documents for example, the information about the original terms might be lost.

Over stemming and under stemming are the most common problems of the stemming process. Gormley and Tong [125] define these issues as follows:

- Under-stemming occurs when two words with the same meaning cannot be reduced to the same root. As a consequence, the amount of 'False negative' increases making relevant documents irretrievable.
- Over-stemming occurs when two words with distinct meanings cannot be kept separate, e.g., *general* and *generate* can both be stemmed to '*gener*'. As a consequence, the amount of 'False positive' documents increases making IRS answer users' queries by returning irrelevant documents.

"*The Under-stemming and Over-Stemming Indexes are metrics of specific errors that occur during the implementation of a stemming algorithm. According to these metrics, a good stemmer should produce as few under-stemming and over-stemming errors as possible."* [127].

The implementation of an effective stemmer is based on finding the perfect balance between light stemming and heavy stemming [116, 119].

6.3. Our Proposed Method

We propose a synergetic hybrid solution as our stemming algorithm combines features from algorithmic stemmers and dictionary stemmers (i.e. decision based and hybrid stemmers). Indeed, as [67], our major objective is to maximize the proportion of the meaningful stems (root words), without compromising the other performance criteria. Moreover, the same word may have two meanings, e.g., the word "*Novel*" (noun) is a "*fictional book"* of significant length, but it does also mean something "*new*" or "*different*". To overcome those cases, we proposed an algorithm

that combines dictionary stemmers' advantages and those of rule based stemmers. Actually, the overall algorithm proposes root words much comparable to a lemma. While, both Lemmatization and stemming aim to normalize related words by identifying their canonical representative, the lemmatizing process is more complicated insofar as it needs to understand the context in which a word is used in order to make decision about its meaning. Indeed, lemmatization would try to distinguish the different word senses, while stemming would incorrectly conflate them.

The present chapter proposes an improved version of the *CAS* algorithm for the English language. The proposed stemmer is evaluated using the wt2g dataset together with the Terrier platform. With this latter, the performance of a stemmer is computed by calculating the precision and recall of the algorithm in retrieving the 247,491 documents of the dataset regarding the 50 queries. The obtained results show an improvement in stemming accuracy, compared with the *CAS* stemmer, but also compared to the original *Porter* stemmer. We proved, in addition, that the new version of porter stemmer affects the information retrieval performance as it takes into account the semantic and contextual dimensionality of the generated stems.

Despite the fact that the *CAS* algorithm does, in some cases, derive meaningful stems from the words (which imply that the derived stems are linguistically correct when compared with most Porter's stem words), we remarked that it fails sometimes to conflate words to their real stem. In addition, it fails to detect the right stem in the case of irregular verbs or plural nouns.

In this sub-section, we describe our Semantically Enriched Context-Aware Stemming (*SECAS*) algorithm in detail.

6.3.1. Overview of the SECAS algorithm

SECAS algorithm was proposed to lessen the problems of traditional stemming approach that performs blind transformation of all query and

document terms without considering the context of the stemmed word for effective search with regard to context awareness. The application flow involves the enrichment of the *CAS* algorithm, by adding a dictionary stemming phase to ensure the generation of meaningful root words.

This proposed methodology includes some pre-processing steps that are tokenization, removal of digits, punctuations and stop word removal before entering into a stemming process. Further it concentrates on the generation of meaningful stems by using Wordnet database to obtain the most accurate roots according to the nature of the original words (*noun*, *verb, adverb* or *adjective)*. The stem identification with the Wordnet is based on the computation of semantic similarity measure [78], which have been proven to provide good results in the case of Natural Language Processing applications.

The design process of our system (as depicted in *Figure 6.4*), includes two main types of treatments: pre-processing and indexing.

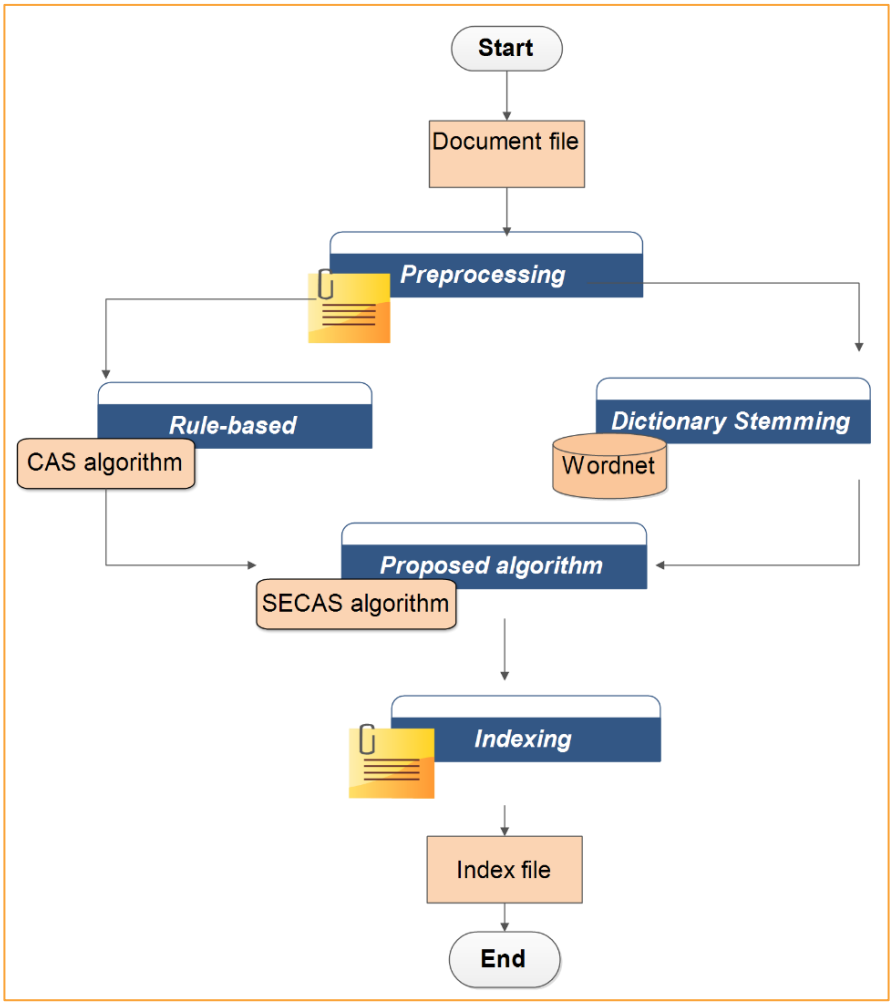


Figure 6-4: Architecture diagram of the indexing method.

Figure 6.4 illustrates that the stemming process begins with an acquisition phase of the document, followed by a pre-processing phase; including tokenization, stop-words removing, etc. According to the above flowchart, the stemming process include both a rule-based stemming phase with an altered version of the *CAS* algorithm (addition of pre-processing step) and a dictionary stemming phase; based on the Ontology Wordnet. The obtained results are compared and finally the *SECAS* stemming algorithm is applied to obtain the final index terms. This architecture can be broken down to 7 steps:

- Step1: Text document is given as input to the stemmer.
- Step 2: Removal of punctuation, digits and stop word like preposition, conjunction, article, etc.

- Step 3: Identification of the *CAS* descriptors.
- Step 4: Identification of the Wordnet descriptors.
- Step 5: Application of the *SECAS* algorithm and comparison of the descriptors.
- Step 6: Identification of the most relevant descriptors.
- Step 7: Creation of the final index file as an output to the stemmer.

The affix removal approach along with a dictionary stemming algorithm contribute to reduce the space occupied by the word in the memory of the computer; mainly because in natural languages, there are many words that differ in the way they are written but share similar roots.

In case of inflection, the word variants are extracted according to the language specific syntactic variations like plural, gender, case, etc. Whereas in the case of derivation, the word variants are extracted according to the part-of-speech (POS) of a sentence where the word occurs.

In the upcoming sub-sections, a detailed explanation of all the phases described before is given.

6.3.1.1. Natural language processing module

Hence, the retrieval decision is made by comparing the terms of the query with the index terms (important words or phrases) appearing in the document itself, the decision may be binary (retrieve/reject), or it may involve the estimation of the relevance degree between the document and the query. So, before the documents are stemmed, data pre-processing techniques are applied on the collection in order to reduce its size by deleting as many structural variants of words with same meanings as possible. This action would increase the effectiveness of IR System [128].

Pre-processing is one of the most important phase in many Text-Mining, Natural Language Processing, and Text Indexing applications. Before applying the SECAS algorithm, we added a pre-processing phase that includes:

- *Tokenization:* According to Jayanthi and Jeevitha [112] "Tokenization is the process of breaking the sentences as well as the text file into word delimited by a white space, a tabulation, or a new line". In other words, Tokenization is the process of converting a stream of characters (the text of the documents) into a stream of words (the candidate words to be adopted as index terms) – i.e. identification of the different words in the text.
- *Stop words removing: "A stop-word is a word that holds no meaning on its own."* [115]. The most common words make up around 50% of all text's content in most languages [129]. These words are not useful in describing the user's information need. The same goes for determiners, coordinating conjunctions, prepositions, articles, auxiliary verbs, relative pronouns, etc. Thus, stemming those words would only cause the IRS to slow down without enhancing, nor improving the indexing phase or the query expansion [116].

  The removal of stop-words and duplicates is a function that the current Porter stemming algorithm as well as the *CAS* algorithm does not address. However, in the case of *SECAS* algorithm, we included a stop-word removing phase. In this regard, a list of 710 stop word was used; including 210 HTML Mark-ups. Moreover, duplicate names were deleted and some other nouns like emails, and website names were cleaned.
- *Punctuation Removal*: In order to obtain good results and speed up the stemming process, it is also important to remove all the punctuation as well as accents enclosed in texts' contents as they are meaningless and irrelevant for a given search task.

6.3.1.2. Indexing module

After processing all the steps aforementioned, document's words are ready for the stemming process. Dictionary approach together with rule-based approaches were taken into consideration in the development of the stemming algorithm.

Hereafter, we describe the *SECAS* algorithm. For more details about the porter and the *CAS* algorithm, please refer to [67].

With the *SECAS* algorithm we wanted to propose an indexing technique that combines features from dictionary stemmers as well as algorithmic stemmers.

Algorithm 1: The Semantically Enriched Context-Aware Stemming Algorithm

```
1  SECAS (word)
   Input  : word as a string
   Output: indexTerm a string representing the best index term
2  begin
3      wordnetIndexTerm ⟵ WordnetStemming(word);
4      casIndexTerm ⟵ CAS(word);
5      indexTerm: empty string;
6      if word is a compound noun then
7          indexTerm ⟵ word;
8      end
9      else if wordnetIndexTerm <> word OR casIndexTerm <> word
        then
10         if length(wordnetIndexTerm) < length(casIndexTerm) AND
            length(wordnetIndexTerm) > 0 then
11             indexTerm ⟵ wordnetIndexTerm;
12         end
13         else if casIndexTerm has at least one definition then
            casIndexTerm is shorter than wordnetIndexTerm
14             indexTerm ⟵ casIndexTerm;
15         end
16     end
18     return indexTerm
19 end
```

Even in the case of *CAS* and *Porter* algorithm, we chose to keep the compound nouns without stemming them, because the application of stemming rules would alter the meaning of the former words.

Algorithm 2: Wordnet Stemming

```
1  WordnetStemming(word)
   Input  : word as a string
   Output: stem the shortest Wordnet stem as a string
2  begin
3      stem: string;
4      noun: shortest stem whose type is noun;
5      verb: shortest stem whose type is verb;
6      adjective: shortest stem whose type is adjective;
7      adverb: shortest stem whose type is adverb;
8      set nounPriority = 1 AND verbPriority = 2 AND
       adjectivePriority = 3 AND adverbePriority = 4 ;
       /* 1 is the highest priority.  Nouns being more
       significant than verbs, verbs more significant than
       adjectives, Etc.                                    */
9      stem ⟵ shortest meaningful stem with the highest priority;
11     return stem
12 end
```

6.3.2. Experimental Results

The criteria for judging stemmers include the retrieval effectiveness measured by recall, precision, speed, etc., the compression performance, and the correctness of the stems [114 – 117, 119, 124, 127].

Stemmer evaluation measures have been discussed and evaluated in literature [67]. The main objective of most IRS consists of improving the recall while preserving the precision. A recall increasing method which can be useful for even the simplest Boolean retrieval systems is *stemming*. Moreover, another important point in evaluating an Information System is improving the *effectiveness* and the *efficiency* of a search. Where, *effectiveness*, measured by recall and precision, represents how well the rankings generated by a search engine correspond to the rankings based on user relevance judgments and *efficiency* represents the time and space requirements for the algorithm to generate those rankings. Noteworthy, there is no reliable technique that significantly improves

effectiveness that cannot be incorporated into a search engine due to *efficiency* considerations.

Among the notable criteria for judging stemmer performance, includes compression performance, stemming speed, retrieval performance, and correctness. In addition, the extents of over stemming and under stemming are two other measures that indicate how incorrect a stemmer can be.

In this section, we present our evaluation platform for the three stemming algorithm: *Porter*, *CAS*, and *SECAS*. We focus on *Precision* and *recall* to evaluate the *accuracy* and *effectiveness* of the proposed algorithm.

6.3.2.1. Learning corpora (dataset)

Test Collections are the basis for advancing knowledge in IR. *TREC* collections are constantly increasing in size and widely used in the IR field. In our evaluation, we used the *WT2G* dataset (from TREC collection) together with the *Terrier* platform to prove the effectiveness of our stemming, retrieving, and ranking algorithm. TREC analysts judge a document as relevant if it contains the information that could be used to help write a report on the query topic. The drawback of TREC evaluation is that the relevance is binary.

We made use of the open-source version of *Terrier*[27], which provides a comprehensive, flexible, robust, and transparent test-bed platform for research and experimentation in IR [133].

6.3.2.2. Evaluation platform

Figure 6.5 describes the process flow of our evaluation platform. The corpus of documents is handled by a processing and indexing module, which generate the index files of all the corpora containing the relevant index terms (in our cases, we have three index files for each document).

[27] http://terrier.org/

Moreover, the *WT2G* queries; known as topics are also processed, then, refined versions of the topics based on their narration are generated. After the similarity measures have been computed, it is a matter of finding the relevance assessment of each query; answering the question: which documents are relevant to the query? Likewise, a file similar to the terrier platform result file is generated with the purpose of computing the performance results. The final step consists of evaluating this file by the Terrier platform to obtain precision and recall of all the algorithms. The different modules are described here after.

Document processing

In the case of *WT2G* dataset, the cleaning process of a document includes the following phases:

- corpus cleaning;
- document's "*docno*" extraction;
- document's "*docno*", "*title*", "*metadata*", and "*text*" cleaning;
- Storage of plain texts.

A first cleaning phase is necessary in order to divide the corpus into a set of documents. Where, each document is represented by an identifier "*Docno*", a "*Title*", "*Matadata*", and a "*Text*". A second cleaning consist of: deleting HTML Mark-ups, transforming text to lowercase, trimming the words by deleting blank characters, and deleting the unnecessary characters and punctuations. At the end of this phase, we obtain the results in plain text.

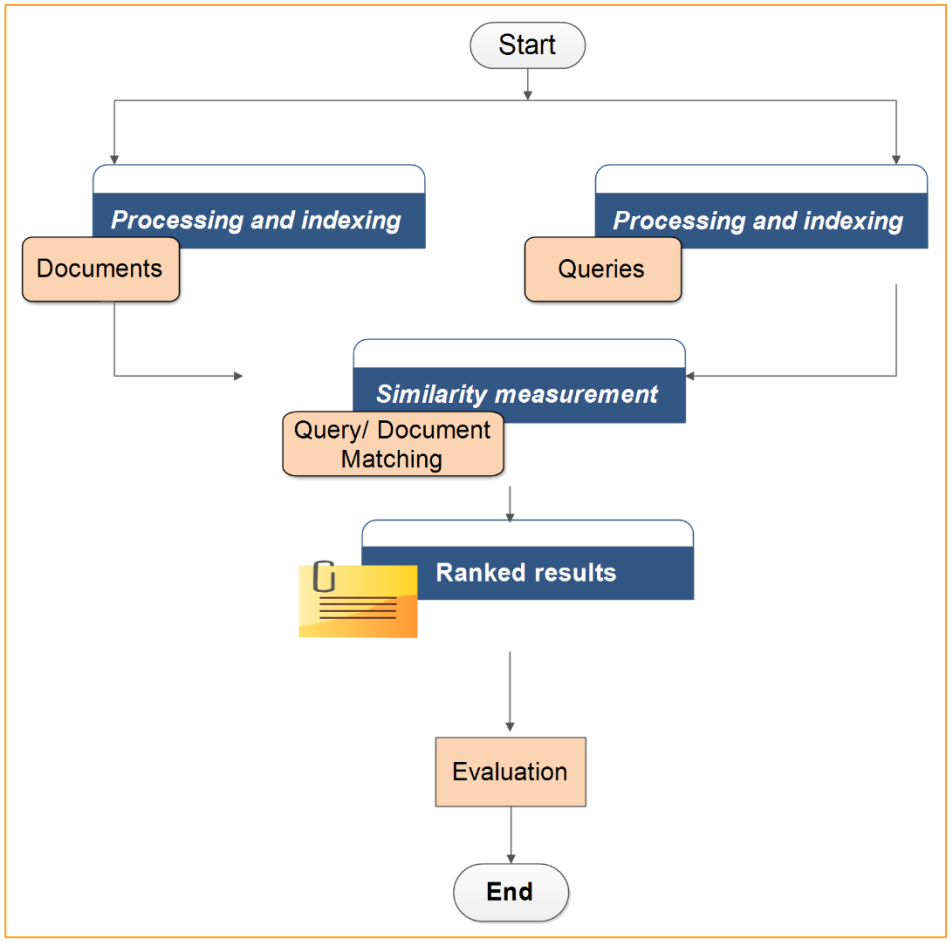


Figure 6-5: Architecture diagram of the evaluation Platform.

Query processing

In *WT2G,* we talk about topics rather than queries. The processing of the query is almost the same as that of the document, except for the cleaning phase. Indeed, there are no HTML Mark-ups, only those corresponding to "*top*", "*num*", "*title*", "*desc*", and "*narr*", which will help us represent the set of 50 queries (i.e. topics) according to their number (*num*), title, description, and narration respectively.

Once we obtain plain texts of the queries and documents, we proceed by a natural language pre-processing phase as described above.

Document indexer

The "SECAS indexing" algorithm is described below.

Algorithm 3: Document Indexer

```
1  DocumentIndexer (corpus)
   Input  : corpus WT2G documents
2  begin
3      Browse the corpus;
4      foreach document in corpus do
5          list ⟵ empty List of strings;
6          if document not indexed then
7              foreach word in document do
8                  Find SECAS(word);
9                  Insert the indexTerm in list;
                   /* The ordering of the words is respected      */
10             end
11         end
12         Refresh the index;
13     end
14 end
```

Query indexer

The "*query indexing*" algorithm is described below.

Algorithm 4: Query Indexer

```
1  QueryIndexer (query)
   Input  : queries a table containing records of WT2G topics; represented
            by their number (num), title, description (desc), and
            narration(narr)
2  begin
3      hashMap ⟵ empty HashMap;
4      foreach query in queries do
5          Insert queryNum and queryNarration in the hashMap from the
           database table
6          foreach term in queryNarration do
7              Find indexTerm;
8              Update hashMap;
9          end
10         Update the database table;
11     end
12 end
```

6.3.3. Similarity computation module

The query-document similarity in the *queryDocumentSimilarity algorithm* was computed using the same process as in the work Merging Ontology by Semantic enrichment MOnSE [131].

An Ontology is a formal, explicit specification of a shared conceptualization [132]. Where, the conceptualization is the couching of knowledge about the world in terms of entities (things, the relationships they hold and the constraints between them). The specification is the representation of this conceptualization in a concrete form. One step in this specification is the encoding of the conceptualization in a knowledge representation language. The goal is to create an agreed-upon vocabulary and semantic structure for exchanging information about that domain.

Likewise, as the semantic aspects are the most dominant aspects in an Ontology, the aim of our system was to overcome the significant limitations encountered in previous Ontology Merging systems by proposing a relevant similarity measure. This measure is based on a weighted combination of a low level similarity measures, namely terminological similarity (itself, a combination of a lexical and a syntactic similarity) and structural similarity. As the obtained results were very promiscuous, we altered the method for it to be adapted to the information retrieval issue.

For evaluation purposes, five similarity threshold (0.6, 0.7, 0.8, 0.9, and 1.0) were established and then the similarity algorithm returns 1 if the computed similarity is greater than or equal the threshold and 0 in case the similarity is less. First, the query keywords and the document indexes are represented by vectors upon which we build a matrix of correspondences' where each term of a query is compared, respectively, to all the indexed terms. Finally, the similarity of a query with the entire corpus' documents is computed, recursively, to generate our result file of the WT2G collection. Here after, we describe the *queryCorpusMatching* algorithm and then the *Query-Document similarity measures*.

Algorithm 5: Query-corpus matching

```
1  QueryCorpusMatching (threshold)
   Input  : threshold as a double
   Output: qrels.secas a file to be compared with the qrels.wt2g from
           WT2G using Terrier platform
   Data:
   simDegree: double;
   descriptors: a file containing all the index terms of the corpus'
   documents. Where, each line corresponds to a document (represented by
   its docno according to WT2G dataset);
   queries: a file containing all the keywords of the dataset, where each line
   corresponds to a query (represented by its num according to WT2G
   dataset);
   qrels.wt2g: Terrier evaluation file of the WT2G dataset;
2  begin
3      docno, docValue, num, queryValue ⟵ empty strings;
       qrels.secas ⟵ file to be compared with the qrels.w2tg using Terrier;
4      foreach num in queries do
5          foreach docno in descriptors do
6              docValue ⟵ line of descriptors corresponding to a specific
               docno;
7              queryValue ⟵ line of query keywords corresponding to a
               specific num;
8              couple ⟵ is a string equals to num + "0" + docno; if couple
               has not been treated yet then in case of shutdown or crash
9                  simDegree ⟵ queryDocumentSim(queryValue,
                   docValue); if simDegree >= threshold then
10                     add couple + "1" to qrels.secas;
11                 end
12                 else
13                     add couple + "0" to qrels.secas;
14                 end
15             end
16         end
17     end
19     return qrels.secas
20 end
```

More precisely, the similarity between a query keyword and a document indexer is computed from a semantic correlation Matrix. Where, the lines represent query terms and the columns represent the document index terms and between them the computed semantic similarities. The semantic similarities are based on the calculation of global similarity

measures by the combination of terminological and structural similarities. A detailed explanation is given bellow.

6.3.3.1. Terminological similarity

Pairs of words are compared and Terminological Similarity Measures is computed based on syntactic and lexical comparison. Syntactic methods are based on the comparison of words, strings or texts based on the letters they have in common. Linguistic or lexical methods make use of external resources (dictionaries, taxonomy…) to perform the comparison; where the similarity between two entities, represented by terms, is calculated using semantic links that exist in those resources. In our mapping algorithm, the Jaro distance [131] is used for the syntactic similarity and WordNet for lexical similarity.

6.3.3.2. Syntactic similarity

The Jaro distance [131] takes into account in the comparison of two character strings, on the one hand, the number of characters in common, and also the order of the characters. It has been proved that the Jaro distance provide an interesting performance and is faster as compared to other syntactic similarity computation methods like Monge-Elkan[28] for example.

This measure is particularly adapted to short chains comparison and then will be perfect in mapping document descriptors and query keywords as these later have been subjected to an indexing phase. The result is normalized, so as to have a measure between 0 and 1, zero being the absence of similarity. Jaro distance between strings s1 and s2 is defined by:

[28] For more details, refer to the paper [133] (A. Monge and C. Elkan, “The field-matching problem: algorithm and applications.” 1996).

$$synSim = \frac{1}{3}\left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m}\right) \textit{(13)}$$

Where: m is the number of corresponding characters, and t is the number of transpositions.

Two identical characters of S_1 and S_2 are considered as corresponding, if their distance (i.e. the difference between their positions in their respective chains) does not exceed:

$$\left(\frac{\max(|S_1|,|S_2|)}{2}\right) - 1 \quad \textit{(14)}$$

The number of transpositions is obtained by comparing the i^{th} character S_1, with the corresponding i^{th} character of S_2. The number of times these characters are different, divided by two, gives the number of transpositions.

6.3.3.3. Lexical similarity

The lexical methods require the use of an external resources. Several types of resources can be used, but we chose WordNet. WordNet[29] is a large lexical database of English where, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets) each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. Freely and publicly available for download, WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

The lexical similarity function used in SECAS is somehow different from the function used in MOnSE, in way that the group of synset upon which the similarity is computed must be selected manually by experts in MOnSE system, but is randomly generated in SECAS. Let

[29] For more details, see http://wordnet.princeton.edu/wordnet/

$\min(|Syn(S_1)|,|Syn(S_2)|)$ be the minimum of the cardinalities of two sets $|Syn(C_1)|$ and $|Syn(C_2)|$ and $\beta = |Syn(S_1) \cap Syn(S_2)|$ *(14)*, the set of common sysnsets generated by Wordnet. Then the similarity between two strings S_1 and S_2 is defined as follows:

$$lexSim(S_1, S_2) = \frac{\beta}{\min(|Syn(S_1)|,|Syn(S_2)|)} \quad (15)$$

This measure return *1* if at least S_1 and S_2 have 1 common synset. *0* is returned in case S_1 and S_2 are not synonyms, and have no lexical relation (hyponymy, antonyms…).

After the lexical and syntactic similarity have been measured, the obtained values are combined by the following formula:

$$terSim(S_1, S_2) = \frac{(lexSim(S_1, S_2) \times lexCoeff) + (synSim(S_1, S_2) \times synCoeff)}{(lexCoeff + synCoeff)} \quad (16)$$

Where *Coeff* is a numerical coefficient calculated as follows:

$$coeff = Exp^{sim} \quad (17)$$

Structural similarity methods deduce the similarity of two words, using structural information, when the entities involved are linked to others by semantic links, forming a hierarchy of entities. The internal structural methods calculate the similarity between two concepts (words, strings…), using the information on their internal structure, whereas external structural or conceptual methods use the hierarchical structure of an ontology, by counting the number of arcs in the hierarchy to determine the semantic similarity between two entities. Wu and Palmer [78] define the similarity, in terms of the distance which separates two words in a hierarchy (in this case Wordnet hierarchy) and also by their position comparing to the root. The similarity is defined relatively to the distance between two words, taking into account their Lowest Common Ancestors

(LCA) and the root of the hierarchy. In our system, the structural similarity is calculated by measuring the Wu and Palmer value given bellow.

$$strSim(S_1, S_2) = \frac{depth(LCA(S_1, S_2))}{(depth(S_1) + sdepth(S_2))} \quad (18)$$

Where $LCA(S_1,S_2)$ is the lowest common ancestor of *S1* and *S2* and $\text{depth}(\text{LCA}(S_1, S_2))$ is the number of edges between $\text{LCA}(S_1, S_2)$ and the root. In the same way, $\text{depth}(S_1)$ and $\text{depth}(S_2)$ represent the number of edges between the string *S1* and *S2*, respectively, and the root.
Finally, the semantic similarity is calculated by the combination of terminological and structural similarity.

$$semSim(S_1, S_2) = \frac{(terSim(S_1, S_2) \times terCoeff) + (strSim(S_1, S_2) \times strCoeff)}{(terCoeff + strCoeff)} \quad (19)$$

After these steps we obtain a matrix of similarity measures "*Correlation matrix*" and to the
query-document mapping to be computed, we follow these final steps.

Let *M be* the correlation matrix, where each element *(i, j)* describes the semantic similarity $semSim(qi, tj)$ between a query keywords q_i and a document index terms t_j. From this Matrix, a similarity vector *simVector* is generated, as the length of the vector equals the number of query keywords and each element v_i of the vector represents the $max(semSim(qi, tj))$ of the corresponding line in the matrix. This step ensures that only the highest similarity value is retained for the query keyword with a given document. Finally, in order to get the query-document mapping value "*simDegree*", we proceed by the calculation of the average similarity. Such as $AVGSim\left(simVector\right) = \frac{SUM\,(Vi)}{|simVector|}$ *(20)*

$$qi \overbrace{\begin{bmatrix} M1,1 & \dots & M1,n \\ \vdots & \ddots & \vdots \\ Mm,1 & \dots & Mm,n \end{bmatrix}}^{dj} \rightarrow \overbrace{\begin{bmatrix} \boldsymbol{v}1 \\ \vdots \\ \boldsymbol{v}m \end{bmatrix}}^{\max(semSim(qi,tj))} \rightarrow simDegree = AVGSim(simVector) \quad (21)$$

To evaluate our indexing algorithm and compare the obtained results with *Porter* stemming algorithm and *CAS* algorithm, we proceed, in the same way with the three algorithms and finally submit the three generated files to the Terrier platform in order to compare them.

Hereafter, *Figure 6.6* summarizes the IRS we developed to test our stemming algorithm.

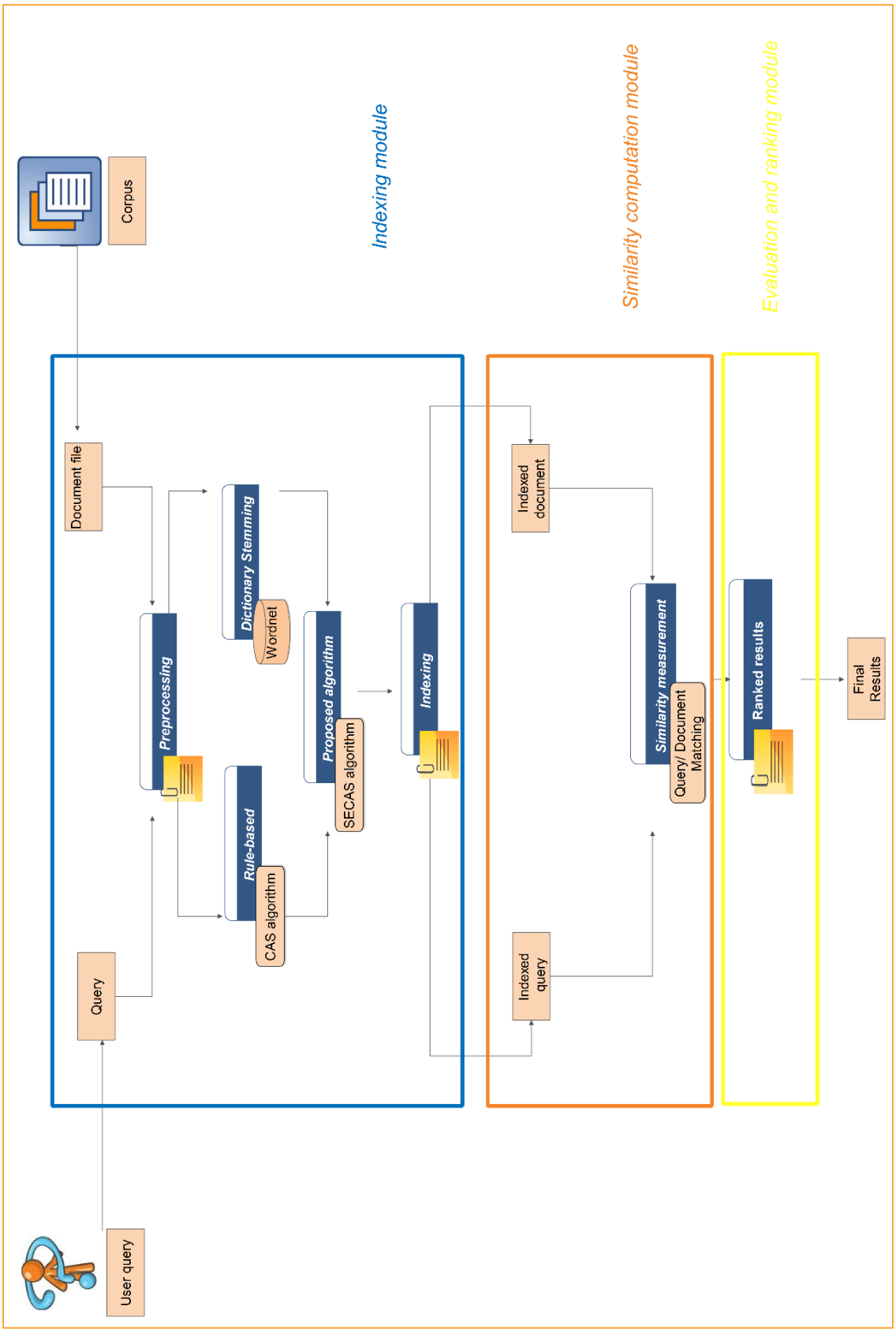


Figure 6-6: Architecture diagram of the Information Retrieval System.

6.4. Evaluation results

As mentioned before, the *TREC-8 WT2G* was used as a training dataset and the Terrier platform was only used to evaluate the final files, as *indexing* and *query-document* mapping were performed using our own platform. The Web Track Two-Gigabyte (*WT2G*) 1999 dataset, is a 2.1 gigabyte dataset roughly comparable with the original *TREC* text collection. It contains 1,081 files and 247,491 web documents grouped in collections. The dataset has successfully been used in information retrieval experiments. Moreover, queries and relevance judgments are also available. *Table 6.2* summarizes the obtained performance measures.

Table 6-2: Evaluation results.

Threshold		1.0	0.9	0.8	0.7	0.6
Porter	*Precision*	0.00	0.00	0.00	0.00	0.92
	Recall	0.00	0.00	0.00	0.00	0.34
	F-score	N/D	N/D	N/D	N/D	0.50
CAS	*Precision*	0.00	0.00	0.00	0.26	0.91
	Recall	0.00	0.00	0.00	0.37	0.80
	F-score	N/D	N/D	N/D	0.31	0.85
SECAS	*Precision*	0.00	***0.15***	***0.43***	***0.95***	***0.95***
	Recall	0.00	***0.18***	***0. 68***	***0.81***	***0.81***
	F-score	N/D	***0.16***	***0.53***	***0.87***	***0.87***

As depicted in *Figure 6.7*, we note that when the correspondence threshold equals *0.6* (i.e. low similarity), *SECAS* outperforms *CAS* and Porter algorithm and *CAS* outperforms Porter. Furthermore, *SECAS* outperforms *CAS* when the threshold is put to *0.7* in terms of precision (*95%* for SECAS and *26%* for CAS), recall (*81%* and *37%* respectively), and F-score (87% and *31%* respectively). Noteworthy, *SECAS* also gave some results with rigid threshold *0.8* and *0.9*, while *CAS* and Porter gave none. Finally, no algorithm was able to find even irrelevant results with threshold equals *1.0*.

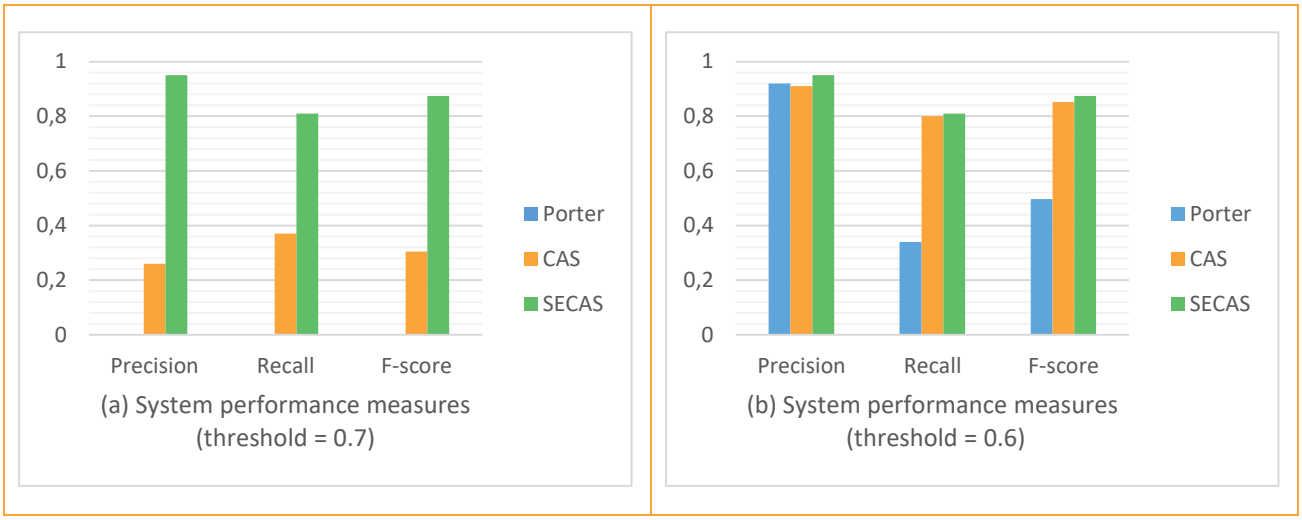


Figure 6-7: System performance measures.

Furthermore, a theoretical comparison with well-known stemmers in the literature based on the work of Singh & Gupta [119] shows the superior performance of our stemmer as reported in *Figure 6.8*.

We believe that these very encouraging results were achieved, not only thanks to the Stemming algorithm, but also because of the significant improvement obtained by the new mapping method. So, in order to deepen the evaluation about the Stemming algorithm, we analysed the compression of the corpus and found that its size was reduced to one third ($\frac{1}{3}$) thanks to our stemming algorithm and its normalization phase.

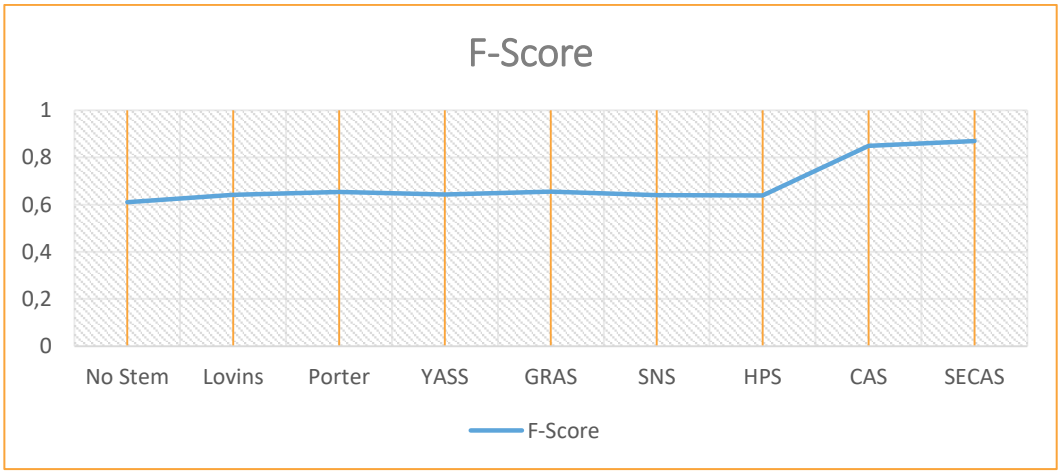


Figure 6-8: Classification of various stemmers in terms of accuracy.

Moreover, we analyzed the indexing speed (*Table 6.3*) of the *WT2G* dataset with *SECAS*, *CAS*, and *Porter* respectively and got to remark that

the indexing time in *SECAS* takes twice as much time as those of *CAS* or *Porter* algorithm. This is due to the fact that the stemming phase, in *SECAS* is based on a combination of two stemming algorithms. Finally, it is worth noting that the necessary time to index the *WT2G* collection using the *Terrier* platform varies from *7.53* to *23.60* min; depending on the number of phases (*1 or 2*) and whether the terms positions (blocks) are stored or not. This corresponds to half of the amount of time necessary to *SECAS* to index the whole collection. That is to say that the obtained results are encouraging, but despite that, we aim to improve the performance of both the stemming and query-document matching platform after completing some additional experiments on some larger and latest corpus.

Table 6-3: SECAS Indexing speed.

	Algorithm	Milliseconds	Minutes	Hours
Cleaning	N/D	412206	6.87	
Query indexing	Porter	227	0.004	
	CAS	211	0.003	
	SECAS	484	0.008	
Documents indexing	Porter	1572604	26.210	0.44
	CAS	1405798	23.421	0.39
	SECAS	3747068	62.451	1.04

6.5. Discussion

In *SECAS*, all the obtained stems are valid English terms. The proposed algorithm provides a morphological analysis of all document's words and identifies their base form. *SECAS* can analyze and generate inflectional and derivational morphemes; reducing each word to one of the four following forms that can be found in the dictionary: singular nouns, verbs in their infinitive form, adjectives in their positive form, and finally adverbs.

Our stemming algorithm generates stems with respect to the form and semantics of the original words, e.g., government stems to govern

while department is not reduced to depart since the two forms have different meanings. All stems are valid English terms, and irregular forms are handled properly.

The advantages of *SECAS* are:

- It works well with large documents.
- All stems are valid words since a lexical database that provides accurate forms for the words is used in the stemming process.
- It has been proved to give better results than the *CAS* and original Porter stemmer in the case if Information Retrieval application.

One of the disadvantages is that the algorithm is language dependent as the output depends on the lexical database which may not be exhaustive. Since this method is based on a dictionary or a thesaurus (Wordnet database in our case), it cannot correctly stem words which are not part of that dictionary. It is of utmost importance that the lexicon being used is totally exhaustive which, is a matter of a language study.

Moreover, the rule based approach may not always give correct output and the stems generated may not always be correct words.

Table 6.4 synthesizes some of the advantages and drawbacks of the *SECAS* algorithm.

Table 6-4: Comparison between SECAS, CAS, and Porter.

	Precision	Recall	Speed	Compression	Language dependence
SECAS	+++	+++	+	+++	+
CAS	++	++	+++	++	+++
Porter	+	+	++	+	+++

"In practice, a good algorithmic stemmer usually outperforms a dictionary stemmer" [125]. There are a couple of reasons for this. First, a dictionary stemmer is as good as its dictionary only. Secondly, the meaning of words might change over time. Finally, if a dictionary stemmer is confronted with a word that is not in its base, it does not know how to handle it. On the other hand, an algorithmic stemmer is relatively smaller,

faster, and simpler. It will always apply its precise set of rules and thus, provide the same results, whether these results are correct or not.

Many stemmers were developed in order to meet the degree of performance provided by the rich linguistic features of natural languages [118]. Most of the stemmers made explicit statistical-based or linguistic-based decisions to select only one root. Other stemmers use rankings to express their selection preference rather than simply supplying a single root. However, at the end, there always remain a unique stem is chosen.

The work exposed in this chapter supports the overall idea that the addition of features such as semantic and context is very valuable to improve the stemming results.

6.6. Conclusion

In this chapter, the Porter and CAS stemmers were studied with the aim to propose a new hybrid stemming method for Information Retrieval purposes. The main advantage of our method over existing methods is that it provides an accurate stemmer with meaningful stems in 99% of cases. The stemming algorithm begins by a pre-processing phase, then combines features from algorithmic stemmers and dictionary stemmers aiming to maximizing the proportion of the meaningful stems, without compromising the other performance measures (i.e. enhance recall without decreasing precision). Indeed, the new hybrid stemming method is based on a combination of affix stripping (based on Porter Stemming algorithm), context-aware techniques (based on the Context-Aware Stemming “CAS” algorithm), and corpus based techniques for English language (based on Wordnet). The Semantically Enriched Context-Aware Stemming Algorithm (SECAS) proposed can be effectively used in pre-processing stages of text summarization and classification systems in the context of Information Retrieval.

Besides the fact that our Semantically Enriched Context-Aware Stemming algorithm reduces the size of index files as much as 60%, it

also enhances recall and precision as compared with *Porter* and *CAS* algorithms in an evaluation with the *WT2G* dataset. Indeed, usually, stemmers increase recall at the cost of a decreased precision. These results were partly achieved because over stemming and under stemming problems were reduced by taking into consideration the syntax, as well as the semantics of the words and their POS in the stemming process. This in conjunction with a dictionary look-up helped in reducing the errors and converting stems to meaningful words.

However, no perfect stemmer has been designed so far to match all the requirements of an Information Retrieval System and *SECAS* is of no exception in its first version. Indeed, its advantages can constitute at the same time its main drawbacks. For instance, the same word can have a lot of synonyms and forms and thus, performing the corpus-based stemming phase in *SECAS* would be very heavy and time consuming. Moreover, we have seen that unlike *MOnSE*, where the calculations were made upon manually selected synsets, the mapping algorithm we used for evaluation generates random synonyms and then the lexical similarity in the *query-document* mapping phase takes a long time then it is allowed in nowadays in-a-hurry and mobile world.

So as to cope with this issues, our current work focusses on proposing a context-aware indexing algorithm as well as a context-aware mapping technique by integrating other contextual dimensions than part of speech and position of the word in a document or a query. More precisely, the focus of our upcoming work concerns the social dimension in Context-aware Information Retrieval Systems. Indeed, we succeeded to obtain very encouraging results by using an improved version of our Context-Aware Information Retrieval System in the domain of Big Data (more precisely, Social Bookmarking).

CHAPTER 7 PROPOSITION OF A FOLKSONOMY-BASED INDEXING ALGORITHM

7.1. Introduction

Different knowledge organization systems (KOS) help in the support of sophisticated document indexing. Common examples of KOS include classification systems (taxonomies), thesauri, and controlled keywords (nomenclatures) [134].

Most prominent are approaches of document indexing (i.e., assigning content-descriptive keywords to documents) [135]. This enhances retrieval techniques and aids users in deciding on a document's relevance. During the last decade, a well-known problem of indexing documents with content-descriptive metadata has been addressed from a new, user-centred perspective. Within the so-called "Web 2.0", web users have begun publishing their own content on a large scale and started using social software to store and share documents, such as photos, videos or bookmarks. In addition, they have begun to index these documents with their own keywords to make them retrievable. In this context, the assigned keywords are called "*tags"*, the indexing process is called "*social tagging"*, and the totality of tags used within one platform is called *folksonomy [134]*.

Indeed, Social networks have become a popular medium for people to communicate and distribute ideas, content, news and advertisements [136]. Moreover, the rise of Web 2.0 technologies has led to a significant grow of annotations of digital content created by web users with different backgrounds and motivations and which is commonly referred to as User Generated Content (UGC). The mid-2000s have seen swift progress in

levels of interest in these kinds of techniques for generating descriptions of resources for the purposes of discovery, access, and retrieval [137].

Flickr[30], Delicious[31], digg[32] and last.fm[33] are just a few popular examples for WWW services that leverage this kind of annotations to make their content more accessible. This kind of UGC, which is also often referred to as social annotations, can be used in a variety of applications [136]: information visualization, for the creation of ontologies in the field of Semantic Web or the improvement of enterprise IR systems respectively web search.

Social networking sites have offered Internet users a novel way to organize their online digital content and share content with other users [136]. In general, users of social media sites contribute content, which is not restricted to one media type (e.g., documents, photos, URLs). Depending on the social media site, users can annotate content using descriptive text (e.g., title and description of photos in Flickr) or with metadata (i.e., tags) [137-138]. User-generated content mostly comprises of free, unstructured text, which often does not adhere to grammatical and syntactical rules, contains slag terms and abbreviations and is often of restricted length (e.g., 140 characters in Twitter[34]). To improve online content organization, categorization, search and filtering, users have adopted tags (or hashtags). The ability of users to select tags from an unrestricted vocabulary has led to the creation of personalized taxonomies, offering greater malleability and adaptability in information organization than formal classification systems, which impose users with

30 http://www.flickr.co

31 https://del.icio.us

32 http://digg.com

33 https://www.last.fm

34 https://twitter.com/

the restriction to annotate content based on predefined keywords [136, 139].

In current social tagging systems organization, classification and search tend to be rather simplistic in nature, often relying on keyword-based retrieval algorithms or aggregated results stemming from collaborative filtering techniques.

7.2. Knowledge Organization Systems

Knowledge representation methods are applied to provide a better basis for information retrieval tools. This may basically be done in two ways [134]: by abstracting the topics of a document and by indexing a document, i.e., assigning content-descriptive keywords or placing it into a concept scheme. For indexing documents with content-descriptive keywords, different types of knowledge organization systems (KOS) have been developed. The most important methods – classifications, thesauri and nomenclatures – comprise a controlled vocabulary, which is used for indexing. The vocabulary of classifications and thesauri usually has the form of a structured concept hierarchy, which may be enriched with further semantic relations, e.g., relations of equivalence and concept associations.

The two latest developments that have entered the spectrum of KOS are [134, 140]: folksonomies and ontologies. They complement traditional techniques in different ways. Folksonomies include novel social dimensions of user involvement; ontologies extend the possibilities of formal vocabulary structuring.

We may classify different KOS according to the complexity of their formal structure (mainly defined by the number of specified semantic relations in use for structuring the vocabulary) and the extent of the captured domain (Figure 7.1).

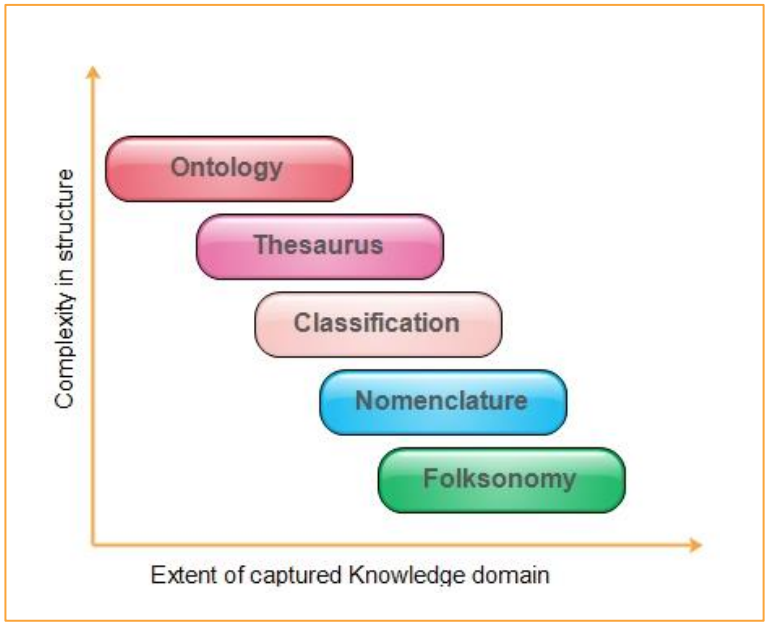


Figure 7-1: Classification of Knowledge Organization Systems [134].

Both aspects are inversely proportional: the more complex the structure, the smaller the captured domain will have to be, due to feasibility reasons. Folksonomy is a completely unstructured method of document indexing. While in most other cases trained indexers or other experts are responsible for indexing documents, folksonomies allow the producers or the users of certain content to take over this task. There is no authority, which controls the terminology in use. This also means that folksonomies are in no way limited to a certain domain of interest. They can be easily applied to all given contexts, as long as a community of interest exists.

7.3. Social Software

The key areas of Social Software are considered to be the weblogs, the wikis, and the social-network services of different kinds (Figure 7.2). Social network services range from some focused purely on networking, to others designed to share different types of resources, or meant for open coordination purposes. A strict classification is hard to derive, because the categories of Social Software tend to intertwine and to rely on each other [141].

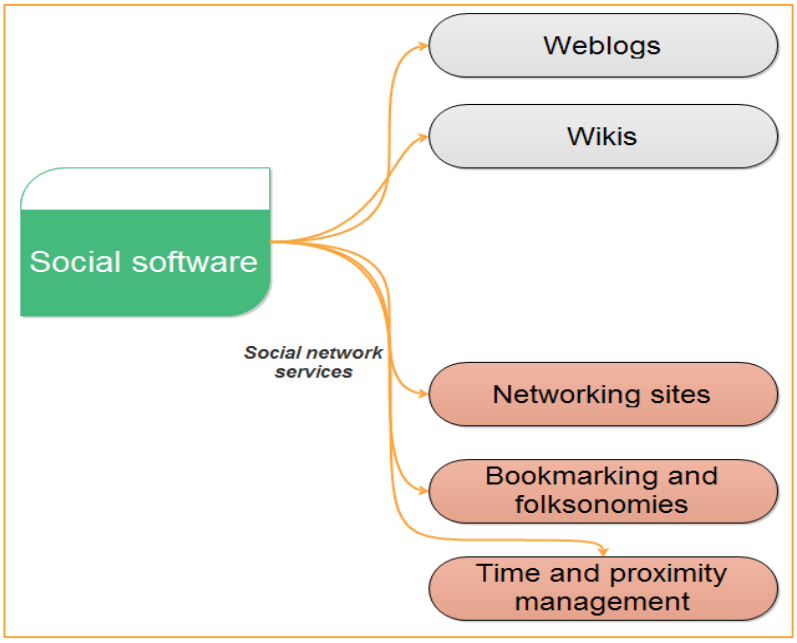


Figure 7-2: Key areas of social software [141].

7.4. Emergence of the Social Web

The first social networking services such as Classmates[35] (1995) and SixDegrees[36] (1996) transformed the structure of the Web from a hypertext environment that links data to a *"Web of people"* environment that connects family, friends and colleagues [142]. With the launch of the first blogging service OpenDiary[37] in 1998, Internet users were given the opportunity to publish their own content on the Web. They became able to interact with each other and post comments on published content. Interaction between users was promoted later by Wiki platforms, namely Wikipedia[38] (2001).

Such service enabled online communities, on the first hand, to exchange their knowledge, and on the other, to collaborate efficiently online [142]. The popularity of these websites is followed by the growth of

[35] http://www.classmates.com/

[36] http://www.sixdegrees.com/

[37] http://www.opendiary.com/

[38] http://www.wikipedia.org/

other social networking services such as Myspace[39] (2003), Facebook[40] (2004), LinkedIn[41] (2006) and Twitter[42] (2006).

These websites have not only instated a novel practice on the Web but also introduced a new life style for Internet generation. Social networking services have widely impacted communication, education, and entertainment as well as commercial, financial and governmental services [142].

With the exponential growth of the social Web, the role of Internet users has been transformed from passive information consumers to active producers. Thus, social networks and UGC could be integrated along retrieval processes as information source for relevance feedback and personalized access.

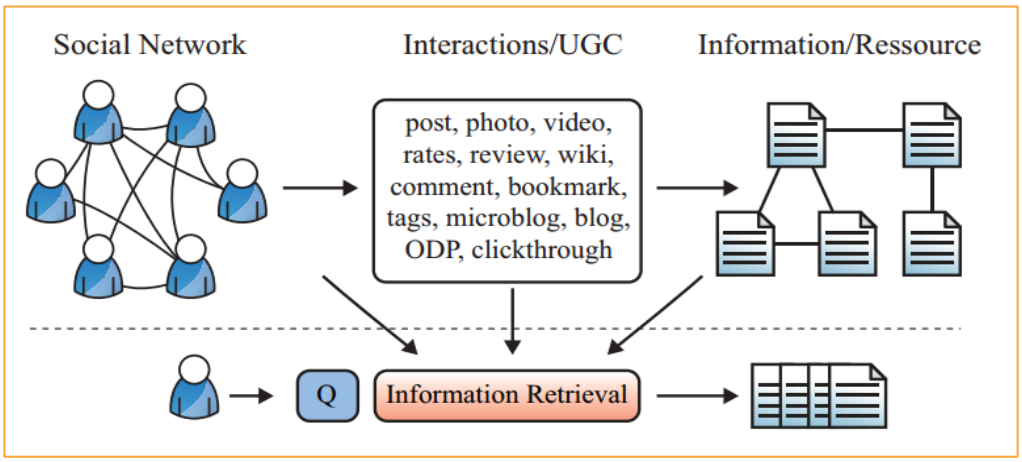


Figure 7-3: Using UGC to enhance information retrieval [142].

7.5. Social tagging

An annotation system open for users to apply subject headings is called “folksonomy”, the freely chosen subject headings are called “tags”.

[39] http://www.myspace.com/

[40] http://www.facebook.com/

[41] http://www.linkedin.com/

[42] http://www.twitter.com/

The process of indexing by means of folksonomies is named “(social) tagging” [143-144].

In contrast to the pre-defined categories and terms of a classification scheme, social tagging systems enable users to create and assign tags that meaningfully organize the content of a website. Aggregation of tags leads to the generation of a folksonomy, a socially owned vocabulary, whose terms define and organize the content of a website from the perspective of members of the user community rather than that of experts. Also known as collaborative tagging, it refers to assigning specific keywords or tags to items and sharing the set of tags between communities of users.

In short, tags can be considered as a meta-information on shared Internet resources. They are keywords generated by internet users on platforms that are used to describe and categorise an object, concept or idea. On some platforms, other users can also vote on tags that have already been added providing an additional social aspect to social tags.

Tagging of course is not a new concept, especially to librarians, indexers and classification professionals. What is new is that the tagging is being done by everyone, no longer by only a small group of experts, and that the tags are being made public and shared [139]. The development of the internet and the web, and of search engines, led to users doing their own searching. In the Web 2.0 environment users are now also doing their own content creation and information management. Tagging is used in a range of sites for many different types of resources. Tagging is done somewhat differently at different websites, but the all of the following example use some type of user tagging [139]:

- Blogs (Technorati: http://technorati.com/) ;
- Bookmarks (Delicious: http://del.icio.us/);

- Books (Librarything: http://www.librarything.com/, Amazon: http://www.amazon.com/);
- Emails (Gmail: http://mail.google.com/);
- Events (Going to meet: http://www.goingtomeet.com/);
- People (Tagalag: http://www.tagalag.com/);
- Pictures (Flickr: http://www.flickr.com/);
- Podcasts (Odeo: http://odeo.com/);
- Videos (YouTube: http://www.youtube.com/);
- Even perhaps tagging of tags? (http://tagtagger.com/).

The user chooses a tag that is meaningful to him or her. Once the tags have been assigned, they act as index terms and they may be public or private. When they are public, the tags together can all be searched by all users, creating a “folksonomy”.
It is important to remember that users have complete freedom in the tags they choose and may assign tags for their own organising purposes, without regard to any other users who may wish to make use of them. Even if this is the case, there may still be valuable information in the collection of tags that develops.

The aggregation of all the tags allows a site like Flickr to organise resources better for all users, and also informs the site owners about the popularity of tags and of resources. This can be described as a bottom-up rather than top-down building of categories [150].

Tags, once assigned, can be grouped, shared, displayed, published and managed in several ways. It is possible to see all tags assigned to a resource, all people who have used a particular tag, other tags that have been used for similar items, popular tags, recent tags etc.

The terminology about user tagging is still fairly fluid and many terms for the same phenomena are being used, often in slightly different ways, with debate starting about the exact usage and meaning of the terms

[151]. These terms currently include [137]: Collaborative tagging, shared tagging, user tagging, social bookmarking, collaborative bookmarking; folksonomies, tagsonomies, tagonomies, collabularies, tagosphere, folksonomic zeitgeist, social indexing, and collaborative indexing.

The idea of sharing bookmarks online goes back to ItList [141]. ItList was the first website to apply this idea in 1996! During the dot com bubble era numerous bookmark sharing websites appeared: Backflip, Blink, and Clip2. Nevertheless, these early attempts did not work out, which is often blamed on the burst of the bubble. Social bookmarking sites did not seriously become popular with the launch of Del.icio.us in 2003.
Delicious pioneered tagging and coined the term social bookmarking. In 2004, as Delicious began to take off, Furl and Simpy were released, along with Citeulike and Connotea (sometimes called social citation services). Other popular examples of social bookmarking sites: Diigo, Blue Dot, BookmarkSync, Cloudytags.com, Digg, GiveALink.org, Ma.gnolia, and My Web.

7.6. Definition of Folksonomies

A folksonomy is an indexing method open for users to apply freely chosen index terms. Peter Merholz entitles this method “metadata for the masses”; the writer James Surowiecki refers to it as one example of “the wisdom of crowds.” [134]. The term “folksonomy”, as a conflation of the worlds “folk” and “taxonomy” used to refer to an informal, organic assemblage of related terminology [148]. This term was introduced in 2004 by Thomas Vander Wal and cited in a blog post by Gene Smith to introduce what he called “user-generated classification, emerging through bottom-up consensus”[139]. Smith uses the term “classification” for paraphrasing folksonomies. This term arouses a misleading and faulty connotation. The same holds for the term “taxonomy.” Folksonomies are not classifications or taxonomies, since they work neither with notations nor with semantic relations [134, 143-144]. They are, however, a new type

of knowledge organization system, with its own advantages and disadvantages.

In folksonomies, we are confronted with three different aspects [143, 145]: (a) the documents to be described, (b) the tags (words), which are used for description, and (c) the users (prosumers), who are indexing.

Hence, a folksonomy can thus, be considered as a collection of tag assignments and *folksonomy systems* are those systems that allow for the evolution of folksonomies [145].

Today, there exist many diverse folksonomy systems in various domains. For example, Last.fm enables users to annotate music, bookmarks can be tagged in systems such as Delicious, BibSonomy supports social tagging of research articles, Amazon enables their customers to tag products, and Google Mail users can organize their emails via freely chosen labels.

Users might tag for different reasons [137, 145]:

- Future retrieval;
- Contribution and sharing;
- Attracting attention;
- Play and competition;
- Self-presentation; and
- Opinion-expression.

In short, Folksonomies “include everyone’s vocabulary and reflect everyone’s needs without cultural, social, or political bias [134,146].

There is a debate about the nature of these concepts and terms. Some writers have distinguished between a folksonomy (a collection of tags created by an individual for personal use) and a collabulary (a collective vocabulary). Other writers however use folksonomy to mean a collective vocabulary. Although private uses of this type of facility are also

valid, it is, we argue, only when tags are publicly shared that a folksonomy develops [139]. In this thesis, we use tagging to refer to labelling of web items, “user tagging” when that tagging is done by the user, and folksonomy to refer to the collection of users’ tags A key feature of a folksonomy is that tags may be reused many times, providing information about the popularity of the tags themselves (which synonyms come to be more popular over time) as well as information about emerging areas of interest.

Hence, folksonomies are characterized as user oriented, empowering, democratic, low-cost, dynamic and instructive [137]. Therefore, such user warrant based indexing processes are considered as alternative route to supplement and complement the roles of the information professionals in subject indexing and to facilitate information retrieval and knowledge organisation over the web.

7.6.1. Types of folksonomies

Folksonomies can be considered nowadays as user generated ontologies (or taxonomies), dynamic ontologies, or community-influenced ontologies [147].

A study of folksonomies helps uncover tags’ types [147]:

- *Content-based tags*: describe the content of an item (e.g., “bus”, “student”, “university”);
- *Context-based tags*: describe the context of an item (e.g., “Blida City”);
- *Attribute tags*: describe implicit attributes of an item (e.g., “black and white”, “homepage”);
- *Subjective tags*: describe an item subjectively (e.g., “pretty”, “amazing”, “extraordinary”);
- *Organizational tags*: helps to organize items (e.g., “to-do-list”, “my pictures”).

Abel [145] talks about eight main categories of tags: topic, time & location, type, author/owner, opinions/qualities, usage context, and self-reference.

Moreover, Vander Wal proposes to study social tagging systems by defining Broad (Figure 7.4) and Narrow (Figure7.5) Folksonomies [148]. “Broad” folksonomies (such as del.icio.us) result when many people tag the same item; analysing tags reveals a power law distribution, tapering to a 'long tail' of items in which only a few people exhibit interest. “Narrow” folksonomies (like Flickr) result when only one person (or a few people) tags an object, usually one they themselves created. In this narrow, individually-defined context it may be more difficult to determine meaning in relationships between tags, because language may be personal [148].

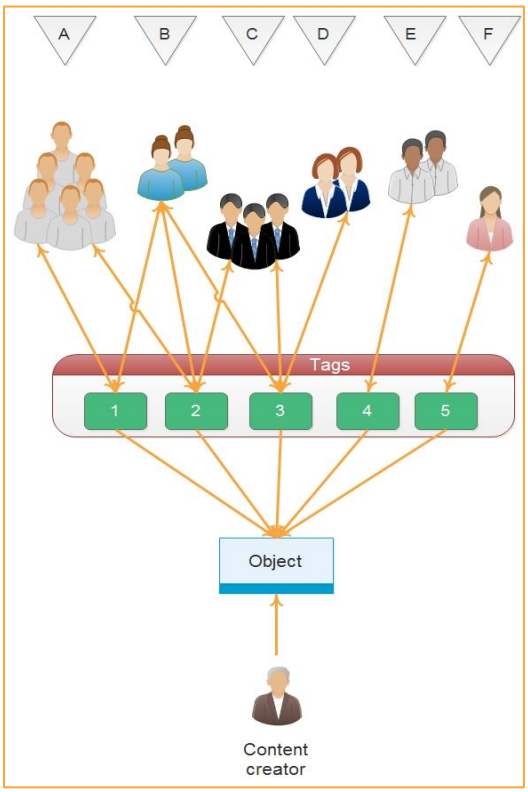


Figure 7-4: Folksonomy with multiple tag application (“broad”) [143].

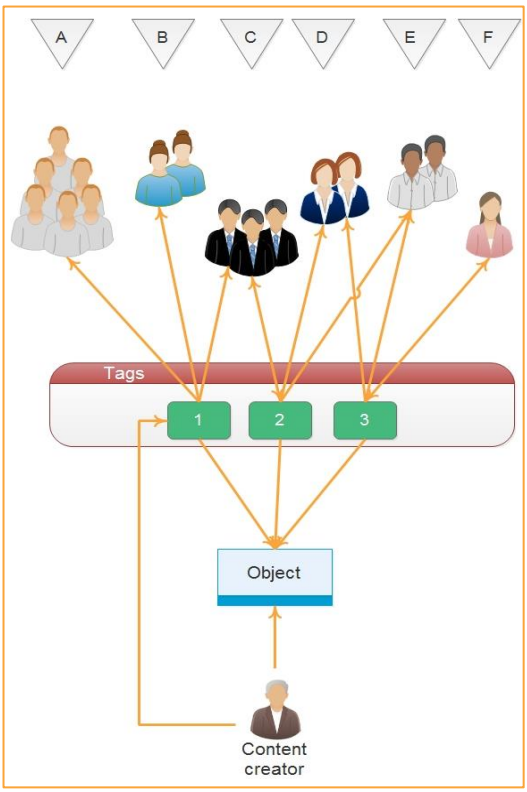


Figure 7-5: Folksonomy with single tag application (“narrow”) [143].

Finally, Peters et al. compared some aspects of tagging systems and corresponding “-onomies” [149]:

- Folksonomy (all tags from an information service);
- Personomy (all tags from a person);
- Docsonomy (all tags of a concrete document);
- Joursonomy (all tags from a concrete journ);
- Tweetonomy (hashtags in Twitter).

7.6.2. Folksonomies VS Formal taxonomies

It is worth reviewing some features of folksonomies and comparing them to formal classification systems [137, 134, and 145].

Table 7-1: Comparison between Folksonomies and Taxnomies.

	Folksonomies	Taxonomies
Pros	- Inclusiveness of vocabularies of community users - Currency of descriptors - A low-cost device in implementation and reuse - More browsable ressources	- Increased precision - Professionally assisted - Elaborated knowledge representation techniques
Cons	- Lack the preciseness in information retrieval - No control over the vocabulary - Lack of hierarchy	- Systematic set of metadata (controlled vocabulary) - Need of expert skills in indexing

7.6.3. Characteristics of Folksonomies

Weller et al., [134] introduced some key aspects of a critical reflection on folksonomies: (a) the confrontation of user’s language versus vocabulary control; (b) the social and personal objectives in tagging behaviour; and (c) the contrast between retrieval and exploration. Table 7.2 summarizes the main benefits and problems with folksonomies [134, 146, and 150].

Table 7-2: Main benefits and problems of Folksonomies.

Benefits	Problems
- Represent an authentic use of language; - Allow multiple interpretations; - Recognise neologisms; - Are cheap methods of indexing; - Are the only way to index mass information on the web; - Give the quality control to the masses; - Allow searching and better browsing; - Can help to identify communities; - Are sources for collaborative recommender systems; - Are sources for the development of ontologies, thesauri, or classification systems; - Freedom and flexibility in tag choice and use (timeliness and multiple perspectives); - Make people sensitive to information indexing issues.	- Have no vocabulary control and do not recognise synonyms and homonyms (the well-known "vocabulary problem"; - Synonyms, trans-language synonyms, homonyms, polysems, spelling variants and abbreviations are not distinguished; - Do not make use of semantic relations between tags; - Mix up different basic levels; - Merge different languages; - Do not distinguish formal from content-descriptive tags; - Include spam-tags; - User-specific tags and other misleading keywords.

7.6.3.1. Benefits of folksonomies

Because many social bookmarking sites display recently added lists and popular links, they allow to both keep up with what is current and see relevant information. So, what started out as a way to send bookmarks to friends has really grown into social search engines. It is no longer required to page through thousands of results to find something that real humans would recommend enough to save for themselves and share with others.

Tagging reflects the prosumers' conceptual model of information and tags authentically represent the language of authors and users. This sort of indexing leads to "multiple interpretations", different (and sometimes disparate) opinions and "multicultural views" of the same piece of information. "Shared inter-subjectivities" enable the users "to benefit, not just from their own discoveries, but from those of others" [143].

The development and updating of controlled vocabularies can profit from folksonomies, because the tags, their frequency and their distribution are sources for new controlled terms, for modifications of terms and perhaps for deleting concepts in the sense of a “bottom-up categorization”. In this way tags guarantee a fast response to changes and innovations in the knowledge domain [143].

A study analysing the structure of collaborative tagging systems found “regularities in user activity, tag frequencies, kinds of tags used, bursts of popularity in bookmarking and a remarkable stability in the relative proportions of tags within a given url.”

The following are characteristics of tagging and folksonomies that can be seen as beneficial features [141, 143, and 148]:

- Browsing other people’s bookmarks can save hours of work and is an effective alternative to Google and catalogue searches;
- Social bookmarking gives users the opportunity to express differing perspectives on information and resources through informal organizational structures. Thus, create new communities of like-minded individuals;
- Tags are easier to understand than more standardized or top-down indexing terms.
- Human beings understand the content of the resource, as opposed to software, which algorithmically attempts to determine the meaning of a resource;
- People can find and bookmark web pages that have not yet been noticed or indexed by web spiders;
- They are multidimensional: users can assign a large number of tags to express a concept and can combine them;
- Users can use their own language: words that have meaning for them. These words are likely to be current and reflect local usage;

- Tags can be shared, creating knowledge through aggregation. Millions and millions of people are saying, in public, what they think pages and images are about. That is crucial information that can be used to pull together new ideas and information across the endless sea that is the internet;
- Instead of having to store an item in a single folder, it can be tagged with many different terms and each of these could be used to generate an instant collection (e.g. if a website is bookmarked with tags such as wedding, family, holiday, Europe, sub-collections can be readily assembled by searching for single tags or pairs);
- Public tagging has been described as having an altruistic appeal, allowing people to contribute to a shared knowledge base. Social tagging fosters the development of communities around similar interests and viewpoints;
- Social tagging provides insight into users' information needs and habits to professional providers and managers. Thus, highlighting areas of interest and how they are being described;
- Tagging is very quick, simple and straightforward. Users can apply tags without formal training in classification or indexing;
- User and Time are core attributes: because we can derive information such as "this is who", "this link was tagged by" and "this is when it was tagged". Inclusion and exclusion around people and time can be performed, not just tags.

7.6.3.2. Disadvantages of folksonomies

In folksonomies we find different word forms, nouns in singular ("library"), nouns in plural ("libraries") and abbreviations ("IA" or "IT"). Sometimes, users create phrases by leaving out blanks between single words ("informationscience") or by combining words with an underscore ("information_science"), which lead to the introduction of new Natural Language Processing criteria. There is no control of synonymy and homonymy, there are many formats for dates and a lot of typing and orthographic errors. About 40% of tags are either, misspelt or compound words consisting of more than two words or a mixture of languages [143].

The prosumers, who tag documents, act in different contexts, have different tasks and different motivations. One user tags a document from her or his work-related view; another tags it by keeping the aspect of vacation in mind. We have to consider (particular for Del.icio.us, probably not for Technorati, Flickr or YouTube) that “a significant amount of tagging, if not all, is done for personal use rather than public benefit“.

Of course, we find tags identifying what a document is about. Nevertheless, and this is the problem, we find tags identifying the owner of the document or a formal description (e.g. “cooking”) as well. Besides, some tags do not describe the document, but give a judgment (“stupid“, “good”, or “useless”). Therefore, those kinds of tags are virtually meaningless to anybody except their creators [143].

It will be readily apparent that many of the features of folksonomies listed above as advantages can also lead to problems for effective classification and information management. Indeed, the simplicity and ease of use of tagging can result in poorly chosen and applied tags. While it could be argued that this is a necessary feature of user tagging and insignificant, nevertheless, the following issues need to be considered [137, 141, 143, 145, 148, 152-153]:

- The major drawback is the lack of standardization. There is no controlled vocabulary that is a list of standard keywords. So, many errors can occur due to misspelling, synonym confusion, tags with more than one meaning, or tags that are too personalized;
- Social bookmarking is done by amateurs. There is no oversight as to how resources are organized and tagged. This can lead to inconsistent or otherwise poor use of tags;
- Because social bookmarking reflects the values of the community of users, there is a risk of presenting a skewed view of the value of any particular topic. For example, users may assign pejorative tags to certain resources;

- Tag descriptions present some drawbacks such as tag scarcity or concept inconsistencies;
- A disadvantage of today's folksonomy systems is that they are designed for humans and do not comply with the vision of the Semantic Web;
- Tags can be applied at different levels of specificity by different users (or even by the same user at different times) e.g. the tag cats may be used in one case and animals or pets in another. Or the tag Kitty may simply be used. Different terms may be used for the same concept (again by different users or by the same user – users will not necessarily be consistent). So felines may be used for some items and cats for others. A person searching for articles about cats will have to use many different terms to be sure of finding all items;
- Tags with personal meaning only are frequently used (example on Flickr: viewfrommywindow). This tag on its own is of virtually no use to anyone else. Conversely, the same term can be used for different concepts. Typically, no information about the meaning of a tag is provided (although some systems, del.icio.us being one, do allow tag descriptions). The word play could occur in an educational resource collection in the drama context or the games context. The word tag itself has more than one meaning. Without even considering the issue of other languages, English itself has a huge number of words with multiple meanings;
- Uncontrolled tagging can result in a mixture of types of things, names of things, genders and formats. Many of these problems can arise even with specialist indexers, for example using video as a subject heading when the item is a video, when it should only have that subject heading if it is about videos. If it is already difficult for people to comply with requirements such as these, it will be far more difficult to have precision when there are no indexing guidelines other than those developed by individual users for their own practice and unlikely to be made explicit. Moreover, regular indexing and cataloguing rules such as singular vs plural forms, use of hyphens and spelling conventions are not established in a folksonomy;

- People's choice of tags may change as new trends evolve — e.g., it is likely that blog, weblog, blogs and blogging will all be used for the same concept. Many systems only allow single word tags. It may be difficult to assign terms to complex concepts using only a single word and running two or more words together is difficult in many ways – the resulting words will be highly idiosyncratic and difficult to read and to search with precision;
- Moreover, datasets like CiteUlike, or Connotea are incomplete (as they do not represent all users or the entire tag vocabulary or tagging activity of any particular user);
- Social tagging systems are vulnerable to spam and malicious practice;
- A more subtle issue is that people may behave differently (consciously or unconsciously) when tagging other people's items as opposed to their own. The objectivity of a professional indexer is not necessarily a feature of social tagging;
- Another high-level concern is that over time tags may come to represent a dominant view, discouraging usage of less popular concepts (and terminology). Users will tend to use popular tags and may not realise that there is a more precise term available for their concept.

7.6.3.3. Discussion

The value and contribution of social tagging and folksonomy have not yet been fully established. This is, perhaps why the creation and application of tags by users who are not experts in information management led to the problems described above. However there are also clearly great benefits in user tagging and folksonomies, especially in the richness, currency, relevance and diversity of the terms used, and the collections of resources created. It is important to try to retain those qualities in any attempt to control folksonomies.

Furthermore, tagging is not about accuracy, authority, and not about right descriptors or wrong descriptors, but about recalling, user warrant

and user acceptance based on users' needs [137]. Hence, librarians must think of using both social tags and traditional information organisation systems like controlled vocabularies and use it simultaneously to complement and supplement information retrieval.

7.7. The turtledove Indexing technique

Success with keyword access to textual sources (as in Google searches) has led to an exploration of alternative methods of generating access points in the field of textual indexing. These could be within the resource itself (e.g. title words compared to subject headings assigned by cataloguers or author keywords compared to professional index terms). All these studies point to the possibility to enhance professional indexing with materials from other sources, issues that are explored in depth in the text mining and natural language processing literature.

The rapidly developing literature clusters into three broad approaches to the study of social tagging and folksonomy, focused first, on the *folksonomy* itself (and the role of tags in indexing and retrieval); secondly, on *tagging* (and the behaviour of users); and thirdly, on the nature of *social tagging systems* (as socio-technical frameworks). In this thesis, we focused more in the vocabulary that results from social tagging rather than tagging as an activity.

There are basically three different approaches aiming to solve the present problems of folksonomies. All approaches complement each other:

- First, one can focus on the actors and try to educate users to improve “tag literacy”;
- The second approach comprises combinations of social tagging with other knowledge organization systems; and finally

- Finally, we may generally consider tags as elements of natural language and treat them by means of automatic methods of natural language processing (NLP) for better retrieval results.

Moreover, there are three groups of actors who are able to index documents: authors, professional indexers and users. Probably all of them may use different ways for indexing and along with it may focus on different characteristics. Text-oriented methods make use of the author's language, e.g. in forms of indexing titles, abstracts or references. In contrast to text-orientated methods, folksonomies do not only represent the producer's view, but the views of the consumers as well. Ontologies and other tools of controlled vocabularies (like thesauri or classification systems) are in need of interpreters: (a) experts who create such vocabularies and (b) other experts who are able to use the controlled terms in order to index the documents. Ontology-creating interpreters have to analyse, literature, needs, actors, tasks, domain, activities, etc."– Undoubtedly a time-consuming procedure.

Tagging has dramatically lower costs because there is no complicated, hierarchically organized nomenclature to learn. Users simply create and apply tags on the fly. The low costs of user tagging ground the economically deterministic view that folksonomy will prevail over structured metadata [148].

In short, Folksonomies - collections of user-contributed tags, proved to be efficient in reducing the inherent semantic gap. However, user tags are noisy; thus, they need to be processed before they can be used by further applications. Indeed, the problems encountered by the use of Folksonomies may be avoided by the application of vocabulary control techniques.

In this thesis, we propose an approach for bootstrapping semantics from folksonomy tags and providing an IRS with the unified power of ontologies and folksonomies.

7.7.1. Primary Tests

The online survey we have conducted clearly shows that the limit between ‘Personal preferences’ and ‘Social network preferences’ are vanishing. That is to say people do take into account the view of their (physical and virtual) social network proportionally to their own ‘Personal preferences’. In this regard, we tried to explore the use of folksonomies and evaluate their impact on the relevance of the proposed IRS. Especially, in terms of accuracy and index size reduction.

Indeed, folksonomies can provide ambiguous yet contextualized indexers (called tags) for a given web resource. Thus, we enriched the SECAS algorithm, by subjecting the obtained indexes to a filtering phase based on a bag of meaningful social-tags. But, before talking about the turtle dove indexing technique, we present some prior tests, by briefly describing the dataset used for the evaluation.

7.7.2. The SocialBM0311

SocialBM0311[43] is a large-scale social tagging/bookmarking dataset (11GB) collected from Delicious.com. It contains the complete bookmarking activity for almost 2 million users from the launch of the social bookmarking website in 2003 to the end of March 2011. The dataset contains:

- 339,897,227 bookmarks;
- 118,520,382 unique URLs;
- 14,723,731 unique tags;
- 1,951,207 users.

[43] http://www.zubiaga.org/datasets/socialbm0311/

The files contain one bookmark per line. Each bookmark is represented by a vector containing the following fields separated by tabs: url_md5, user_id, url, unix_timestamp, tags. Where:

- 'url_md5' is the MD5 hash of the bookmarked URL. Note that Delicious uses the MD5 hash as the ID for URLs, and can be used to find it through http://www.delicious.com/url/url_md5;
- 'user_id' is the ID for the user who saved the bookmark. The usernames have been fully anonymized for this dataset, and the user IDs provided with the dataset have been randomly assigned to users;
- 'url' is the URL being bookmarked;
- 'unix_timestamp' refers to the date in which the bookmark was saved, using the standard UNIX time format. Note that these timestamps are rounded to days, and do not provide the specific time,
- 'tags' include a tab-separated list of the tags (keywords) used in the bookmark.

The socialBM dataset was used not only because of the contextual information it contains, but also because of the richness of the corpus. Effectively, each resource can provide many contextual information that help us in the modeling of our Context-Aware Information Retrieval Model.

7.7.3. Data cleansing

In the beginning, we choose randomly 10000 lines from the dataset, which correspond to 10000 web pages Then, we removed the broken links (only valid links were selected so that the documents can be easily indexed through NLP techniques), meaningless or erroneous tags (only meaningful tags were retained so that our conceptual indexing algorithm can be used properly), recurring tags (only unique and relevant tags were retained), and tags in different languages than English. Moreover, we selected only closely related web categories (Psychological science

resources). Finally, we ended up with 3000 lines, which were grouped according to their urls.

7.7.4. Evaluation Results

In this section, we present the performance measurements of our methods and algorithms.

First, we wanted to compare the relevance of tags vis-à-vis metadata and the indexers obtained by the SECAS algorithm. Thus, we computed the query-document matching measures in three different cases:

- Documents are represented by a forward index;
- Documents are represented by a set of metadata (extracted by the Meta Tags Extractor[44]);
- Documents are represented by a set of tags (social bookmarks).

The aim was to draw correlations among them and to develop the turtledove-indexing algorithm. The obtained precision values are summarized in Table 7.3.

Table 7-3: The Turtle dove matching technique's evaluation results.

		TP	FP	Precision
SECAS Indexers	Precision	63	33	0.66
	P@5	82	14	0.85
	P@10	86	10	0.90
	P@20	93	3	0.97
Metadata	Precision	51	45	0.53
	P@5	61	35	0.64
	P@10	64	32	0.67
	P@20	67	29	0.70
Tags	Precision	49	47	0.51
	P@5	64	32	0.67
	P@10	68	28	0.71
	p@20	74	22	0.77

[44] http://www.webtoolhub.com/tn561365-meta-tags-extractor.aspx

We can notice that the indexers obtained by SECAS outperform metadata and tags. Moreover, we observed that the precision at 5, 10, and 20 was better in the case of tags than of metadata. Because, recurring metadata can be found among the datasets (same content creator). Contrariwise, there was a variety in the use of tags.

Furthermore, the indexing time obtained while using tags was considerably short in the case of folksonomies than of metadata (tale 7.4).

Table 7-4: Indexing time comparisong SECAS/ Folksonmies/ Metadata.

	SECAS	Tags	Metadata
Indexing time (millisecond)	2931768	43097	154520

This is why we thought about the use of the tags, in the one hand to reduce the size of the forward indexes without compromising the effectiveness of the results, and in the other hand, to provide indexers that are more representative about the user's context.

Thus, the turtledove-indexing algorithm result in a two phased algorithm (figure 7.6):

- First tags and indexers are extracted,
- Then, we use our similarity measure to restrict the number of the former indexers by retaining only those, which are closely similar to the user's tags.

The tags/indexers similarity measurement (turtledove indexing technique) algorithm resembles to the query-document matching algorithm (Algorithm 5), with the tiny difference that in this case, we deal with one document at a time and we retain the indexers not the documents.

We observed that the precision was enhanced up to 88% and the recall up to 75%. Moreover, the size of index files was reduced up to 42% as compared with previous ones, which will have a considerable impact in the reduction of the query-document mapping time.

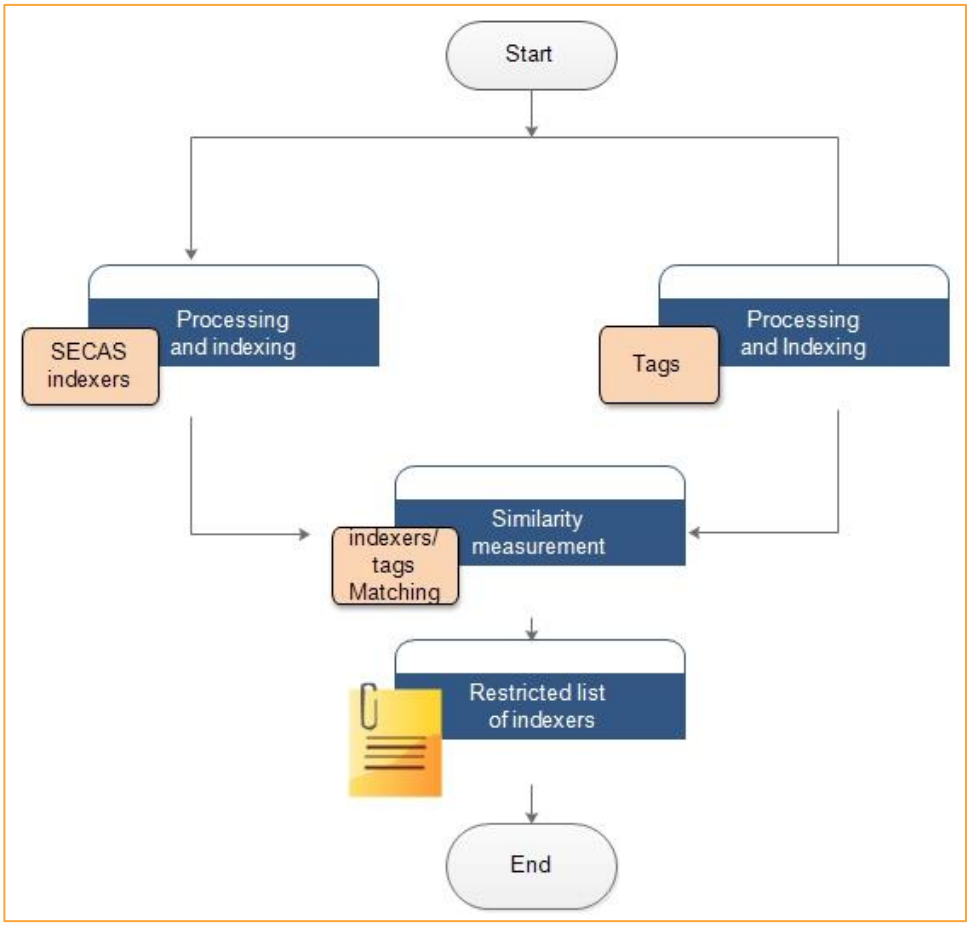


Figure 7-6: The turtledove indexing algorithm.

Unfortunately, the socialBM datasets does not provide queries to perform further tests. Consequently, our current work focus on the finding of more complete datasets to evaluate our final version of the context-aware IRS.

7.8. Conclusion

In this chapter, we have introduced the notions of social tagging and presented our contribution to a Folksonomy-based indexing technique. The obtained results are very encouraging and motivate us to explore the use of Folksonomies as a complement to benefit, wisely, from the ever-increasing power of the masses powered by the use of internet.

GENERAL CONCLUSION

1. Conclusion

Over the last decades, there have been remarkable shifts in the area of Information Retrieval (IR) as huge amount of information is increasingly accumulated on the Web. The gigantic information explosion increases the need for discovering new tools that retrieve meaningful knowledge from various complex information sources.

A key characteristic of traditional Information Retrieval Systems (IRS) is that the degree of document-query matching depended only on the number of shared keywords, which led to a “lexical focused” relevance estimation dismissing completely the context dimension of the user. Indeed, in such IRS, relevant documents were not retrieved if they did not share words with the query, and irrelevant documents that had common words with the query were retrieved even if these words had not the same meaning in the document and the query.

In this context, Semantic search was introduced trying to improve search by understanding the contextual meaning of the terms. In addition to the semantic problem, IR does encounters many issues that have been highlighted in this thesis. We have seen, that the inclusion of a contextual dimension may solve some of the inherent issues. Thus, some valuable work were studied in order to form a categorisation of context dimensions in IR, then a survey was conducted to understand the new search habits and the most important context-dimensions to take into consideration.

We retained the inclination of users towards: (a) social network preferences proportionally to their own personal preferences, also (b) users concern about accuracy and time, and finally (c) nature of queries which became shorter and thus more ambiguous.

In this thesis, we provided a new perspective to address IR problems with the inclusion of two contextual dimension: Social dimension and content dimension. In particular, we focused on proposing new models to index documents and queries, to compute the similarity between them and finally to propose the most relevant possible results.

First, the Porter and CAS stemmers were studied with the aim to propose a new hybrid stemming method for Information Retrieval purposes. The main advantage of our method over existing methods is that it provides an accurate stemmer with meaningful stems in 99% of cases. The stemming algorithm combines features from algorithmic stemmers and dictionary stemmers aiming to maximizing the proportion of the meaningful stems, without compromising the other performance measures. Indeed, the new hybrid stemming method is based on a combination of affix stripping (based on Porter Stemming algorithm), context-aware techniques (based on the Context-Aware Stemming “CAS” algorithm), and corpus based techniques for English language (based on Wordnet).

Besides the fact that SECAS reduces the size of index files as much as 60%, it also enhances recall and precision as compared with *Porter* and *CAS* algorithms in an evaluation with the *WT2G* dataset.

Then, we explored the use of folksonomies as a new alternative to ontologies in knowledge representation. This choice was motivated by the fact that Folksonomies give the quality control to the masses, allowing them to provide more accurate descriptors for web resources in a more flexible way. Indeed, social bookmarking is a promising technology, which appears highly relevant to today’s knowledge work, and is interesting because of the role it plays in knowledge sharing.

An evaluation with the socialBM dataset showed that in some cases, tags can provide a higher expressive power than metadata with a precision equal to 77% and 70%, respectively. Moreover, we observed

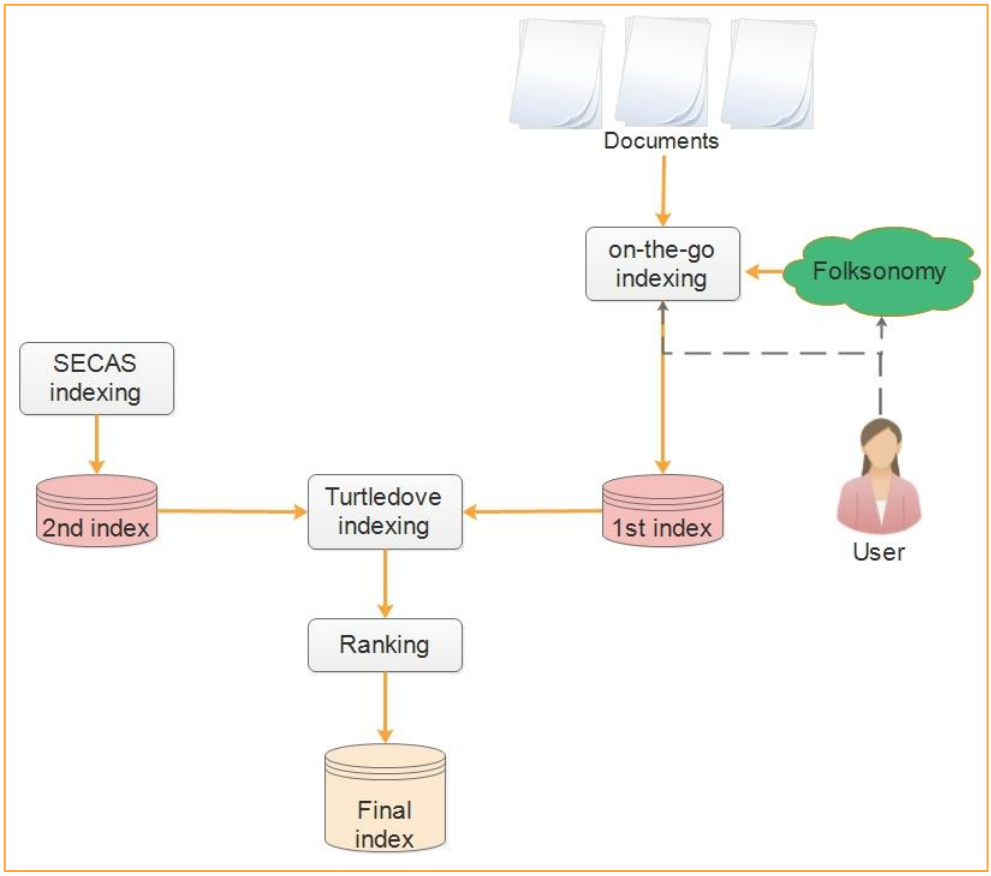
that the turtledove indexing technique, based on the reduction of the index size (up to 42%) by retaining only context-sensitive indexers helped to enhance even more the precision up to 88% and the recall up to 75% as compared with the previous version of the SECAS algorithm.

We think that this will result in a considerable impact on the reduction of the query-document mapping time. But, suitable datasets have to be found so as to prove our statements.

We note that despite the fact that most Web 2.0 services rely heavily on folksonomies for describing user-generated content and users seem to like using them, professional services like libraries or information service suppliers still often hesitate to let users tag their content on their platforms. Reasons for this reluctance on the professionals' side can be looked at as fear of loss of control. Thus, we may wonder if folksonomies are here to stay.

As a matter of fact Folksonomies present a valuable addition to the spectrum of knowledge representation methods. They appear in the context of user collaboration in Web 2.0 environment and provide easy and comprehensive access to large data collections. With web users taking control over document indexing, folksonomies offer an inexpensive way of processing large data sets. User-centred approaches to tagging have multiple benefits, as they can actively capture the authentic language of the user, are flexible and allow new ways of social navigation within document collections. Yet some problems derive from the unstructured nature of tags which may be solved by improving the users' tag literacy, by (automatic) query refinements, or by processing tags through natural language processing. We believe In the future, the advantages and shortcomings of folksonomies will be considered more closely as advanced approaches to the use of social tagging applications are emerging.

Our current work focus on the proposition of an on-the-go indexing technique. Where, web content can be indexable (and thus searchable), right after it is published. Evaluation tests are still in progress, but the basic idea can be summarized in the following figure.



2. Future work

Indeed, Folksonomies and traditional knowledge representation methods are not to be viewed as rivalling systems. But, as long as tagging is performed by single users within their personal workspace, the social component is lacking and we cannot speak of folksonomies in a strict sense, but of *personomies* and this constitute the core of our future work.

A Personmy represents all the annotations of a user in the context of a folksonomy. It also has a broader sense of a cluster of information a user associates with on the web from text, images, video annotations to blog posts…

Nowadays the Web is omnipresent, reaching into almost everyone's life. More and more Web users do not switch off their devices all the time, continuously receiving and sending messages, frequently looking for information, now and then evaluating this information, and so on. The means to reach the Web do thereby not stop at personal computers, but increasingly also include mobile devices. More and more users are sharing information online, are working collaboratively on a topic, as well as maintaining their relationship in the Web. All of this is so pervasive that it feels absolutely natural. Consequently it is not surprising that topics related to the Social Web are experiencing a surge of interest, both from the scientific community as well as the industry.

For future work, one main direction is to investigate whether it is worthwhile and how to use the prototype systems described in this thesis so as to focus even more on the user dimension by deepening the study about social search. This will necessarily involve several key things:

- We need to carefully study users' characteristics and behaviours in the case of social search so as to model their profiles including as many context factors as possible;
- Then, we need to find appropriate datasets to test our turtledove indexing technique and our on-the-go indexing technique and investigate the question whether they can be suitable to index content other than text. Indeed, knowing 95% of the information available on the web is of textual nature and although a picture is worth a thousand words and a video worth a thousand images. We believe that without text, no image, nor video be annotated, indexed, and found;
- Finally, if those results are satisfying, we would like to consider the proposition of a similar context-aware Information Retrieval System for other languages than English, the Arabic language in our case.

GLOSSARY

Browsing	: Unstructured exploration of a body of information.
Collection or corpus	: A set of documents
Document	: An information entity that the user wants to retrieve
Fielded searching	: Methods that search on specific bibliographic or structural fields, such as author or heading.
Folksonomy	: The totality of tags used within one platform.
Full text searching	: Methods that compare the query with every word in the text, without distinguishing the function of the various words.
Index	: A representation of information that makes querying easier
Information retrieval	: Subfield of computer science that deals with automated retrieval of documents (especially text) based on their content and context.
Linking	: Moving from one item to another following links, such as citations, references, etc.
Ontology	: Ontology defines the terms used to describe and represent an area of knowledge.
Query	: A string of text, describing the information that the user is seeking. Each word of the query is called a search term. A query can be a single search term, a string of terms, a phrase in natural language, or a stylized expression using special symbols.
Searching	: Seeking for specific information within a body of information. The result of a search is a set of hits.
Semantic search	: Semantic search consists to improve search by understanding the contextual meaning of the terms and tries to provide the most accurate answer from a given document repository.
Social tagging	: The indexing process in the context of social web.
Tag:	: Keywords assigned to resources in the context of social web.
Term	: A word or concept that appears in a document or a query

LIST OF SYMBOLS AND ABBREVIATIONS

AI : Artificial Inteligence
ASK : Anomalous State of Knowledge
BIR : Boolean Information Retrieval
BOW : Bag of Words
CAS : Context Aware Stemmer
CIR : Context-aware Information Retrieval
IDF : Inverse Document Frequency
IR : Information Retrieval
IRS : Information Retrieval System
KOS : Knowledge Organisation System
LCA : Lowest Common Ancestor
MAP : Mean Average Precision
MOnSE : Mergind ONtologies by Semantic Enrichment
NIST : National Institute of Standards and Technology
NLP : Natural Language Processing
OWL : Ontology Web Language
POS : Part Of Speech
PRM : Probabilistic Retrieval Model
PRP : Probability Ranking Principle
QA : Question Answering
RDF : Resource Description Framework
RDFS : RDF Schema
SECAS : Semantically Enriched Context-Aware Stemming Algorithm
TF : Term Frequency
TREC : Text REtrieval Conference
UGC : User Generated Content
DRAPA : Defense Advanced Research Project Agency
URI : Universal Ressource Identifier
URL : Unified Ressource Locator
VSM : Vector Space Model
W3C : World Wide Web Consortium
WSD : Word Sense Disambiguation
WWW : World Wide Web
XML : Extensible Markup Language

REFERENCES

1. G.N.K, Suresh Babu and Sankar. K (2013). A Study of Text Mining For Web Information Retrieval System From Textual Databases. Volume 3, Issue 12**:** 669-673.
2. Aruleba, K. D., et al. (2016). A Full Text Retrieval System in a Digital Library Environment. Intelligent Information Management: 1-8.
3. Mezzi, M. and N. Benblidia (2015). Aspects of Context in Daily Search Activities - Survey about Nowadays Search Habits. International Conference on Web Information Systems and Technologies, Lisbon, Portugal, SCITEPRESS (Science and Technology Publications, Lda.).
4. Crestani, F. and G. Pasi (1999). Soft Information Retrieval: Applications of Fuzzy Set Theory and Neural Networks. Neuro-fuzzy Techniques for Intelligent Information Systems.
5. Manning, C. D., et al. (2009). An Introduction to Information Retrieval. Cambridge, England.
6. Grossman, D. A. and O. Frieder (2004). Information Retrieval: Algorithms and Heuristics, Springer.
7. Zhu, D. (2014). Information Retrieval for Reducing Manual Effort In Biomedical and clinical Research. Computer Science. United States University of Delaware. Doctor of Phylosophy.
8. Lee, S. (2016). Multi Domain Semantic Information Retrieval Based on Topic Model, Georgia State University.
9. Roshdi, A. and A. Roohparvar (2015). "Review: Information Retrieval Techniques and Applications." International Journal of Computer Networks and Communications Security 3(9): 373-377.
10. Hofmann, K. (2013). Fast and Reliable Online Learning to Rank for Information Retrieval. The Netherlands, Universiteit van Amsterdam.
11. BASKAYA, F. (2014). Simulating Search Sessions in Interactive Information Retrieval Evaluation. Finland, University of Tampere.

12. Bondi, L. (2016) Text-Processing. Material for Information Retrieval Part. Accessed in June 2018, http://home.deib.polimi.it/lbondi/index.html.
13. Zhao, H. (2015). The Role of Document Structure and Citation Analysis in Literature, Drexel University.
14. Zhao, J. (2015). Term Association Modelling in Information Retrieval. Computer Science and Engineering. TORONTO, ONTARIO, YORK UNIVERSITY.
15. Nadji, N. (2013). Information Retrieval of Digitized Medieval Manuscripts. Institut d'Informatique. Suisse, Université de Neuchâtel.
16. Han, J., et al. (2010). Research of cognitive and user-oriented information retrieval. 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010), Chengdu, China, Institute of Electrical and Electronics Engineers (IEEE).
17. Saracevic, T. (2010). The notion of context in "Information Interaction in Context". The Information Interaction in Context Symposium, Rutgers University in New Brunswick, NJ, USA.
18. Daoud, M., et al. (2009). Contextual Query Classification For Personalizing Informational Search (regular paper).
19. Mirceska, A., et al. (2010). Location based systems for retrieval using mobile devices. Information and Communication Technologies (ICT Innovations 2010), Macedonia, Communications in Computer and Information Science.
20. Fujita, E. and K. Oyama (2011). Efficient Top-k Document Retrieval Using a Term-Document Binary Matrix. Asia Information Retrieval Symposium, Springer, Berlin, Heidelberg. 7097: 293-302.
21. MyGenShare (2012) How to Search the Internet like a Genealogist. Accessed in June, 2018, http://genealogybybarry.com.
22. Miwa, M. (2015). The Past, Present and Future of Information Retrieval: Toward a User-Centered Orientation. 12th EASTICA General conference and Seminar "Archives in the Digital Era: Revisited,". Japan.
23. Ruthven, I. (2008). Interactive Information Retrieval. Annual Review of Information Science and Technology. 42: 43-92.

24. Ingwersen, P. and K. Jarvelin (2005). Information Retrieval in Context – IRiX. ACM SIGIR Forum, ACM New York. 39: 31-39.
25. Saini, B., et al. (2014). "Information Retrieval Models and Searching Methodologies: Survey." International Journal of Advance Foundation and Research in Science & Engineering 1(2): 57-62.
26. Belkin, N. J. and W. B. Croft (1992). "Information filtering and information retrieval: two sides of the same coin?" Commun. ACM 35(12): 29-38.
27. Ughetto, L., et al. (2011). Différentes interprétations d'un modèle de RI à base d'inclusion graduelle**:** 295-310.
28. Salton, G. (1969). Evaluation Problems in Interactive Information Retrieval, Cornell University - USA.
29. Maxwell, T. (2014). Term Selection in Information Retrieval. Institute for Communicating and Collaborative Systems, University of Edinburgh.
30. Caragea, C. (2016) Web Search and Information Retrieval.
31. Wachsmuth, H. (2015). Text Analysis Pipelines: Towards Ad-hoc Large-Scale Text Mining, Springer International Publishing.
32. Guezouli, L. and H. Essafi (2016). "CAS-based information retrieval in semi-structured documents: CASISS model." Journal of Innovation in Digital Ecosystem: 1 5 5 – 1 6 2.
33. Neves, M. (2015) Information Retrieval. Natural Language Processing. In the IEEE-10th Systems of Systems Engineering Conference, Hasso Plantner Institut, IT Systems Engineering, Universitat Potsdam.
34. Cleverdon, C. W. (1962). Aslib Cranfield research project: report on the testing and analysis of an investigation into the comparative efficiency of indexing systems, Michigan University: 95-107.
35. Joe Buzzanga, M. (2015). "Beyond Keywords: The Revolution in Search." Accessed in June 2018, https://www.sla.org.
36. Hearst, M. (2009). Search User Interfaces, CAMBRIDGE UNIVERSITY PRESS.
37. Broder, A. (2002). "A taxonomy of web search." SIGIR FORUM 36(2): 3-10.
38. Scherer, Mo. (2013). Information Retrieval for Multivariate Research Data Repositories. Deutschland, der Technischen Universität Darmstadt.

39. Blair, D. C. and M. E. Maron (1985). "An evaluation of retrieval effectiveness for a full-text document-retrieval system." Commun. ACM 28(3): 289-299.
40. Liu, T.-Y. (2011). Learning to Rank for Information Retrieval, Springer Heidelberg Dordrecht London New York.
41. Ranwez, S., et al. (2013). How Ontology Based Information Retrieval Systems May Benefit from Lexical Text Analysis. New Trends of Research in Ontologies and Lexical Resources: Ideas, Projects, Systems. A. Oltramari, P. Vossen, L. Qin and E. Hovy. Berlin, Heidelberg, Springer Berlin Heidelberg: 209-231.
42. Kirchhoff, L. (2010). Applying Social Network Analysis to Information Retrieval on the World Wide Web: A Case Study of Academic Publication Space. Graduate School of Business Administration, Economics, Law and Social Sciences. Germany, University of St. Gallen,.
43. Plachouras, V. (2006). Selective Web Information Retrieval. Department of Computer Science. United Kingdom, University of Glasgow.
44. Nettey, C. (2006). Link-Based Methods for Web Information Retrieval. INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION. Netherlands, Universiteit van Amsterdam.
45. Bush, V. (1945) As we May Think. The Atlantic. Accessed in June 2018, https://www.theatlantic.com.
46. Allen, R. (2017). "Search Engine Statistics 2017." Smart Insights (Marketing Intelligence) Ltd. Accessed in June, 2018, 2017, from web http://www.smartinsights.com/search-engine-marketing/search-engine-statistics/.
47. Arora, M., et al. (2010). Challenges in Web Information Retrieval. Innovations in Computing Sciences and Software Engineering, Springer, Dordrecht: 141-146.
48. Xu, H. and A. Li (2014). Two-Level Smart Search Engine Using Ontology-Based Semantic Reasoning. 26th International Conference on Software Engineering and Knowledge Engineering, Knowledge Systems Institute Graduate School: 648-652.

49. Li, Z. (2013). A Domain Specific Search Engine with Explicit Document Relations. Stockholm, Sweden, KTH, School of Information and Communication Technology.
50. Tjin-Kam-Jet, K. (2013). Distributed Deep Web Search. Netherlands, Centre for Telematics and Information Technology.
51. Baeza-Yates, R. and C. Castillo (2004). Crawling the Infinite Web: Five Levels Are Enough. Algorithms and Models for the Web-Graph: Third International Workshop, WAW 2004, Rome, Italy, October 16, 2004, Proceeedings. S. Leonardi. Berlin, Heidelberg, Springer Berlin Heidelberg: 156-167.
52. Santos, R. L. T. (2013). Explicit Web Search Result Diversification. School of Computing Science. United Kingdom, University of Glasgow.
53. Baeza-Yates, R. A. and B. Ribeiro-Neto (1999). Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc.
54. Elbedweihy, K. M. (2014). Effective, Usable and Learnable Semantic Search. Department of Computer Science. England, University of Sheffield.
55. Berners-Lee, T. (2004). Semantic Web, InterWord Communications for the Centre for Research in Web-based Applications, Rand Afrikaans University. 6.
56. Khan, H. U., et al. (2013). "Ontology Based Semantic Search in Holy Quran." International Journal of Future Computer and Communication, 2(6): 570-575.
57. Huang, L., et al. (2015). Semantic Search for Scientific Publications Based on Rhetorical Structure. IWOST-2.
58. S, S. R. B. and S. S (2014). "Ontology based Semantic Search Engine for Cancer." International Journal of Computer Applications 95(5): 39-43.
59. Vlachidis, A. (2012). Semantic Indexing via Knowledge Organization Systems: Applying the CIDOC-CRM to Archaeological Grey Literature. United Kingdom, University of Glamorgan.
60. Gruber, T. R. (1993). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." International Journal Human-Computer Studies 43: 907-928.

61. Boubekeur, F. and W. Azzoug (2013). "Concept-based Indexing in Text Information Retrieval." International Journal of Computer Science & Information Technology 5(1): 119-136.
62. Bast, H. and B. Buchhold (2016). "Semantic Search on Text and Knowledge Bases." Foundations and Trends in Information Retrieval 10(2-8): 119–271.

63. Lombardi, S. (2014). Context-awareness and context modeling. Ubiquitous Computing Seminar FS2014. The Distributed Systems Group at the ETH (Swiss Federal Institute of Technology) Zurich, Swiss.
64. Tian, J. (2010). Rich mobile context computing. The 2nd Workshop on Mobile Information Retrieval for Future (MIRF). Daejeon, Korea.
65. Bouidghaghen, O., et al. (2009). Dynamically Personalizing Search Results for Mobile Users. 8th International Conference, FQAS 2009., Roskilde, Denmark, Springer Berlin Heidelberg.
66. Gross, T. and R. Klemke (2002). Context Modelling for Information Retrieval - Requirements and Approaches. IADIS International Conference WWW/Internet ICWI 2002, Lisbon, Portugal.
67. Kehinde, A., et al. (2012). Context-Aware Stemming Algorithm for Semantically Related Root Words. African Journal of Computing & ICT: 33-42.
68. Jaimes, A. (2012). What Can Search Tell Us? A Human-Centered Perspective. International Workshop on Search Computing. Brussels.
69. Zhang, Y., et al. (2010). MQuery: Fast Graph Query via Semantic Indexing for Mobile Context. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Toronto, ON Canada, IEEE.
70. Nyiri, K. (2006). "The mobile telephone as a return to unalienated communication." Knowledge, Technology & Policy 19(1): 54-61.
71. Morgan, R. (2012). Relevance for the masses. The search solutions 2012 - Innovations in Web & Enterprise Search. London.
72. Kapor, M. (1993). Where is the digital highway really heading? The case for a Jeffersonian In-formation Policy. Wired Magazine. New York. 1: 1-13.

73. Alikilic, O. A. (2008). "When people are the message. Public participation in new media: User generated content." Journal of Yasar University 3(10): 1345-1365.
74. Banu, W. A., et al. (2011). "Mobile Information Retrieval: A Survey." European Journal of Scientific Research 55(3): 394–400.
75. Neisse, R., et al. (2008). Trustworthiness and Quality of Context Information. The 9th International Conference for Young Computer Scientists ICYCS 2008, Zhang Jia Jie, China.
76. Gicquel, P.-Y. (2010). Vers une modélisation des situations d'apprentissage ubiquitaire. Actes des troisièmes Rencontres Jeunes Chercheurs en EIAH, Lyon, France.
77. Saracevic, T. (1997). The stratified model of information retrieval interaction: Extension and applications. The American Society for Information Science Annual Meeting ASIS, Washington, DC.
78. Poveda-Villalon, M., et al. (2010). A context ontology for mobile environments. Workshop on Context, Information and Ontologies - CIAO 2010, Lisbon, Portugal.
79. Abowd, G. D., et al. (2001). Towards a Better Understanding of Context and Context-Awareness. Handheld and Ubiquitous Computing - First International Symposium, HUC'99 Karlsruhe, Germany, September 27–29, 1999 Proceedings, Springer Berlin Heidelberg. 1707: 304-307.
80. Dey, A. K. (2001). "Understanding and Using Context." Personal and Ubiquitous Computing 5(4).
81. Belkin, N. J., et al. (1999). Relevance Feedback versus Local Context Analysis as Term Suggestion Devices: Rutgers' TREC-8 Interactive Track Experience. TREC.
82. Bertrand, R., et al. (2012). "Filtrage Contextuel par Cache pour Application de Réalité Augmentée Mobile." Lavoisier - Document numérique 1: 57-77.
83. Kamvar, M. and S. Baluja (2006). A large scale study of wireless search behavior: Google mobile search. The SIGCHI Conference on Human Factors in Computing Systems, Montréal, Québec, Canada, ACM New York, NY, USA.

84. Kostadinov, D., et al. (2004). Système Adaptatif d'aide à la Génération de Requêtes de Médiation. 20èmes journées Bases de données avancées. France, Actes (Informal Proceedings: 351-355.
85. Ryan, N., et al. (1997). Enhanced Reality Fieldwork: the Context Aware Archaeological Assistant. Computer Applications & Quantitative Methods in Archaeology: 269-274.
86. Ryu, J., et al. (2010). Automatic Extraction of Human Activity Knowledge from Method-Describing Web Articles. 1st Workshop on Automated Knowledge Base Construction, Grenoble, France.
87. Bahsoun, L. T. a. W. (2006). Définition d'un profil multidimensionnel de l'utilisateur : Vers une technique basée sur l'interaction entre dimensions. Conférence en Recherche d'Infomations et Applications. France, Université de Lyon.
88. Kessler, C. (2007). Modeling and Using Context. CONTEXT: International and Interdisciplinary Conference on Modeling and Using Context. B. K. C. R. R. R.-B. Vieu. Denmark, Springer-Verlag Berlin Heidelberg 2007. 4635: 277–290.
89. Brown, P. J. and G. J. F. Jones (2001). "Context-aware Retrieval: Exploring a New Environment for Information Retrieval and Information Filtering." Personal and Ubiquitous Computing 5(4): 253-263.
90. Pasi, G. (2010). Issues in Personalizing Information Retrieval IEEE Intelligent Informatics Bulletin Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society. 11: 3-7.
91. Go, Y.-C. and J.-C. Sohn (2005). Context modeling for intelligent robot services using rule and ontology. The 7th International Conference on Advanced Communication Technology ICACT 2005., Phoenix Park, Dublin, Ireland, IEEE.
92. Lee, S. w., et al. (2010). Context Modeling Reflecting the Perspectives of Constituent Agents in Distributed Reasoning. IEEE/ACM Int'l Conference on Green Computing and Communications (GreenCom) & Int'l Conference on Cyber, Physical and Social Computing (CPSCom), Hangzhou, China, IEEE.

93. Mcheick, H. (2014). Modeling Context Aware Features for Pervasive Computing. The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014), Nova Scotia, Canada, Elsevier B.V.
94. Kalyan, A., et al. (2005). Hybrid context model based on multilevel situation theory and ontology for contact centers. Third IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom 2005), Kauai - Hawaii, IEEE.
95. Khattak, A., et al. (2014). "Context Representation and Fusion: Advancements and Opportunities." Sensors 14(6): 9628-9668.
96. Wu, Y.-L., et al. (2011). Using context models in defining intelligent environment information. 9th World Congress on Intelligent Control and Automation (WCICA 2011), Taipei, Taiwan, IEEE.
97. Wojciechowski, M. and M. Wiedeler (2012). Model-based Development of Context-Aware Applications Using the MILEO Context Server. IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2012), Lugano, Switzerland, IEEE.
98. Bettini, C., et al. (2010). "A survey of context modelling and reasoning techniques." Pervasive Mob. Comput. 6(2): 161-180.
99. Najar, S., et al. (2009). Semantic representation of context models: a framework for analyzing and understanding. Proceedings of the 1st Workshop on Context, Information and Ontologies (CIAO'09) Heraklion, Greece, ACM.
100. Bhargava, P., et al. (2012). An ontological context model for representing a situation and the design of an intelligent context-aware middleware. The 2012 ACM Conference on Ubiquitous Computing (UbiComp '12), Pittsburgh, PA, USA, ACM New York, NY, USA.
101. Taconet, C. and Z. I. Kazi-Aoul (2010). "Building context-awareness models for mobile applications." JDIM : Journal of digital information management 8(2): 78-87.
102. Bolchini, C., et al. (2007). "A data-oriented survey of context models." SIGMOD Rec. 36(4): 19-26.

103. Krummenacher, R., et al. (2005). Sharing Context Information in Semantic Spaces. On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, Springer Berlin Heidelberg. 3762: 229-233.
104. Hervas, R., et al. (2010). "A Context Model based on Ontological Languages: a Proposal for Information Visualization." Journal of Universal Computer Science 16(12): 1539-1555.
105. Buchholz, T. and M. Schiffers (2003). Quality of Context: What It Is And Why We Need It. The 10th Workshop of the OpenView University Association: OVUA'03, Geneva, Switzerland, ACM.
106. Preuveneers, D. and Y. Berbers (2007). Architectural backpropagation support for managing ambiguous context in smart environments. Universal Access in Human-Computer Interaction. Ambient Interaction, Springer Berlin Heidelberg. 4555: 178-187.
107. Khemissa, H. and M. Ahmed-Nacer (2012). "Adaptive Guidance based on Context Profile for Software Process Modeling." International Journal of Information Technology and Computer Science 2012(7): 50-60.
108. Akuma, S. (2014). "Investigating the Effect of Implicit Browsing Behaviour on Studens' Performance in a Task Specific Context." International Journal of Information Technology and Computer Science 2014(5): 11-17.
109. Corder, G. W. and D. I. Foreman (2014). Nonparametric Statistics: A Step-by-Step Approach (2nd Edition). Canada and New Jersey, John Wiley & Sons.
110. Legendre, P. (2005). "Species Associations: The Kendall Coefficient of Concordance Revisited." Journal of Agricultural, Biological, and Environmental Statistics 10(2): 226-245.
111. Bouhriz, N., et al. (2015). Text Concepts Extraction based on Arabic WordNet and Formal Concept Analysis. International Journal of Computer Applications: 30-34.
112. Jayanthi, R. and C. Jeevitha (2015). An Approach for Effective Text Pre-Processing Using Improved Porters Stemming Algorithm. International Journal of Innovative Science, Engineering & Technology: 797-807.

113. Atharva, J., et al. (2016). Modified Porter Stemming Algorithm. International Journal of Computer Science and Information Technologies: 266-269.
114. Ganesh Jivani, A. (2011). A Comparative Study of Stemming Algorithms. International Journal of Computer Technology and Applications: 1930-1938.
115. El-Defrawy, M., et al. (2015). CBAS: Context BASED ARABIC STEMMER. International Journal on Natural Language Computing: 1-12.
116. Singh Rajput, B. and N. Khare (2015). A survey of Stemming Algorithms for Information Retrieval. IOSR Journal of Computer Engineering: 76-80.
117. Abu-Salem, H. and M. Al-Omari (1998). Stemming Methodologies Over Individual Query Words for an Arabic Information Retrieval System. JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE: 524-529.
118. Hewlett Packard, E. (2015). "IDOL EXPERT." Vertica Advanced Analytics - myVertica.
119. Patel, C. and J. M. Patel (2015). A Review of Indian and Non-Indian Stemming: A focus on GujaratiStemming Algorithms. International Journal of Advanced Research: 1701-1706.
120. Widjaja, M. and S. Hansun (2015). Implementation of PORTER'S Modified Stemming Algorithm in an Indonisian Word Error Detection Plugin Application. International Journal of Technology: 139-150.
121. Vijayarani, S., et al. (2015). Preprocessing Techniques for Text Mining - An Overview. International Journal of Computer Science & Communication Networks: 7-16.
122. Ruba Rani, S. P., et al. (2015). Evaluation of Stemming Techniques for Text Classification. International Journal of Computer Science and Mobile Computing: 165-171.
123. Frakes, W. B. and R. Baeza-Yates (1992). Stemming Algorithms. Information Retrieval - Data Structures and Algorithms, Prentice Hall.
124. Xapian (2011). "Stemming Algorithms." The Xapian project.
125. Gormley, C. and Z. Tong (2014). Elastic search - The Definitive Guide. Accessed in June 2018, https://www.elastic.co.

126. Bimba, A., et al. (2015). Stemming Hausa text: using affix-stripping rules and reference look-up. Springer Science+Business Media Dordrecht 2015.
127. Ben Abdessalem Karaa, W. (2013). A New Stemmer to Improve Information Retrieval. International Journal of Network Security & Its Applications.
128. Subbu, K. and G. Vairaprakash (2014). Preprocessing Techniques for Text Mining.
129. Haven, o. (2016). "Text Tokenization and Processing in Text Indexes." Haven onDemand. Accessed June 2018, https://www.havenondemand.com.
130. Iadh, O., et al. (2006). Terrier: A High Performance and Scalable Information Retrieval Platform. SIGIR Open Source Workshop '06, ACM.
131. Fareh, M., et al. (2013). Merging ontology by semantic enrichment and combining similarity measures. International Journal of Metadata Semantics and Ontologies: 65-74.
132. Studer, R., et al. (1998). "Knowledge Engineering: Principles and Methods." Data & Knowledge engineering 25(1-2): 161-197.
133. A, M. and E. C (1996). The field-matching problem: algorithm and applications. 2ndInternational Conference on Knowledge Discovery and Data Mining.
134. Weller, K., et al. (2010). Social Interaction Technologies and Collaboration Software. Germany.
135. Lavoué, É. (2011). Social Tagging to Enhance Collaborative Learning. Advances in Web-Based Learning - ICWL 2011: 10th International Conference, Hong Kong, China, December 8-10, 2011. Proceedings. H. Leung, E. Popescu, Y. Cao, R. W. H. Lau and W. Nejdl. Berlin, Heidelberg, Springer Berlin Heidelberg: 92-101.
136. Chelmis, C. and V. K. Prasanna (2013). "Social Link Prediction in Online Social Tagging Systems." ACM Trans. Inf. Syst. 31(4): 1-27.
137. Vaidya, P. and N. S. Harinarayana (2016). "The role of social tags in web resource discovery: an evaluation of user-generated keywords." Annals of Library and Information Studies 63(4): 289-297.

138. Cantador, I., et al. (2011). "Categorising social tags to improve folksonomy-based recommendations." Web Semant. 9(1): 1-15.
139. Hayman, S. (2007). Folksonomies and Tagging: New developments in Social Bookmarking. Ark Group Conference: Developing and Improving Classification Schemes. education.au. Rydges World Square, Sydney.
140. Antipolis, U. N. S. (2011). Social Bookmarking : Gérer ses signets. France, Service commun de la documentation.
141. Al-Rasheed, A. and J. Berri (2014). "Social Bookmarking as a Knowledge Sharing Tool." International journal on information.
142. Jabeur, L. B. (2013). Leveraging social relevance: Using social networks to enhance literature access and microblog search. Institut de Recherche en Informatique. France, Université Toulouse 3 Paul Sabatier.
143. Peters, I. and W. G. Stock (2008). Folksonomy and Information Retrieval. The American Society for Information Science and Technology. 44.
144. Nam, H. and P. K. Kannan (2014). "The Informational Value of Social Tagging Networks." Journal of Marketing 78: 21-40.
145. Abel, F. (2011). Contextualization, User Modeling and Personalization in the Social Web: From Social Tagging via Context to Cross System User Modeling and Personalization. Von der Fakultat fur Elektrotechnik und Informatik. Deutschland, Gottfried Wilhelm Leibniz Universitat Hannover.
146. Socialadr (2010). Social Bookmarking for Dummies. Accessed in June 2018, http://socialadr.com.
147. Pei, J. (2010). Information Retrieval and Web Search. Social Search. Canada, Simon Fraser University.
148. Trant, J. (2009). "Studying Social Tagging and Folksonomy: A Review and Framework." Journal of digital Information 10(1).
149. Peters, I., et al. (2011). Social tagging & folksonomies: Indexing, retrieving… and beyond? Association for Information Science and Technology.
150. Barnes, L. L. (2011). "Social Bookmarking Sites: A Review." Collaborative Librarianship 3(3): 180-182.

151. Portmann, E. (2013). The Social Semantic Web. The FORA Framework: A Fuzzy Grassroots Ontology for Online Reputation Management, Springer-Verlag Berlin Heidelberg: 13-36.
152. Font, F., et al. (2012). Folksonomy-based Tag Recommendation for Online Audio Clip Sharing. International Society for Music Information Retrieval. Portugal: 73-78.
153. Mousselly-Sergieh, H., et al. (2013). Tag Similarity in Folksonomies. INFORSID 2013, Paris, France.