FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Implementation of an Information Retrieval System within a Central Knowledge Management System

Daniel Fidalgo Rodrigues

Report of Dissertation
Master in Informatics and Computing Engineering

Supervisor: Ana Paula Rocha, PhD.

26th July, 2010

Implementation of an Information Retrieval System within a Central Knowledge Management System

Daniel Fidalgo Rodrigues

Report of Dissertation
Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Jaime dos Santos Cardoso (Assistant Professor)

External Examiner: José Luis Guimarães Oliveira (Associated Professor)

Internal Examiner: Ana Paula Cunha da Rocha (Assistant Professor)

_____

26th July, 2010

Abstract

With the advent of internet, corporate intranets and mass storage supports dissemination, there was a colossal increase of the amount of available data worldwide. Traditional Information Retrieval (IR) techniques became inadequate for the increasingly vast amounts of information, as only a small fraction of it would be relevant to a given individual. Individuals require modern IR frameworks to evaluate and rank the significance of the retrieved information, while uncovering better models for matching the user's information needs with information that will satisfy their requirements.

In order to pursue current Knowledge Management demands from actual corporations, an Information Retrieval system was developed in order to provide efficient indexing and full text searching features over a large collection of documents.

This system was later improved with a Passage Retrieval mechanism, in order to retrieve portions of text instead of full documents. This is motivated by the fact that, as passages are ranked, users can be rapidly driven to the relevant information within the document.

Finally, an evolutionary approach using Genetic Algorithms was integrated within Information Retrieval context in order to take advantage of query expansion. The algorithm exploits not only the syntax from user's questions, but also the semantics beneath them via a linguistic question analyzer module and a semantic network. Then, the GA combines several keywords along with several operators in order to optimize the selection of the most relevant terms and weights, progressively refining the queries.

Some experiments were conducted in order to measure the efficiency from the Genetic Algorithm applied to the Passage Retrieval mechanism, and to understand how passages should be defined and how they should be used in retrieval. The results were promising as all the passage retrieval approaches using Genetic Algorithms yield performance improvements, when compared with the standard Information Retrieval systems.

.

Resumo

Com a proliferação da Internet, intranets corporativas e difusão dos suportes de armazenamento, assistiu-se a um aumento colossal da quantidade de dados disponível. Técnicas tradicionais de Recuperação de Informação (*Information Retrieval*) tornaram-se inadequadas tendo em conta a actual procura de informação, uma vez que apenas uma pequena parcela da informação é relevante para o utilizador. Existe uma necessidade de ferramentas modernas de Recuperação de Informação que permitam avaliar e classificar a importância da informação obtida, em conjunto com modelos melhores de cruzamento das necessidades de informação dos utilizadores com informação que satisfaça os seus requisitos.

De modo a completar os requisitos de Gestão do Conhecimento das empresas contemporâneas, foi implementado neste projecto um sistemas de Recuperação de Informação de forma a fornecer uma indexação eficiente e pesquisa em texto apartir de grandes colecções de documentos.

Posteriormente, esse sistema foi aperfeiçoado com um mecanismo de Recuperação de Passagem (*Passage Retrieval*), a fim de retornar partes de texto ao invés de documentos completos. Esta abordagem é motivada pelo facto de que, como as passagens são ordenadas segundo uma classificação, os utilizadores podem ser rapidamente direccionados para a informação pertinente do documento.

Finalmente, uma abordagem evolutiva utilizando Algoritmos Genéticos foi também integrada no contexto de Recuperação de Informação, de modo a tirar partido da expansão das interrogações (*Query Expansion*). O algoritmo explora não só a sintaxe das questões dos utilizadores, como também a semântica inerente a elas através de um módulo que se encarrega do processamento linguístico das questões e de uma rede semântica. Seguidamente, o Algoritmo Genético combina várias palavras-chave e experimenta diferentes operadores com o intuito de optimizar a selecção dos termos mais relevantes e os seus pesos, refinando progressivamente as pesquisas.

Algumas experiências foram realizadas para medir a eficiência do Algoritmo Genético aplicado ao mecanismo de Recuperação de Passagens e compreender igualmente como as passagens devem ser definidas e usadas num contexto de Recuperação de Informação. Os resultados foram promissores já que todas as abordagens usando Algoritmos Genéticos mostraram melhorias de desempenho, quando comparadas com sistemas padrão de Recuperação de Informação.

Acknowledgements

I would like to thank:

Prof. Ana Paula Rocha for all the help and relevant recommendations for my work;

Wipro Portugal SA, particularly Eng. Hugo Neto for the vigorous support and the continuous encouragement;

Wipro's trainees, for the wonderful environment. Thanks for making each day of work always an enjoyment!

Rui, Martha and Ivo for the support and the critical review over my work;

Carolina for all the encouragement;

Prof. Cristina Ribeiro for the relevant feedback and the useful hints;

Prof. Rui Camacho for the suggestions;

And most of all, Domingos, Fernanda, Joana and all my family for their never-ending support.

Daniel Fidalgo Rodrigues

Contents

List of Figures

List of Tables

Abbreviations

CLEF	Cross Language Evaluation Forum
FAQ	Frequently Asked Questions
GA	Genetic Algorithm
IR	Information Retrieval
IT	Information Technology
KM	Knowledge Management
MAP	Mean Average Precision
MG4J	Managing Gigabytes for Java
NL	Natural Language
NLP	Natural Language Processing
PR	Passage Retrieval
QA	Question and Answering
SCM	Supply Chain Management
TREC	Text REtrieval Conference

Glossary

Bing:	Microsoft’s search engine.
Bookmarking:	Organize bookmarks with informal tags.
Data Mining:	Process of extracting patterns from data.
Department Store:	Retail establishment which specializes in satisfying a wide range of the consumer's product needs.
Entity Recognition:	Find and classify a word in a category.
Expert System:	System that attempts to provide an answer to a problem, where normally one or more human experts would need to be consulted.
Information Retrieval:	Science of searching documents for relevant information.
Lucene:	Information Retrieval framework.
Natural Language:	Natural language spoken by the humans.
Natural Language Processing:	Methodologies regarding interactions between computers and human natural languages.
Ontology:	Representation of knowledge by a set of concepts within a domain and their relationships with other concepts.
Open-Source:	End-product, source-material, and documentation available at no cost to the public.
Podcast:	Series of audio or video files that are released episodically.

Retailing:	Sale of goods or merchandise from a fixed location.
RSS feeds:	Formats used to publish frequently updated works.
Search Engine:	Tool designed to search for information.
Semantic Network:	Represents semantic relations among concepts.
Stemming:	Reduce words to their morphological root.
Stop-Words:	Words from the linguistic point of view don't provide information.
Supply Chain Management:	Management of the network of interconnected businesses involving the provision of products to customers.
Text Mining:	Process of deriving high-quality information from text.
TREC:	Challenges focusing on a variety of different information retrieval research areas.
Wal-Mart:	American corporation that runs a chain of large department stores.
Wikipedia:	Free web encyclopaedia.
WordNet:	Lexical database for the English language.
Yahoo:	Web search engine.

Chapter 1

Introduction

This manuscript embodies the documentation of the final Dissertation Project, from the *Master in Informatics and Computing Engineering*, from *Faculty of Engineering of the University of Porto*. Furthermore, it should be noted that this project is carried out within business environment, more precisely within Wipro Portugal SA - *Wipro Retail*.

This first chapter situates the reader with the context of the problem and provides the motivation behind this project. The following sub-section 1.1 introduces the background and scope of the project. Afterwards, the sub- section 1.2 describes the incentive behind the project, and later, the sub- section 1.3 presents the main problem to solve plus the main goals to accomplish. Finally, sub- section 1.4 gives an overview over the organization of the document.

1.1. Context

Wipro Technologies is a global IT services business division from Wipro Limited, dedicated to provide integrated technological solutions on a global delivery platform. *Wipro Retail*[1] is a division of Wipro Technologies, only dedicated to provide business solutions to retailers from around the world, such as food or fashion retail chains. "*Retailing is the sale of goods and services to the ultimate consumer*" [CB04].

With the current advent of modern enterprise collaboration tools, the Knowledge Management topic regained new relevance once more [Fra10]. Information preservation is essential for the company's strategy. Knowledge is one of the biggest differentiator features

[1] ***Wipro Retail**: http://www.wipro.com/industries/retail/retail/index.htm*

between companies, allowing organizations to generate competitive advantages over competitors [Irw79].

However, the effort required to achieve a superior knowledge sharing level over teams who typically work in a distributed way as *Wipro Retail* does, tends not to be so effectively or timely.

Following that *Wipro Retail* is a global scale corporation, the amount of information exchanged is very significant. On the one hand, *Wipro Retail* is organized by area experts, such as price management, warehouse management, stores operation management and so on. Therefore, business knowledge is often spread along many entities. On the other hand, different collaborators could be working in the same project, but at different time zones. From this point of view, it is highly inflexible and inconvenient for one employee that is currently working from the London office (UTC 0) to call at 18:00 pm to other specialist teammate that is working at Bangalore (India, UTC +5:30), just to attempt to find proper details regarding some module which is not from his expertise area. Moreover, it's highly costly and time consuming to browse vast collections of documents in order to uncover solutions to problems.

Additionally, as *Wipro Retail* repeatedly provides project consultancy to customers, several new collaborators are introduced in the company. Many times, new collaborators are not aware of the business technology terminology used. Hence, the effort required to train new collaborators could be reduced, as there is no need of overburden employees explaining trivial concepts to the trainees.

Considering that the amount of information produced by large companies is very considerable, more than ever is mandatory to support proper management for digital assets, especially between team members who have to work at different physical locations. Indeed, Knowledge management tools became a necessity as large teams which are geographically dispersed, need to consult relevant documentation on a real-time basis, in order to coordinate their activities. In short, it is essential to set all conditions, so that employees can engage with information seeking process, helping them to achieve their goals.

Enterprises that focus on provide services to customers, have to be in the vanguard of thinking better ways to manage knowledge as their own success depends heavily on building, applying and selling ideas. Following that view, Wipro's Innovation Department proposed the conceptualization of a model that could actively support Knowledge Acquisition, Knowledge Discovery and Knowledge Recovery.

Therefore, as shown on Figure 1, *Prymas* is a complete Knowledge Management architecture, supporting knowledge development cycle. *Prymas* is a research and exploration project developed by a team of three members, over the course of sixteen weeks.

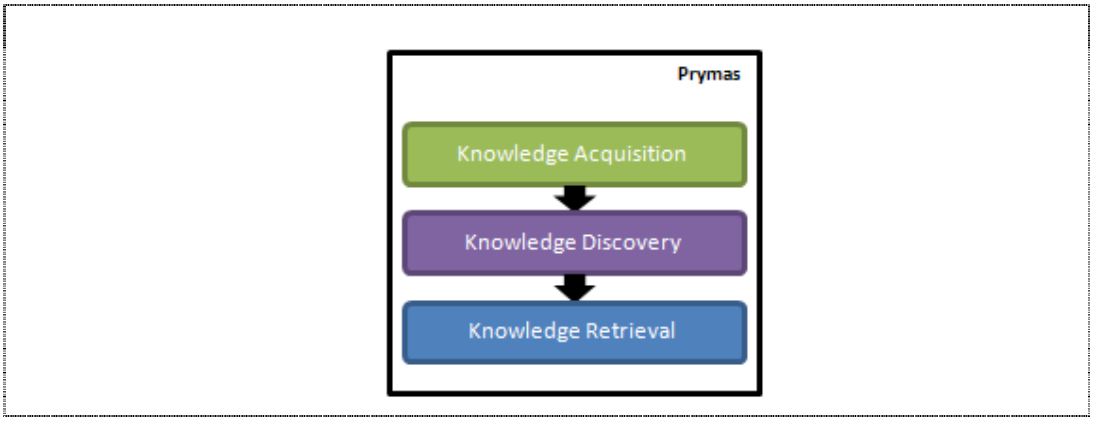

Figure 1 - Prymas Architecture

Nonetheless, all the contributions and solutions that this manuscript proposes, focuses solely on Knowledge Retrieval scheme and on the pursuit of better mechanisms to expose more relevant information to users.

1.2. Motivation

"An investment in knowledge always pays the best interest."

Benjamin Franklin[2]

As Andriessen attempted to explain, information is data within a context which gives data a meaning [And04]. Still, knowledge is information experienced in a particular situation, and therefore connected to feelings and understanding. Then again, Novak made further remarks, stating that knowledge is more than just information; it's about beliefs and commitments [Nov98].

With the advent of internet, corporate intranets and mass storage supports dissemination, there was a colossal increase of the amount of information available worldwide [GHM05]. On a daily basis, companies have to strive with huge amounts of data in order to scrutinize knowledge that could be applied into their business activities. Even so, information can't be knowledge unless suitable mechanisms can extract value out of it [Liu10], plus better paradigms to interact with information were employed in order to better fulfil the present business demands.

Still, large corporations are the ones more concerned with producing, gathering and storage of information [Gar09], making it available to the world in such overwhelming quantities that far surpass the human processing capability. Nowadays, it is unlikely that we can't come across a subject not already seen by someone [Jaw10], even already documented and available to the

[2] ***Benjamin Franklin**: American politician (1706-1790)*

public. However, we are still in the moment of objective exchange of data and information without generating any social value, that is, knowledge. As Thomas S. Eliot[3] once said, "*where is the knowledge we have lost in information?*"

At present, Knowledge Management scheme is a critical harassment for all organizations. One of the main obstacles towards an effective information retrieval is due to the fact that almost 80% of documentation produced within corporations is weekly structured and lacks from a formal representation [Mur03]. Considering all the information hosted in enterprises, that data is not actually stored on rows and columns, but across different media including technical reports, voice messages, digital presentations and videos. The high-value material underlying these vast collections of unstructured information is, unfortunately, buried in lots of superfluous data. This crushing amount of records makes processing unstructured data for storage and retrieval, a major challenge for most organizations, as information overload places an obstacle to knowledge sharing, and leads to work duplication and reinventing-the-wheel.

Indeed, information overload is one pertinent subject matter that Knowledge Management endeavours to resolve. Many companies have realized that and put a lot of effort in order to overcome this issue [Wra10]. Due to the continuous and exponential growth of the amount of data worldwide [BR99], automated extraction methodologies from previously unknown and potentially useful information are essential. In addition, hasty recovery mechanisms that could extort relevant information accordingly to user's requests, turns into more compulsory demands particularly under business environment. Judging all the corporate documentation ranging from best-practices, technical reports, issues reports, customer communications and contracts, to emails and voice messages, all those assets require proper management and proficient search infrastructures in order to engage individuals with their tasks. Within these piles of natural language artefacts, over and over again lays the pieces of knowledge decisive for understanding important market trends, patterns to solve recurrent problems and ideas to create new product opportunities.

In short, research on Information Retrieval subject becomes more and more imperative as it tackles the Information Overload issue. Individuals have difficulty understanding problems and making decisions by the presence of too much information. Following current business demands, exploration on how to organize, represent and interact with information assets, along with uncovering better models to match user needs to relevant information, present themselves as pertinent motivations on today's world.

1.3. Problem Description and Main Goals

Prymas Knowledge Management Scope

[3] ***Thomas S. Eliot****: American poet (1888- 1965)*

In order to face actual business requirements and keep abreast current market trends, Wipro's innovation department came up with the challenge to conceive an architecture that could successfully convert unstructured information into knowledge, and could afford proper knowledge management mechanisms. Moreover, as users prefer to express their information needs in their natural language, the solution has to sustain a ready-and-easy interaction to organization's documented base of facts. This project was called *Prymas*.

Nevertheless, the critical point in order to develop a complete solution that takes from unstructured information to usable knowledge, a variety of linguistic processing models and information management frameworks have to be integrated. These components must perform a comprehensive analysis task over the information, so that the results can be funnelled into systems that could allow users to rapidly find and exploit the discovered knowledge.

Developing such knowledge system as *Prymas,* undertakes the following targets:

- Actively manage intellectual capital from the organization. Central Knowledge Management systems offer a unified view over knowledge, supporting knowledge gathering, discovery, integration, and sharing.
- Leverage the expertise and make the knowledge available all along the organization.
- Allow employees to obtain significant insights fitting their requirements, supporting the development and maintenance of products and service providing assignments.
- Reorganize operations and decrease costs as redundant or unnecessary processes are skipped.
- Efficiently address information overload problem, through proficient methodologies and tools towards proper management of large collections of documents, while allowing users to perform meaningful searches to them.

It should be noted that this project will be instantiated to the retailing domain, as the intended purpose of *Prymas* is to provide consistent support to retail consulting projects, while lending a hand to training tasks as well.

Knowledge Retrieval Scope

The explosion of digital repositories led to large amounts of information within our reach. Gradually, the amount of information available has become so vast that more alternative and dynamic mechanisms are needed in order to find pertinent information [GHM05].

As Baeza-Yates and Ribeiro-Neto tried to describe, the key goal of an Information Retrieval (IR) system is to provide easy access to the information in which users are interested [BR99]. Traditional techniques become inadequate for the increasingly vast amounts of text data, as typically, only a small fraction of the many available documents will be relevant to a given individual. In truth, without previous knowledge what could be in the documents, it is complex to formulate effectual queries for retrieving useful information, while avoiding missing

relevant one. Individuals require modern information retrieval frameworks to evaluate different documents, grade the meaning and relevance of the fetched information accordingly to their queries, even find patterns and tendencies all across diverse artefacts.

Question and Answering (QA) systems provide easy and intuitive interface to knowledge, as users place questions and obtain answers in their natural language. Recently, there was an increase on the interest of QA systems that do not rely on a knowledge database, but could extract answers from huge unstructured texts [HAL02]. For this reason, IR modules persistently support the Knowledge Retrieval task, as they try to understand and translate the user's information needs into queries, in addition to provide suitable technology to recover information that could fulfil user's requirements.

As the participation of this dissertation focus only on the implementation of an Information Retrieval module within the *Prymas* Knowledge Retrieval layer, the main goals to accomplish are:

- Improve access to the information assets. Systematize all corporations' documentation and make it accessible. Seeking and finding information on mounds of documentation is of paramount importance.
- Address information overload problem providing better methods to fetch pertinent information from large information collections, plus delivering accurate mechanisms for matching user's information needs.
- Build a support infrastructure to a QA system. IR systems decode questions into queries, attempting to identify all relevant and purposeful information that could better support answers to users.
- Polish relevant retrieval matching, enhancing queries to the IR system with suitable information from the discovered knowledge, in an attempt to retrieve as little non-relevant information as possible.

1.4. Structure of the Document

This document is organized as follows:

Counting the Introduction, this dissertation manuscript contains over 5 chapters. First, on Chapter 2 describes the state-of-art and related work to the subject. Chapter 3 explains the implementation methodology from one Information Retrieval System, within a Knowledge Management System. Later, on Chapter 4 are illustrated the experiments and the comparisons done with the system developed. Chapter 5 makes a summary over the development scheme, and draws some conclusions over the results obtained and the goals accomplished. Also, future expansions for this project are presented in this last chapter.

Finally, in the end of the document figures the bibliography used and the appendices useful to further comprehension of the work done.

Chapter 2

Literature Revision

The four following sub-sections disclose the research already published and challenges of handling large amounts of unstructured information plus Knowledge Management topic, Question and Answering systems, Information Retrieval along with Passage Retrieval, and at last, the appliance of Genetic Algorithms headed for query optimization within Question and Answering context, while introducing the current existing scientific and technologic lacunas.

2.1. Knowledge Management Systems

"The only irreplaceable capital an organization possesses is the knowledge and ability of its people. The productivity of that capital depends on how effectively people share their competence with those who can use it."

Andrew Carnegie[4]

2.1.1. Information Overload

Current Trends

IT companies have to do a huge effort to keep up with the current service demands. Actually, expanding the digital universe extremely surpass the capacity of storage from

[4] ***Andrew Carnegie****: Scottish business magnate (1835-1919)*

enterprises. In fact, nearly 70% of the digital content in the world is generated by individuals, but its storage is mainly a corporations' concern [Wra10].

Online information was estimated to exceed all human documentation generated in the first 40,000 years of human history, and is vastly more than all the information on Earth that all humans can learn [Her09]. *Google*, one of the most current popular search engines is reported to processes around 1 *petabyte*[5] every hour [Sef10]. Expanding the digital world size in 2007, it was predicted to be equivalent to 161,000 *petabytes* [Wra07]. A research conducted in 2010 [Wra10], shown that the estimated size of digital universe was 800,000 *petabytes*, increasing almost 5 times more, in just in 3 years. IDC[6] predicts that the estimated size of digital universe in 2020 will be 35,000,000 *petabytes*, increasing about 44 times in 10 years [IDC10].

For instance, UK internet users now spend 64% more time using search engines (31 million hours per month in April 2010) than they did 3 years ago [UKO10]. Globally are done more than 130 billion searches per month [Com10], which shows the pertinent role of Information Retrieval.

A study done by Accenture says that middle managers spend up to 2 hours a day searching for information to do their jobs, and almost 60% of the information they obtain has no value to them, as a consequence of poor information distribution. They miss information that might be valuable to their jobs almost every day because it exists somewhere else in the company and they just can't find it. Moreover, 36% said there is so much information available that it takes a long time to actually find the right piece of data [Acc07].

IDC research found that a company that employs 1,000 IT workers can expect more than $5 million in annual salary costs because of the time wasted looking for information and not finding it [IDC07].

Unstructured Information

One of the main blockages towards effective information management arises from the fact that roughly 80% of the organizational documentation is within an unstructured format [Mur03], that is, doesn't follow a strict template nor a specific semantic. That takes account of all data formats within the enterprise, including the continuing flow of emails and respectively attachments exchanged between employees, videos and presentations used in the training sessions, as well as the spreadsheets and technical reports produced to the customers. Since such a big share of corporation's knowledge sources is randomly or weakly structured, traditional databases techniques cannot be applied to manipulate and exploit the value from the textual contents. In short, that means that about 80% of business knowledge produced from the organization can't be resourcefully employed in the business processes straight away.

Therefore, processing unstructured information for further knowledge retrieval makes a pertinent issue for most organizations, as this information overload leads to inefficient knowledge sharing and work duplication. As knowledge lies on "raw text" without any mark-ups, analysis and retrieval technology can be valuable methods.

[5] ***Petabyte***: *equals to 1000 terabytes.*

[6] ***IDC***: *http://www.idc.com/gms/index.jsp*

2.1.2. Knowledge Management Overview

"*Knowledge management (KM) is the fast-track route to leveraging the intellectual capital in the organisations*" [Fra06].

In fact, KM is defined as the set of practices regarding the management of the organization's knowledge, all the way through a systematically procedure for acquiring, organizing, sustaining, applying and sharing knowledge to enhance organizational performance and create value [All97]. By that, KM establishes a set of processes for capturing and re-using organisational assets, enabling different elements from the company to achieve more successfully and timely their goals.

Typical corporate KM systems support gathering, discovery, classification and storage of information, enabling employees to have ready access to organization's documented knowledge. Sharing solutions and best practices wide along the organization can lead to more effective processes design and managing, while lending a hand to novel or improved concepts.

Nevertheless, the process of knowledge acquisition by employees in reality deviates from the actual KM mechanism. Despite of significant knowledge being stored within KM media, yet mostly employees ask for information throughout interpersonal relations with their co-workers [Bol07].

It was expected that KM tools help to save $31 billion in annual re-invention costs at *Fortune 500 companies*[7] [Mal05].

2.1.3. Technologies Outline

Knowledge Management (KM) requires proper technologies to support the current demanding strategies from enterprises, while providing efficient routines to better capture, retain, and share knowledge. The IT infrastructure should afford the proper flow of all stages of the knowledge cycle, enabling knowledge acquisition and discovery, knowledge storing and categorization, and knowledge retrieval [Zac99].

One of the challenges of KM systems is the effectual ability of searching relevant information on behalf of decision support tasks [Pul08]. Expert systems, Information Retrieval Systems and Search Engines play a key role on knowledge retrieval. Many organizations already included in their IT structure search engines to automatically index their knowledge assets, quickly allowing them to find what they want. There are several search engines available (see section 2.3.4. Technologies Outline, for further details).

Secondly, there are several professional tools available concerning knowledge exploration, most of them based on *Data Mining* and *Text Mining* techniques, like *BLIASoft Knowledge*

[7] ***Fortune 500 companies***: *annual list compiled and published by Fortune magazine that ranks the top 500 U.S. closely held and public corporations by their gross revenue.*

Discovery [BLI10] , *EWA Systems Enterprise Analytics* [EWA10] , and *Megaputer* [Meg10]. There's also some Open-Source software available with similar capabilities like *Weka* [Wek10], and *Rapid-I* [Rap10].

Furthermore, in order to support suitable KM, users have to take the possibility to communicate and collaborate efficiently with their co-workers. Regarding this point, Web 2.0 assumes itself as a pertinent matter, as technology that facilitates content publishing and managing through network applications. Technologies such as weblogs, wikis, social bookmarking, podcasts, and RSS feeds [Pul08] , online communities and e-learning platforms are also relevant services in this matter.

Afterward, there is also a wide range of complete professional software committed to information organization and knowledge sharing such as *IBM Lotus Notes* [IBM10], *Knorg* [Kno10], *Autonomy Agentware i3* [Aut07], *Paradigm OpsLink* [Par10].

Later, many organizations realized the importance Enterprise Content Management (ECM), and started to include that on the companies IT infrastructure. There are several ECM tools available akin to *Sales Force Content Library* [Sal10] or *Box* [Box10]. Besides, there are some Open-Source solutions in the market like *Nuxeo* [Nux10] and *Alfresco* [Alf10] as well.

Finally, corporation's knowledge has to be available and shared anytime and anywhere, all across every collaborator. Indeed, this feature is deeply connected to ubiquitous technologies, as information has to be comprehensively integrated into everyday objects and activities. Ubiquitous computing comprises an extensive assortment of technologies, counting distributed computing, mobile computing, human-computer interaction, and artificial intelligence [Pul08].

2.2. Question and Answering Systems

"For your information, I would like to ask a question."
Samuel Goldwyn[8]

2.2.1. Question and Answering Overview

In recent years, outbreak of digital content along with web search services proliferation has created a demand for instruments committed to rapidly assist users passing over irrelevant information. One interesting approach for this concern is Question Answering (QA) systems. "*The goal of Question Answering is to allow users to ask questions in natural language, using their own terminology, and receive a concise answer, possible with enough validating context*" [HG01].

Given a question, such as "*What is the name of the current president of U.S.A*?" common keyword-based search engine such as *Google* or Yahoo might present the user a bunch of links for relevant documents. Thus, current search engines are able to efficiently retrieve a set of

[8] ***Samuel Goldwyn****: American film producer (1879-1974)*

documents sorted by relevance, even though they do not retrieve concrete answers to user's information needs. In contrast, QA systems would attempt to answer directly to users with the name of the current president of USA.

Users interact directly with QA systems throughout Natural Language, which put across more convenience and understanding to users submit their information needs. Additionally, headed for uncovering the answer, QA systems typically consult a structured database or a collection of unstructured text documents, such as text reports or Wikipedia pages.

QA systems can be classified as closed-domain as they deal with questions under certain and specific domain, for instance, healthcare or motor manufacturing. These systems can be built in order to explore particular domain from ontologies and semantic relationships. On the other hand, QA systems can be classified as open-domain. Open-domain QA systems are more generic, as they deal with general questions about every topic. Thus, they handle with much more information and can only rely on nonspecific and general ontologies.

Development and research in QA has been incited by the Text Retrieval Conference (TREC)[9] series since 1999 [Cui05]. QA track has been running until the present, where the systems partaking in this contest are expected to response questions on any theme by searching a corpus of documents that diverse every year [TRE10].

Typical goals for QA systems might range from on-line help systems that provide technical support, to very sophisticated systems that support complex business processes and comprehensive analysis tasks [May02]. The most famous QA systems developed in the past were closed domain, such as LUNAR[10] and BASEBALL[11]. More to the point, there are several examples of web QA systems available online, such as *AskJeeves* [Ask10], *Wolframalpha* [Wol10] and *Trueknowledge* [Tru10]. *Google* also started to include QA features to their search engines [Goo10].

At last, OpenEphyra is a start-of-the-art Java open framework for QA. It returns answers to Natural Language questions from the Web and other sources [Ope08].

2.2.2. Architectural Challenges

Classic Question and Answering (QA) systems include a question analyser module that classifies the Natural Language question and extracts relevant query parameters. Afterwards, a group of candidate documents is retrieved from the whole collection of documents, comprising all the features that match with the initial query requisites. Following typical TREC systems, an Information Retrieval (IR) system (see section 2.3. Information Retrieval Systems) usually supports this task. After that, parts of documents are spotted and ranked, regarding where a possible answer to the initial information need is likely to be. Finally, the answer extractor module is responsible for processing the small fragments of text, building and retrieving the

[9] ***TREC***: http://trec.nist.gov

[10] ***LUNAR***: *QA system about geological analysis of rocks returned by the Apollo moon missions, 1960*

[11] ***BASEBALL***: *QA system about the US baseball league, 1960*

candidate answer to the user.

The following Figure 2 presents a generic and simplified architecture for a QA system, based on the work done by [HG01].

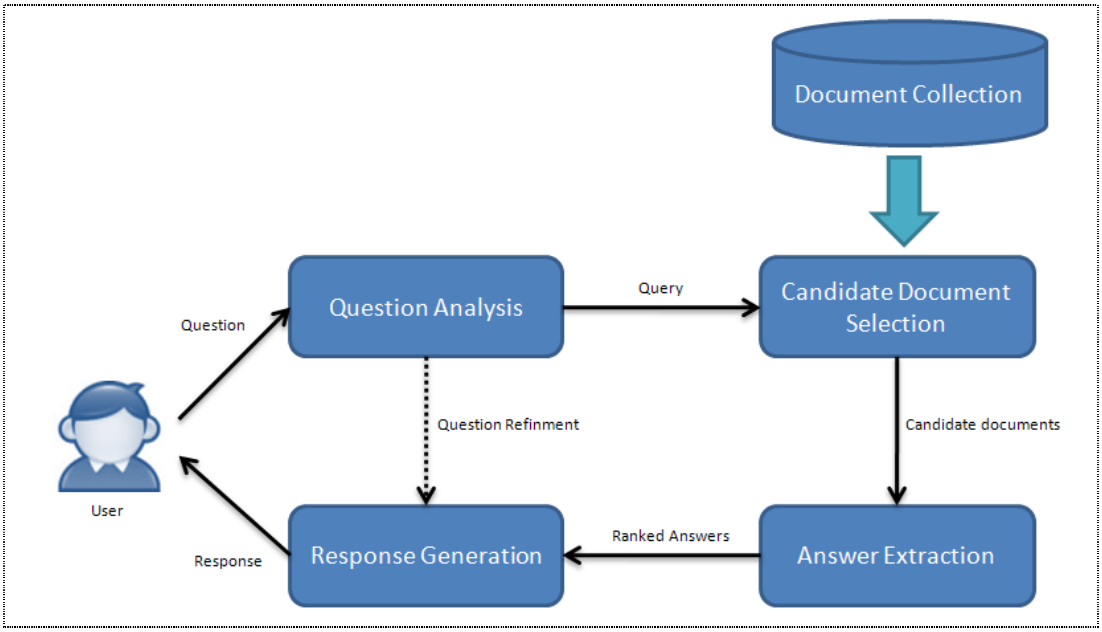

Figure 2 - Simplified Architecture for a Question Answering System

Nonetheless, there are some architectural issues regarding QA systems, many of them researched in detail [BC02]:

Question Analysis

From a Natural Language question, the question analyzer module is responsible for qualifying the question and digging linguistic features. Different types of questions require the use of different strategies to find the answer [May02]. A semantic model of question understanding and processing is essential, that recognizes equivalent questions, apart from the speech act or of the words, syntactic inter-relations or idiomatic forms [BC02].

Questions are usually asked within a context. Moreover, the answer provided is also constrained to some specific context. Thus, context is important as it can clarify the question, resolve ambiguities and keep track of an investigation performed through a series of questions [May02].

Many times, the user may be required to use a controlled language, restringing the vocabulary and syntax. Moreover, it may even be that the user has to use some form-filling interface, restraining the expressivity as well [HG01].

The final result from this process is a representation of the user information. Such

representation will suit the purpose of the following stages. In fact, this representation aims to be an efficient model that could assist the QA system to unravel the answer. This representation model could be for instance, a stemmed[12] and weighted array of input terms, expressing the user's query.

It is often the case that the information need is not well captured by a QA system, as the question processing part may fail to classify correctly the question. More to the point, sometimes the information need is not completely fulfilled as it was not accurately retrieved. In such cases, the user might want not only to reformulate the question, also have a dialogue with the system.

Document Collection Pre-processing

If answers are to be extracted from terabytes of data in quite a few seconds, then pre-processing text sources is compulsory. Thus far most TREC QA systems appear to rely on traditional document indexing engines to carry out this assignment [HG01]. In actual fact, Information Retrieval frameworks provide capable services for indexing large amounts of data, while allow fast retrieval mechanisms (see section 2.3. Information Retrieval Systems for further details).

Before a question can be answered, it must be known what the knowledge is available. If the answer to some question is not present in data sources, regardless of how well the question is analysed, the documents are searched and the extraction is performed, the correct result is never retrieved.

Candidate Answer Document Selection

As referenced before, most existing TREC QA systems use some conventional Information Retrieval search engine to pick a set of candidates from the whole documents corpus. First, if a Boolean engine is used, restricting the number of returned documents that need to be examined still needs to be addressed; however, if a ranked answer engine is used instead, a decision must be made as to how many documents retrieved will be used; Secondly, the search engine may return links to the documents, or else relevant snippets or passages from the documents.

Once a bunch of documents is selected, that is considered a good candidate for fulfilling the answer requisites, text segments from them are further analysed. Typically tasks from this stage regard detection of compound-words[13], and name-entity recognition[14].

Answer extraction

[12] ***Stemming:*** reduction the words to his root, i.e. removing all affixes of the word.

[13] ***Compound-words:*** two words are joined to form a new word.

[14] ***Name-entity recognition:*** identifying persons and organizations.

At this phase the question representation model is compared against the representation of the candidate answers. Then, a set of candidate answers is produced, accordingly ranked to the likelihood of correctness. The matching process may require that a text passage from a candidate answer text (e.g. a sentence) contain a string whose semantic type matches that of the expected answer. Then, once a text unit containing an expected answer type has been found, other constraints may be applied to the text snippet. These constraints may be viewed as mandatory, so failure to satisfy this rules set the candidate as non relevant; or they may be viewed as preference features, which can be used to give a score to the candidate to the rank the answer. Considerable deviation exists in terms of the types of restrictions used at this stage, how constraint satisfaction is carried out, and how constraints are weighted [HG01].

Answer extraction depends on the complexity of the question, on the answer type provided by question processing, on the actual data where the answer is searched, on the search method and on the question focus and must be relevant within a specific context.

Response generation

The response of a QA system should be presented in a way as natural as possible. In some cases, simple snippets extraction is sufficient. However, for other cases, partial answers from multiple documents may have to be combined [HG01]. For the TREC QA evaluations, the style of result that most systems generate is a ranked list of the top five answers, where each answer is a text passage.

Accuracy is a major concern, for QA systems. In fact, precision is extremely important, as incorrect answers are worse than no answers [HG01]. It is pertinent to create measurements for answer evaluation, where a very low value may indicate the nonexistence of accurate information where the answer may be found.

More sophisticated users expect answers from outside the scope of text documents or structured databases. To upgrade a QA system with such capabilities, reasoning components and search engines that operate on several knowledge bases have to be integrated. Moreover, searching within intranets and along web repositories of knowledge it is also relevant expansion focus.

Finally, qualified QA systems have to handle properly with contradiction resolution, multiple alternatives and interpretation [BC02].

2.3. Information Retrieval Systems

"Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it."

Samuel Johnson [15]

2.3.1. Information Retrieval Overview

For thousands of years people have realized the importance of storing and finding information [Sin01]. For roughly 4000 years, mankind have structured information for later retrieval and usage [BR99].

Nowadays, the volume of electronic text available is straggling. Collections of documents stored on personal computers and corporation servers are growing larger as disk spaces becomes cheaper and electronic content becomes easier to produce, download and store. In the course of time, finding and extracting meaningful information from such sets became a necessity.

Data retrieval is concerned on spotting which documents of a collection include the keywords that the user specified in his query. However, Information Retrieval (IR) systems attempt to retrieve information about a subject, rather than just retrieving raw data which satisfies a given query [BR99].

As Manning et. al. explained in a broad sense, "*Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)*" [MRS08].

IR is a field that has been developing alongside with database systems for many years. However, contrasting the database topic which is focused on queries and transactions of structured data, IR is concerned with the organization and fetching information from large collections of unstructured documents. Typical examples of IR systems include on-line library catalogue systems and document management systems.

On the whole, IR systems must somehow match the user information need with the contents of the information items from the collection, and rank the items according to a degree of proximity to the user query. For that reason, IR plays a fundamental role meant for satisfying information demands. As users are interested in the information buried within documents and not the documents itself, pertinent methods are provided to dig information from large collections of documents; plus presenting information accordingly to the relevance to the user.

One of the major challenges that IR systems face comes from the fact that most of the information available is unstructured, that is, was not deliberately created to be straightforwardly indexed, stored and retrieved. With the amount of information worldwide

[15] ***Samuel Johnson**: British essayist (1709-1784)*

growing exponentially and the current digital content in the world estimated on 800,000 million gigabytes [Wra10], traditional IR techniques are not enough for the increasingly information demands. Typically, only a small part of the many existing documents will be relevant to a given user and without previous knowledge what could be within the documents, it is hard to devise effective queries to extract handy information from the text documents. Besides, users need frameworks to compare different documents, rank the importance of the documents and search for patterns across multiple documents.

Later, describing user information need is not a trivial job. Usually, this information need is presented to the IR system as a query which typically consists of an array of keywords. However, tracing information that matches to a question is not an easy task. Information requests can be formulated in many ways and cannot be parsed unambiguously every time Therefore, the system needs to be aware of the context in order to correctly interpret the meaning of the questions and respond accordingly.

Other important issue is that answers may be found in multiple sentences, paragraphs or documents, so proper mechanism need to handle such situations in order to present the user with an appropriate answer.

As we have seen, IR systems use matching mechanisms to evaluate how closely a document is related to the user's query [LC02]. The IR matching mechanism is described through retrieval models:

Boolean Model

Early IR systems were Boolean models which allowed users to specify their information need using a composite mixture of Boolean *AND*s, *OR*s and *NOT*s. Boolean systems have several weaknesses, e.g., there is no inherent notion of document ranking, and it is very hard for a user to shape a good search request. Even though, Boolean systems usually return matching documents in some order, e.g., ordered by date, or some other document feature. Although the research community has shown that Boolean systems are less effective than ranked retrieval systems, many users still use Boolean systems as they feel more in control of the retrieval process.

Ranked Retrieval

However, most everyday users of IR systems expect IR systems to perform ranked retrieval. IR systems rank by their estimation of the usefulness of a document on behalf of a user query. In other words, ranking is sorting the retrieved documents in a way that (hopefully) reflects the relevance of the documents against the user information need [BR99]. Most IR systems assign a numeric score to every document and rank documents by this score. Several models have been proposed for this process. The three most used models in IR research are the

vector space model [SWY75], the probabilistic models [RJ76], and the inference network model [TC99].

Vector Space Model

In this model, queries and documents are represented as weighted vectors. Text is represented by a vector of words, where which word is an independent dimension in the high dimensional space. If a term belongs to a text, it gets a non-zero value in the text-vector along the dimension corresponding to the term. Since any text contains a limited set of terms and the vocabulary can be millions of terms, most text vectors are very sparse. The vector model with *tf-idf*[16] term weights is a good ranking strategy with general collections.

The main advantage is that because term weights are not binary, a document is retrieved even if it matches the query terms only partially. Moreover, documents are ranked between their similarities with the query.

Yet, there are some drawbacks about Vector Space Model. First, it is very calculation intensive, requiring a lot of processing time. Second, each time we add a new term into the term space we need to recalculate all vectors. As pointed out by [LCS97], computing the length of the query vector requires access to every document term, not just the terms specified in the query.

Probabilistic Model

Given a user query q and a document d_i, the probabilistic model tries to estimate the probability that the user will find the document d_i relevant. The model assumes that this probability of relevance depends only on the query q and the document representations. Ideal answer set is referred to as R and should maximize the probability of relevance. Documents in the set R are predicted to be relevant.

This model presents documents ranked in decreasing order of probability of relevance.

However, some disadvantages arise from this model. In fact, this method does not take into account *tf-idf* factors, and the terms are not weighed.

Inference Network Model

Inference network modules are based on deductive relationships between representations of documents and information needs. They provide an epistemological view of the IR problem [BR99]. An inference reasoning process which estimates the probability that one or more queries, met the users information needs given a document. Many Bayesian inference networks are closely related to inference networks.

[16] ***tf–idf***: *term frequency–inverse document frequency.*

There are other IR modules such as Fuzzy Retrieval [LF88] and Latent Semantic Indexing [Dee88].

2.3.2. How to build an Information Retrieval System

Modern Information Retrieval (IR) systems have been applied in various environments. Some of them work as embedded components inside an existing tool searching very specific content, others run as Web search engines on a dedicated server infrastructure, and others run inside a company's intranet and search over a massive collection of documents.

Yet, despite all this diversity, typical IR architecture follows the subsequent key steps: Indexing, Searching and Ranking [BR99]:

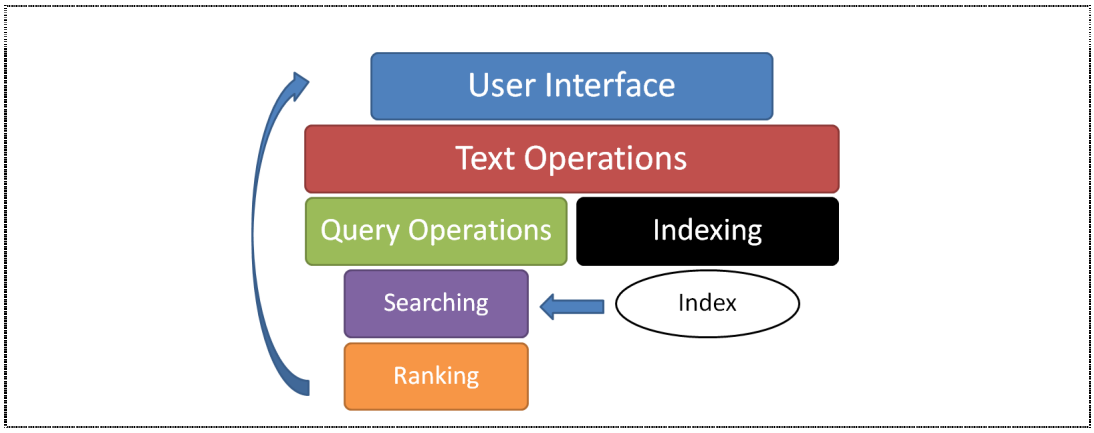

Figure 3 - Typical Information Retrieval architecture

The usual steps carried out by typical IR system are described below:

Document Analysis and Indexing

Document analysis is the task from converting text fields from documents, into the most basic index representation, the index.

Here is a list of standard pre-processing tasks that the IR analyser performs:

- Lexical analysis of the text, in order to handle digits, hyphens, punctuation marks and case letters, lower/upper case letters.
- Stop-words[17] removal, in order to filter words which are common and with very low discrimination values for retrieval purposes. These words usually contain

[17] ***Stop-words****: words that from linguistic point of view contain no information and only play a functional role.*

small information, and their use in a query would be unlikely to filter any document since they are expected to be present in almost every document. This task reduces the size of the indexing structure considerably.

- Stemming words, that is, reduce the word to the morphologic root. Prefixes and suffixes are removed, allowing the retrieval of documents containing syntactic variations of query items.
- Select which words will be used as an indexing terms. This judgment is directly related to the syntactic nature of the words. For instance, nouns have more semantic than adjectives and adverbs, so more relevance should be given to these terms.
- Sometimes synonyms are also identified and indexed under a single representative term. Thesaurus[18] can lend a hand in this task.

We move from a full text representation, to higher logical view over documents by index terms. Modern IR systems, especially web search engines might not use all text operators. The Figure 4 illustrates the logical view of the documents, over several steps in the indexing task [BR99].

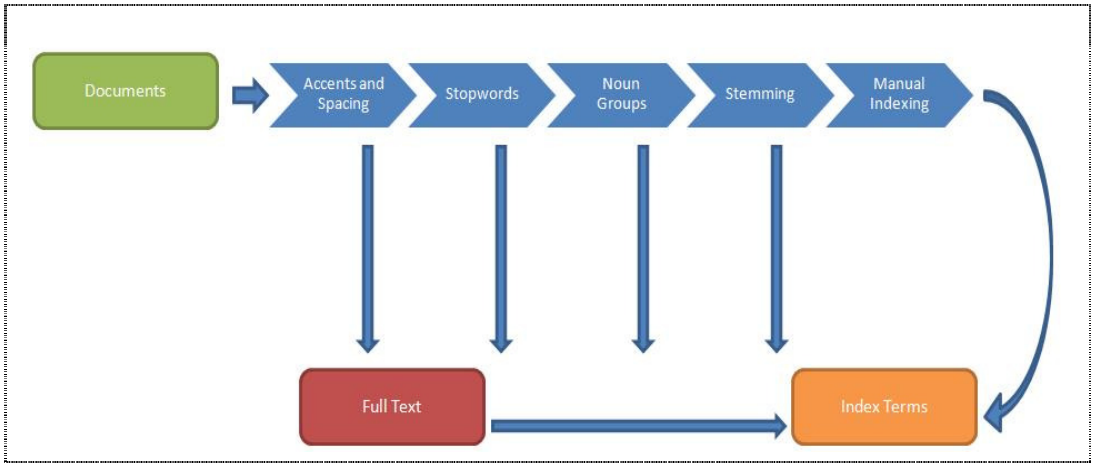

Figure 4 - Logical View of the Documents

Actual IR systems are based on *inverted list indexes,* where they maintain a mapping between terms and sets of documents. This data structure makes efficient use of the disk space, while enables fast access to full text documents that contain the term, along with other relevant information, such as weight and positions of the term in each document [WMB99]. Inverted indexes take advantage from the fact that IR systems are only concerned on scoring a few set of relevant documents that contain query.

[18] ***Thesaurus****: lists words grouped together according to similarity of meaning.*

Searching

"*If we can't find it, it effectively doesn't exist. Even if we have indexed documents, our effort is wasted unless it pays off by providing a reliable and fast way to find those documents*." [GHM05].

Searching is the method of looking up words in an index to find documents where they appear. The search quality typically described using *precision* and *recall* metrics. *Recall* measures the fraction of relevant documents found, while *precision* measures the fraction of irrelevant documents filtered [MRS08].

IR tools provide straightforward and easy search capabilities to look up for information under a repository of millions of documents. Typical features include returning the most relevant documents first and fast retrieval of results.

After the query parser process the natural language question expression, it tries to convert it into an internal representation of a query. That query has some requirements that will constraint the search space. Then, the results are retrieved after the index database being searched by those constraints.

Ranking

In order to meet user's information requirements, one of the most sophisticated features that IR systems provide is the ability to retrieve the most relevant documents first.

When the IR engine is queried, an ordered collection of hits is returned following some evaluation function. Hits are a collection of documents ordered by score, a numerical value of relevance for each document, given a query. Hence, relevance measures the distance between the query representation and the logical representation of the documents.

In the most applications that retrieve ranked results, users only access the first documents. Sometimes it is relevant to sort the documents collection by date or by some particular field (e.g. book's title).

There are some key techniques that support document ranking enhancement, such as:

- **Term Weighting:** Various methods for weighting terms have been developed in the field. Weighting methods developed under probabilistic models rely heavily upon better estimation of probabilities [SB76]

- **Query Modification with synonyms:** In the early years of IR, soon researchers realized that it is hard for users to formulate effective search queries. It was considered that adding synonyms of query words to the original query should enhance search. Early research in IR relied on a thesaurus to find synonyms [Spa71].

- **Relevance feedback for Query Modification** Relevance feedback is motivated by the fact that it's easier for users to judge the documents retrieved as relevant or

non-relevant for their query. Using such relevance evaluations, a system can then automatically refine queries (e.g., by adding related new terms) for further searching [Roc71]. Machine Learning techniques are also applied to expand the queries [CQL07].

2.3.2. Passage Retrieval

"I want minimum information given with maximum politeness."
Jackie Kennedy[19]

With the hasty growth of information resources and the colossal amount of requests submitted to IR Systems, one of the most challenging progressions is to deliver the best related excerpts or passages from documents, regarding user's information need. Sometimes, it's better to apply retrieval algorithms to portions of documents or passages, rather than all the full document text [Cal94].

Passage Retrieval (PR), compared to traditional full document retrieval has not been researched as much great extent detail [TK03]. Still, PR can be especially pertinent when the documents are long or span different topic areas [Cal94]. Moreover, this approach could be very interesting, as evidence shows that users usually prefer passage sized answers over whole documents, as it gives context [LQ03].

Passages are a significant intermediary concerning full documents and strict answers. If the passage is ranked, users can be rapidly driven to the relevant information within the document.

In actual fact, nearly almost QA systems and summarization techniques put into practice some method for extorting textual paragraph-sized snippets from huge documents corpus [TK03] . Besides, almost all QA systems fielded at TREC use some PR method to decrease the complexity of the relevant document set to a manageable quantity of passages, reducing the burden of reading lots of irrelevant documents in order to achieve the required information [SAB93].

The types of passages can be classified into three major classes:

- **Discourse Passages**: based upon textual discourse units, such as sentences, paragraphs, and sections.
- **Semantic Passages**: based upon the subject or content of the text.
- **Window Passages**: based upon the number of words or sentences.

Several passage PR methodologies have been proposed in the QA context. One point from discourse passages is that they could be more effective, as discourse boundaries organize material by content [Cal94]. Here is a brief overview some of the available methodologies.

[19] ***Jackie Kennedy****: wife of the 35th President of the United States, John Fitzgerald Kennedy*

Light

Light et al. ranked passages considering the number of keywords a passage has in common with the user query [LM01].

TextTiling

TextTiling is a technique that subdividing texts into semantic passages. The linguistic clues used for identifying major topic shifts are patterns of lexical co-occurrence and distribution. The algorithm is shown to produce good segmentation by subtopics [Hea97].

Clarke

Clarke used a density based passage retrieval method which benefits small passages enclosing several terms with high *td-idf* values. In this method every passage begins and ends with a query keyword [Cla97].

Gonzalez

Gonzalez considered non-length normalized cosine relationship among query and the passage. Keywords are weighted considering how many times they appear in the passage and in the query, as well as also their *td-idf* values [Gon01].

Cui

Cui et al. explored the use of fuzzy dependency relation matching between query terms and terms within passages. This approach introduces some improvements, when compared to the density based passage retrieval approaches [CS05].

IBM

IBM's passage retrieval algorithm that compute several distance measures for the passage, which in the end are linearly combined to give the final score [IF00]:

- The "*matching words measure*" sums the *td-idf* values of words that appear in both the query and the passage.
- The "*thesaurus match measure*" sums the *td-idf* values of words in the query whose WordNet[20] synonyms appear in the passage.
- The "*mismatch words measure*" sums the *td-idf* values of words that appear in the query and not in the passage.
- The "*dispersion measure*" counts the number of words in the passage between matching query terms,

[20] ***WordNet**: lexical database of English words, providing relationships between them.*

- The "*cluster words measure*" counts the number of words that occur adjacently in both the question and the passage.

Alicante IR-n

Alicante's passage retrieval algorithm proposed by [LV01] computes the non-length normalized cosine similarity between query terms and the passage. It takes into account the number of appearances of a term in the passage and in the query, along with their *td-idf* values.

2.3.3. Retrieval Evaluation

IR has developed as a highly empirical discipline, requiring cautious evaluation in order to make obvious the better-quality performance of new techniques [MRS08].

The type evaluation depends on the objectives of the IR system. When measuring how good IR systems perform, the key measure is user's satisfaction. Speed of the response, size of the index database or style of the answer are factors that make users happy, however the crucial point is to retrieve relevant results.

Following the standard approach proposed by [MRS08], in order to measure IR systems some conditions have to be set up:

- Large document collection that will serve as input
- Set of users information needs expressed by queries
- Set of relevance judgments assessed as relevant or non-relevant for each query-document pair.

The evaluation of *ad-hoc* IR systems revolves around the notation of relevant and non-relevant documents. With respect to a user information need, a document in the test collection is given a binary classification as either relevant or non-relevant. The number of information needs have to be of a reasonable size, 50 documents is a good minimum [MRS08].

Relevance is assessed to the information need, not to a query. A document is relevant if it addresses the stated information need, not because contains all the words in the query. One of the main challenges is that from one keyword is very hard to define what the information need is.

The information needs are better designed by domain experts, as using random combination of queries do not resemble the actual distribution of information needs. Given a set of queries resembling the information demands, an evaluation needs to be done in terms of relevance or non-relevance. The most standard approach of evaluation is *pooling*, where relevance is assessed over a subset of the collection that is formed by the top *k* documents returned.

Typical research IR systems evaluate their systems using standard test collection such as Cranfield, TREC, CLEF[21], or magazine articles.

[21] ***CLEF****:* http://www.clef-campaign.org/

To measuring unranked retrieval sets, *precision* and *recall* are the most important measures.

- ***Precision***: Measures the fraction of retrieved documents that are relevant.

$$Precision = \frac{\#relevant\ items\ retrieved}{\#\ retrieved\ items}$$

Equation 1 – Precision formula

- ***Recall***: measures the fraction of relevant documents that are retrieved.

$$Recall = \frac{\#relevant\ items\ retrieved}{\#\ relevant\ items}$$

Equation 2 – Recall formula

It is conventional that a high-quality IR system should retrieve as many relevant documents as possible (i.e., have a high recall), and it should retrieve very few non-relevant documents (i.e., have high precision). However, these two goals have proven to be contradictory over the years as techniques that tend to improve recall, degrade precision and vice-versa [Sin01].

A single measure that trades off precision versus recall is the *F* measure, which is the weighted harmonic mean of precision and recall:

$$F\ (\beta = 1) = \frac{2PR}{P + R}$$

Equation 3 – F-measure formula

However, *Precision*, and *F-measure* are set-based measures. They are computed using unordered sets of documents *Recall*. In a ranked retrieval context, a *precision-recall* curve can be plotted.

Moreover, it's often to compute the *mean average precision* (MAP) as contains both recall and precision aspects and is sensitive to the ranking provided by the IR system. For a single information need, *average precision* is the average of the precision value obtained for the set of top *k* documents retrieved. For the set of relevant documents for an information need $q_j \in Q$ is

$\{d_1, \ldots, d_{mj}\}$ and R_{jk} is the set of ranked retrieval results from the top result until reach the document d_k, then the equation to calculate MAP is:

$$MAP = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{mj}\sum_{k=1}^{mj} Precision(Rjk)$$

Equation 3 – Mean average precision formula

2.3.4. Technologies Outline

When deciding to incorporate a search engine within the organization, there is the possibility to use a commercial search engine or an *open-source* one. Using a commercial search engine sometimes is not the most feasible solution as requires high fees and licences. On the other hand, *open-source* engines may give the same functionalities as commercial solutions, with the benefits of being free of charge, the software is maintained actively, and components could be customized and improved in order to meet the company's business reality.

Nowadays, there are many suitable *open-source* choices, and having each of them different features that must be taken into thought. Although almost IR tools provide benchmarks reports, they are too focused on speed instead of ranking mechanisms. Other important factor to consider about *open-source* tools is the last-date-of-update of the software and the current activity of the project. Thereafter, it is also essential to consider performance issues with diverse sizes of information.

Here is an overview over some relevant Open-Source Search Engines:

- **Indri/Lemur**: Indri is a search engine built on top of the Lemur modelling toolkit [Ind09] [Lem09]. It was developed as a platform for research with ranking algorithms. What is more, this is a dual project between University of Massachusetts and *Carnegie Mellon University*, USA. It is released under a BSD-like license.

- ***Lucene***: *Lucene* is a mature text search engine library, being part of the Apache Software Foundation [Luc00]. Since it is a library, there are some applications built on top of it, e.g. Nutch, and Solr [Nut09] [Sol09]. *Lucene* has widespread industry adoption. It is used by *Technorati*, *Monster*.com's resume search. It provides search capabilities for the *Eclipse IDE*, the *Encyclopaedia Britannica* CD-Rom/DVD, *FedEx*, *Hewlett-Packard* [GHM05]. It is released under an *Apache* Software license.

- **MG4J**:"*Managing Gigabytes for Java*" is a framework for full text indexing for large collection of documents, based on the classical inverted-index approach [MG405]. It supports flexible scoring schemes and optimized classes for processing strings. Moreover, the index construction is highly configurable. It was developed at the University of Milano, Italy. It is distributed under the GNU GPL license.

- **Terrier**: "*TERabyte RetrIEveR*" is a modular platform that allows rapid development of search engines for large scale retrieval [Ter04]. It allows a flexible relevance ranking method implementation. Despite of being originally designed as a research platform for relevance ranking methods, Terrier is used as an academic or commercial engine. However, Terrier does not support incremental indexing. It was developed at the University of Glasgow, Scotland. It is released under the Mozilla Public license.

- **Zettair**: is a text search engine developed by the Search Engine Group at Royal Melbourne Institute of Technology University, Australia [Zet04]. Its key feature is the ability to handle large amounts of text. The main focus is the research on the performance and scalability of search systems. However it does not allow multithreading. It is distributed under a BSD-style license.

There is a common array of features among this search engines. All of them support phrase search, Boolean search and wildcard search varieties. Moreover, the analyzer also performs typical natural language pre-processing tasks before indexing, such as stop-words removal and stemming. What is more, all the tools are able to deal with large scale of full-text documents. After all, rank sorting is supported as well by every search engines.

This is not a full list of search engines. There are many others research search engines that also disserve attention such as *Galago* [Gal09], *Wumpus* [Wum05] and *Minion* [Min08]. Moreover, *Xapion* [Xap03] is a good industrial search engines reference. This search engines were not included in the overview above because there was no relevant research documentation published, or else relevant comparisons among other search engines or the project was not mature enough.

The Table 1 presents a comparison between five different Open-Source search engines, described below. This comparison was based on the work done by Middleton and Baeza-Yates [MB07], supported via the work done by Dalton [Dal09] and Singh [Sin09], in addition to the information available from each search engine's website. It is noteworthy that all the systems were tested out-of-the box, without any tuned settings. The experiences were conducted with a 2.7 GB collection of documents as input for the indexer. For the comparison of the search times was considering queries with one and two words, with uniform distribution.

- **Last Updated**: Last time that the project was modified. This is very imperative to measure, since a search engine that does not show activity in recent times does not

present further improvements. This may be a severe concern, as it may present problems at the moment of adapting the system to the business requirements.

- **Licence**: Type of licence that the project was distributed.
- **Lang**.: Core programming language of the project.
- **Port**.: Portability of the project. This feature was measured in terms of the number of successful port languages, how many different platforms the system was tested and the degree of modularity of the architecture.
- **Index time**: Time that the search engine takes to index the documents collection. Based mainly on the research done by Middleton and Baeza-Yates [MB07].
- **Index size**: Final size of the index files after the search engine index the documents collection. Based mainly on the research done by Middleton and Baeza-Yates [MB07].
- **Search time**: Time that the search engine takes to search and retrieve a query. Based mainly on the work research by Middleton and Baeza-Yates [MB07].
- **Use**: Typical nature of purpose use for the system.
- **Support**: Degree of support of the search engine, when problems arise. This feature was measured in terms of the size of the active community, availability of discussion groups and forums, the degree of maturity of the project, availability of APIs, Issues Tracks, etc.
- **Concurrency**: Ability of the system to run several threads in parallel.
- **Ready Deployable**: If the system is ready to be deployed in the final environment or not. The system is classified as ready deployable if is an all set featured search application, and not only a software toolkit.

	Last Update	Licence	Lang.	Port.	Indexing Time	Index Size	Search Time	Use	Support	Ready Deployable
Indri/Lemur	Dec 2009	BSD	C++	3/5	Good	Bad	Very Good	Research	3/5	Yes
Lucene	Feb 2010	Apache	Java	5/5	Bad	Very Good	Very Good	Industrial	5/5	No
MG4J	Jun 2009	GLP	Java	2/5	Good	Bad	Very Good	Research	2/5	No
Terrier	May 2010	MPL	Java	2/5	Ok	Ok	Ok	Industrial Research	3/5	Yes
Zettair	Mar 2009	BSD	C	2/5	Very Good	Good	Good	Research	2/5	Yes

Table 1 - Comparison between Open Source Search Engines

In summary, *Zettair* is a good solution for very large scale of information, as it performs fast searching and indexing, and it is ready to be deployed application. However, it lacks of support.

Other relevant tool is *Lucene,* which is one mature framework that suits environments with disk size constraints. However, it has some drawbacks, as it takes too much time to index all the documents collection. If the project does not change too much, i.e., does not require frequent re-indexing, *Lucene* is a good solution for medium scale deployment.

1.4. Genetic Algorithms

"Evolution is nature's mistake. Intelligence is its insistence on making the same mistake."

S.LEM, GOLEM XIV, 1981

2.4.1. Genetic Algorithms Overview

The fundamental principles of Genetic Algorithms (GA) were proposed by John Holland in the early 1970s. GA is inspired by the biological natural selection mechanism where stronger individuals are expected to be better adapted to the world and survive in a competing environment. In contrast to other evolution strategies, Holland's original endeavour was not to develop algorithms to solve problems, but revise the phenomenon of adaptation in the natural environment and to overtake those mechanisms of natural adjustment to computer programs [Hol75].

In fact, GA supposes that a possible solution to any problem can be represented by an individual or chromosome, and be characterized by a set of parameters. These parameters, regarded as the genes of a chromosome, can be structured as a string of binary values. Besides, a positive value, known as the fitness, is used to imitate the degree of goodness or adaptability from the chromosome to the problems.

Throughout genetic evolution, a superior chromosome has the predisposition to breed fine quality offspring. Good chromosomes mean better solutions to problems. In a practical GA application, an initial population of chromosomes has to be set up, many times randomly. The dimension of the population varies accordingly to problem.

Then, each generation is as an evolution process, as the following generation is created from the chromosomes from the previous population. This evolution however can only succeed if a group of these chromosomes, the "*parent*" chromosomes is selected via a specific selection strategy. Many times, this selection is made in terms of choosing the best *n* chromosomes for reproduction. The genes of the parents are mixed and recombined for the production of offspring in the next generation. It is expected that from this course of action of development and evolution, the "*superior"* chromosome generates an extensive quantity of children. So, this "*superior"* chromosome has a privileged prospection of surviving in the consequent generation, reproducing the survival-of-the-fittest-one mechanism in natural biological environment. [MT99].

In short, Figure 5 shows the overall process from GAs, based on the process described by Michalewicz [Mic96].

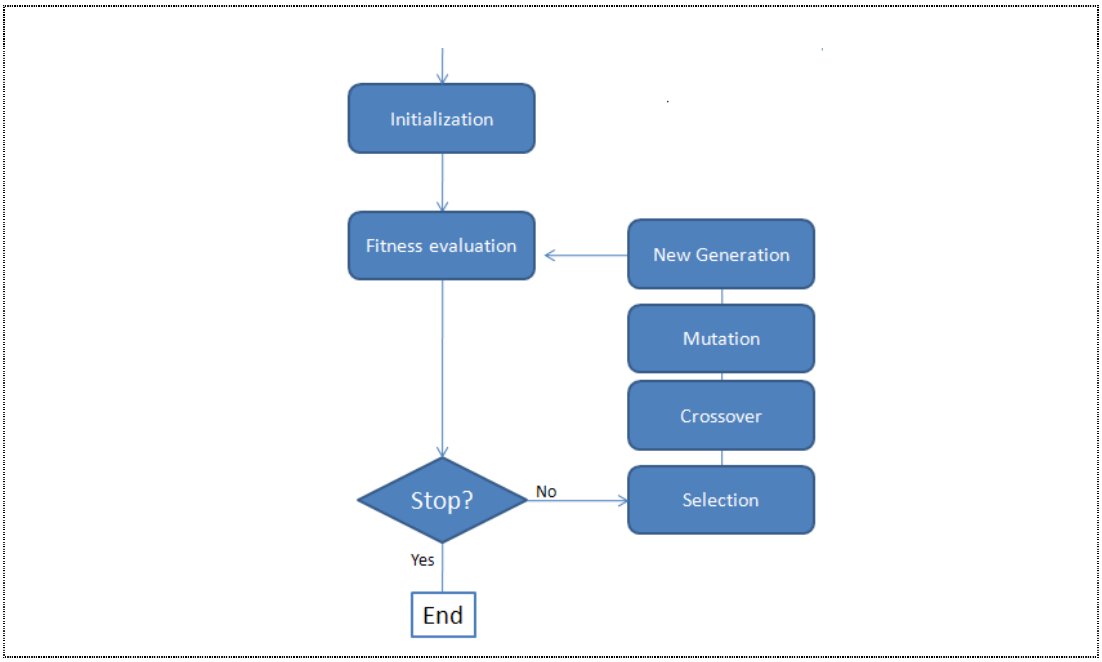

Figure 5 - Genetic Algorithm mechanism

Headed for the solution, many computational problems require searching through a huge amount of possibilities. Mitchell states that "*if the search space to is very large, or is not well understood, or if the fitness function is noisy, and if the task does not require a global optimum to be found, a GA will have a good chance of being competitive with or surpassing other weak methods of searching*" [Mit98].

GAs have been used for a wide variety of optimization tasks, such as scheduling [CS89], finances [YM94], combinatory optimization [KBH94], imaging processing [Ala98] and medicine [CD99]. GAs also have been used to evolve aspects of particular machine learning systems, such as weights for neural networks, rules for learning classifier systems or symbolic production systems, and sensors for robots [Mit98]. Besides in the Economics field, GAs have been used to model processes of innovation, the development of bidding strategies, and the emergence of economic markets [Mit98]. GAs have been also used in social systems, ecology and genetics [Mit98].

2.4.3. Genetic Algorithms for Target Optimization

"*Our lives confront us with many opportunities for optimization. What time to get up, what is the best route to work...When we design something, we shorten the length or reduce the weight as we want to minimize the cost or maximize the appeal of a product"* [Hau04].

Along with the increasing length of full-text collections, more attention was paid on better passage ranking algorithms. Actually, Genetic Algorithms (GAs) can provide an efficient search strategy for optimizing a target in larger spaces with unknown landscapes [Tie07]. Moreover, they have been used earlier in information retrieval field, for document clustering [Gor88], relevance feedback [YK93], and keywords optimization [Che95].

At the present time, users of search engines such as *Google* or *Bing* are educated to polish up their keyword queries, in order to get close the required result; comparing to Question and Answering (QA) system that does not get more information than a regular and well structured natural language question [Tie07]. Therefore, Passage Retrieval units have to make use of all the linguistic clues from questions, in order to get closer possible candidate solutions in a faster and smarter way. Blair & Maron showed that it is very difficult for users to predict the exact words or phrases used by authors in desired documents [BM85].

A traditional QA system usually analyses and processes the question before running the IR system. As a result, useful information for the following components could be extracted straight away, such as the question type and other relevant linguistic features from the source text. Nonetheless, suitable NLP mechanisms have to be incorporated into the QA system, with the purpose of extract pertinent syntactic dependency relations between terms, dig out relevant entities (name-entity recognition, e.g. identify persons, locations, organizations, etc) in addition to stem words. Besides, synonyms networks could also play an important role in this process, as they provide valuable semantic relations between analogous concepts. In short, it is essential to extract *a priori* several features that will lend a hand to the passage retrieval unit, meeting the needs of the QA system, fulfilling better the user's needs.

In actual fact, NLP tools are frequently used in QA systems, although exploiting linguistic features for passage retrieval enhancement is not so common [Tie05]. Within Passage Retrieval context, GAs can take action as a support component for several stochastic optimizations. As most of the IR systems allow several operators in their search query parsers (e.g. boost keywords, set keywords as required, etc), GAs could enhance the selection of the most relevant features, best operators and proper weighting when searching the IR index.

Still, very few researchers have tried to use evolutionary algorithms like genetic algorithms in the area of Information Retrieval [PGF00]. The main research issue is to come upon suitable features that can truly assist passage retrieval enhancement. Nevertheless, an evolutionary approach such as GAs is expected to be a suitable framework for combing several query keywords and experiment various features to progressively refine the retrieved results.

2.5. Summary

Along this chapter was made an overview over the basics and the main challenges from Knowledge Management, Question and Answering Systems, Information Retrieval Systems and Genetic Algorithms.

With the advent of digital content and information overload, it was seen that Knowledge Management systems play an important role within organizations.

Question and Answering systems are efficient interfaces to knowledge, which allow consultants from enterprises to achieve more informed actions. Typical Question and Answering systems implement an Information Retrieval system, along with a Passage Retrieval Mechanism. Passages are an efficient way of response to users, as it gives short relevant answers with context to the users.

What is more, Information Retrieval systems provide broad-based storage to large amounts of information, along with features that allow users to efficiently find and receive information that's relevant to them.

There are many open source Information Retrieval tools in the market; however *Apache Lucene* seems to be the mainstream trend.

Finally, Genetic Algorithms can provide an efficient search strategy as they can refine queries. They lend a hand to Question and Answering systems as they enhance the relevance of the passages retrieved.

Chapter 3

Implementation

This chapter makes a broad description of the proposed solution and methodology to follow in order to achieve the planned goals.

In summary, this section starts by explaining *Prymas* Knowledge Management system architecture and how the several modules are interconnected. Secondly, the Information Retrieval (IR) System design is presented in particular, showing how it supports Knowledge Retrieval module. Moreover, within this section, special relevance is given how to build an IR system using *Java Apache Lucene* framework, plus how to extend a standard *Lucene* IR system to a Passage Retrieval System.

Finally, it is explained how to integrate a Genetic Algorithm within *Lucene*'s query mechanism, in order to enhance Question and Answering retrieval results.

3.1 *Prymas* Central Knowledge Management System

The project described all along this manuscript will run as part of an overall project of creating a central Knowledge Management system, designated *Prymas*. *Prymas* is intended be one answer to the knowledge management problem within organizations, supporting better management practices on large collections of knowledge assets, while allowing a more effective and quickly interaction with knowledge.

3.1.1 *Prymas* Architecture Overview

Prymas is premeditated to support all the knowledge development cycle, allowing recognition, representation, construction and distribution of knowledge, enabling adoption of the best practices and solutions all along the organization, plus a closer engagement of individuals with business processes. Corporate employees will have the opportunity to continuously contribute with knowledge and access information from a central knowledge repository.

As Figure 6 shows, *Prymas* is divided into three key modules; Knowledge Acquisition, Knowledge Discovery, and Knowledge Retrieval.

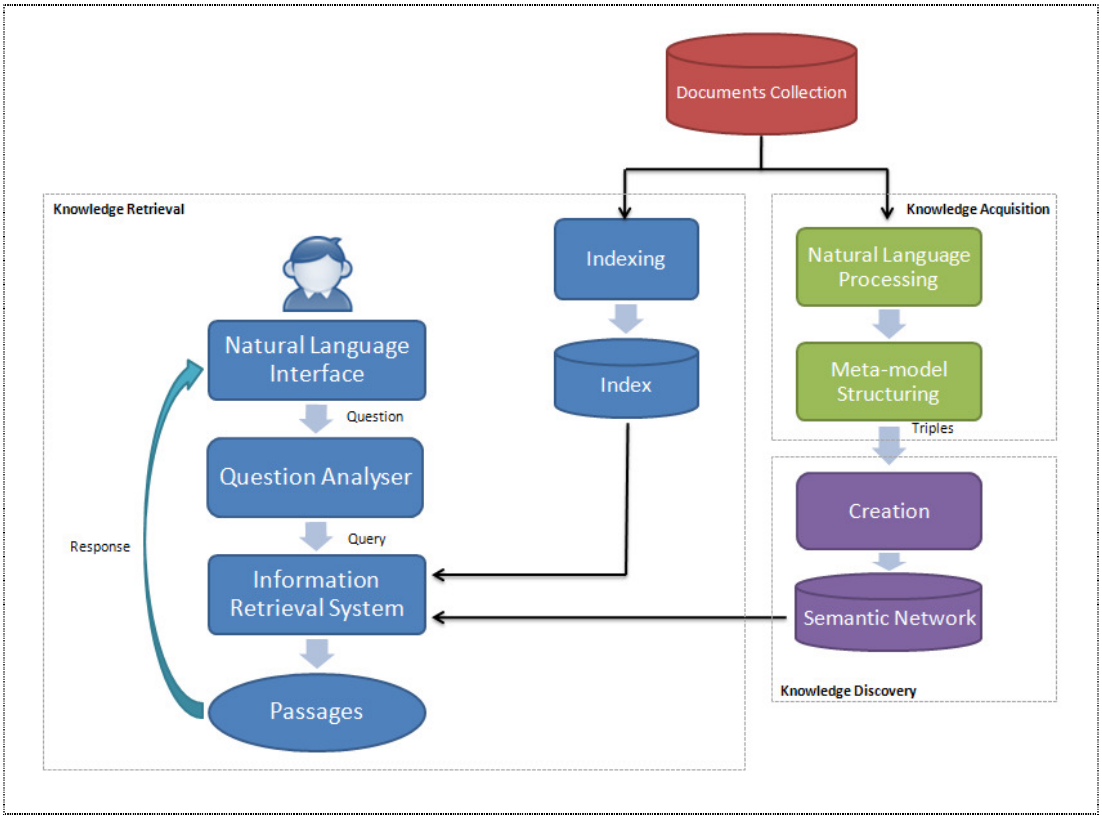

Figure 6 – Prymas macro-architecture

Prymas includes an important Knowledge Acquisition module that analyses the collection of source documents by Natural Language Processing[22] (NLP) techniques, processing the knowledge into an intermediary representation that facilitates the relationships construction by the semantic network system from the Knowledge Discovery module.

Additionally, the same documents collection is also indexed by an Information Retrieval (IR) module. *Prymas* also comprises an interface closer to Question and Answering (QA)

[22] ***Natural Language Processing***: *field concerned with the interactions between computers and human natural languages.*

systems, where users interact with knowledge by the way which is more natural to them. The QA infrastructure is built on top of an IR system, which allows searching relevant passages over the documents collection.

However, following the view over the dissertation project, the contribution of this report will be focused single on the Knowledge Retrieval module, more precisely, on the IR system.

3.1.2 Knowledge Acquisition Layer

The first module from *Prymas*, Knowledge Acquisition, is concerned with purchasing and interpretation of information from unstructured sources, such as text documents and technical reports. Then, through text extraction techniques a semi-structured meta-model is produced, comprising an in-between-representation from data and knowledge, as it already includes some semantic relationships between concepts.

This process includes all the phases from the conversion of the raw text lines from one sentence, to a representation in forms of triples that encode information about the subject, predicate and object.

As we could see in the Figure 7 the knowledge acquisition runs as it follows:

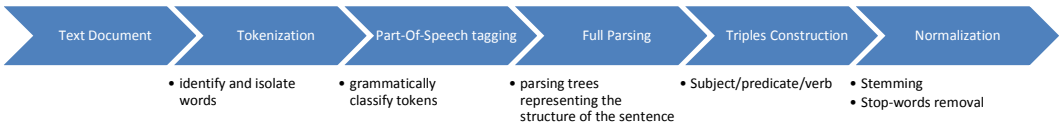

Figure 7 - Knowledge Acquisition Flow

In short, the Knowledge acquisition process comprises application of Natural Language Processing mechanisms in order to interpret and transform the unstructured information into structured and explicit relationships. This task uniforms information so that can be manipulated and exploited by automatic computer mechanisms.

3.1.3 Knowledge Discovery Layer

The second module from *Prymas*, Knowledge Discovery, is responsible to explore knowledge through analysis and association methods. This is supported by the automatic construction of a semantic network where relationships between entities are mapped into a

network. The input for this module is the output meta-model representation from the previous Knowledge Acquisition module.

This form of knowledge representation is embodied in a graph, where the vertices are the concepts, and the edges the relationships between the concepts. Each relation in the network is a representation of a fact that is classified by a relation type. Moreover, this module uses a generic ontology which supports hierarchical organization of knowledge. Ontologies allow mechanisms of reasoning and inference, which allows exploitation from the facts web and knowledge discovery [BCM04].

As we can see in the Figure 8, this *Prymas* architecture includes an important layer of knowledge instantiation and enhancement.

Figure 8 - Knowledge Discovery flow

From a triples representation, the Knowledge Discovery layer is responsible for processing the relations and validating the actual tuples. This means that after being compared with the actual knowledge, relations are mapped into a semantic network. Contradictions detection and duplicates removal is a very important challenge in this module, as the documents are written in natural language.

3.1.4 Knowledge Retrieval Layer

Knowledge Retrieval layer comprises the interaction and communication between the user and the Question and Answering System. It embraces all the phases from one natural language question from a user, to the final response also in natural language. Question and Answering (QA) systems grant users the ability of easily inquire knowledge, requiring only a small learning curve of the system. This is a valuable feature, as we want to have a system that can be used not only by technicians, but also for a wide range of users without technical background.

As the Figure 9 shows, after the unstructured information from documents being indexed by the Information Retrieval engine, relevant passages from documents can be searched.

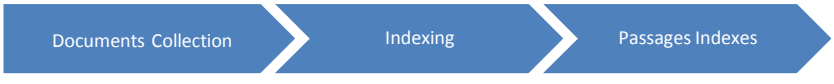

Figure 9 – Indexing Flow

Furthermore, Figure 10 presents the QA flow, from the user question until the final passage being shown as relevant response.

Figure 10 – Question and Answering Flow

This module performs a set of Natural Language Processing tasks by the question analyzer converting the user question into a query of terms that could be decipherable by the IR system. This whole process also involves, besides the identifying the key terms from a query, entity-recognition and extraction of synonyms. The system also does research on the net semantics for the key terms in order to provide additional information to the Information Retrieval system.

3.2. Developing an Information Retrieval system with *Lucene*

The following sections describe the implementation of an Information Retrieval system, within a Question and Answering context.

3.2.1 Why *Apache Lucene*?

Apache Lucene is an Information Retrieval library. Subsequently, it can supply indexing and searching technology to applications that require full-text search over large collections of documents. What is more, *Lucene* smoothes the progress of developing IR applications as it allows the focus on the business rules specific to problem domain, while hiding the complexity of indexing and searching mechanisms beneath the framework layer.

Lucene is the de-facto industrial standard for indexing and search, being widely adopted by open-source and commercial vendors, plus many websites (E.g. *Nutch, Solr, IBM OmniFind Yahoo! Edition, Apple iTunes, FedEx, Eclipse IDE, AOL, Digg, MySpace, LinkedIn, IBM, Wikipedia, Source Forge, Wolfram*) [Pow10].

Despite several IR tools with a wide range of features being available, *Java Apache Lucene* was used within this project as IR framework based on the following reasons:

- Efficient search capabilities over large collections of documents. It allows different types of queries, it has a relevant array of query operators plus supports ranked search;
- Open-source project implemented in *Java*, available for free download;
- Mature and robust project. Moreover, it's a member of *Apache* Jakarta family of projects, licensed under the *Apache* Software License;
- Very good support, with a very active developer community and fine documentation (APIs, books, articles, tutorials, etc);
- Simplicity of the core architecture. It's easy to set up the system and start running. Moreover, it can be modified very easily. There are several additional extension packages that can be ready integrated (Spellchecker, Highlighter, Stemmers etc);
- Portability to web languages (e.g. Microsoft's *.NET* framework). Good for future developments;

However, *Lucene* is a software toolkit, not a ready-to-use application like a desktop search engine tool, a web search engine, or a webmail crawler. It should be noted that *Apache Lucene* lacks the following capabilities:

- Indexing of *MS word*, *MS excel*, *MS PowerPoint*, *pdf*, *rtf* document formats;
- Automatic crawling of local networks and internet sites;
- Graphical User interface for searching, displaying search results, and installation and administration packages of the product;

However, these features can be incorporated making use of other open source projects or building custom J2EE[23] applications. The high customization level that *Lucene* supports, suits the project goals better as custom-made strategies can be pursuit in order to better fulfil the business demands.

3.2.2 Information Retrieval System Architecture

Information Retrieval systems, such as *Lucene*, endorse information overload problem. IR can actively sustain Knowledge Retrieval modules and Question and Answering systems, as IR tools support efficient mechanisms to search within large amounts of information.

The key components from *Lucene*'s IR library are the Analyzer, the Indexer and the Searcher. Figure 11 illustrates the base architecture from one standard IR system built on top of *Java Apache Lucene* framework.

[23] ***J2EE**: platform for server programming in Java*

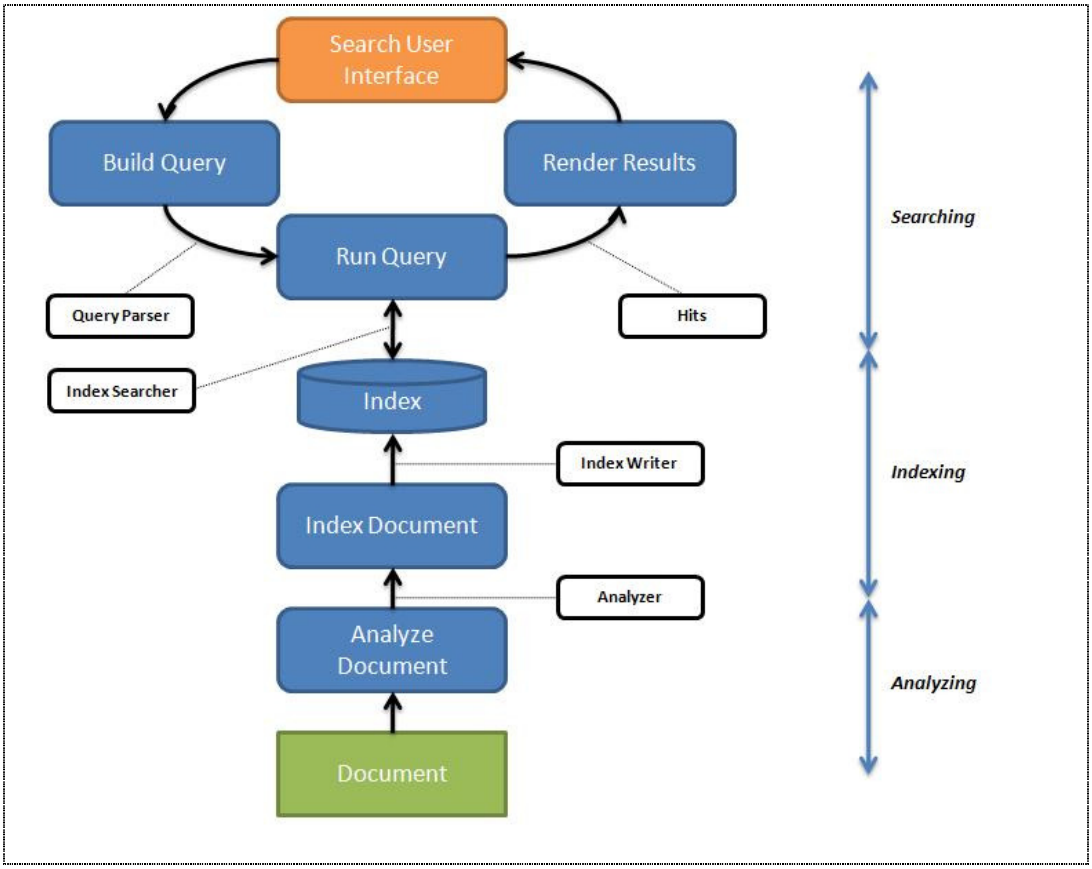

Figure 11 - Core Architecture of Apache Lucene

Afterward, all the steps from indexing a text document, to querying and retrieving results are described in detail:

a) Convert documents into plain text

To index information with *Lucene*, documents have to be previously converted into a stream of plain-text tokens (e.g. txt files). As mentioned before, *Lucene* does not supports direct indexing of PDF documents, thus first they have to be prepared for such task, for instance with an external library such as PDFDocument or Tika tools. Within this project it was used Tika toolkit to perform text conversion [Tik10].

b) Analyze text before indexing

Before all the words from text documents being indexed, *Lucene* first text has to go through an Analyzer in order to make it more suitable for indexing. In order to perform such task, several steps are done:

- Split the source textual data into individual atomic elements called tokens. Each token corresponds to a "*word*" in the language
- Performs several optional syntactic operations on the tokens. Typical syntactic tasks before indexing include:

- Lowercase tokens, in order to make searches case-insensitive;
- Collapse plurals into their singular form;
- Remove all frequent but meaningless from the text, such as stop-words (*a*, *an*, *the*, *in*, *on*, *and*, etc).
- Reduce input tokens them to their morphological root or base form, that is, word stemming.
- Break compound words into several individual words.
- Go through a spell checker
- Inject synonyms in the index in order to expand queries. As a result, querying by different words but with the same meaning can retrieve the same results.

Many implementations can be integrated with the Analyzer, however within this project, it was used the *Lucene*'s Standard Analyzer, which adjusts all the text to lowercase and removes common English stop-words.

c) Indexing words from documents

Following *Lucene* terminology**, *Lucene.Documents*** can be seen as virtual documents that can be retrieved later on. This flexibility allows ***Lucene.Documents*** to be independent from the original source or input file format. Hence, that means that it is irrelevant if the text comes from a *pdf* or *html* file, or else if it is a snippet from a book's chapter, or a report's section.

As we could see in the Figure 12, *Lucene* stores the input data into an inverted index data structure that contains one or more ***Lucene.Documents***.

Each ***Lucene.Documents*** is composed of one or more ***Lucene.Field*** objects. ***Lucene.Fields*** characterizes meta-data coupled with the document. This means that relevant data about author, path, title, last updated date, even the document's content itself can be indexed and stored separately as fields of documents. Soon after, each field can be either queried or retrieved from the index.

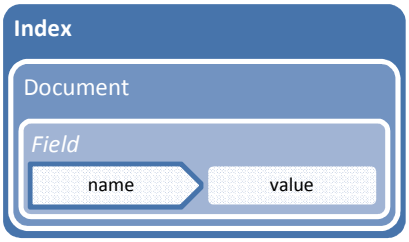

Figure 12 - Apache Lucene Index Representation

Within this project, the absolute path from the input documents was included into the ***Lucene.Document*** object as a ***Lucene.Field***. Since the value of this field is not useful for searching purposes is not indexed. However, it is stored in the index as it is useful for building the search results page, displaying the documents location to the users. Regarding document's content, all the words from it are previously analyzed and indexed. However, the text content isn't stored within the index structure for performance reasons, as for large collections the index

dimension could even exceed the physical computer memory. The Table 2 provides a summary of these settings.

Field	Analyzed?	Indexed?	Stored?
Absolute Path	No	No	Yes
Document's Content	Yes	Yes	No

Table 2 - Fields settings

After the input text been analyzed and converted into a proper stream of plain-text tokens, they are ready to be integrated in the index. The further steps describe the process to index words from documents:

- The index structure is built by *Lucene*. Indexes are stored as specific files on the system. ***Lucene.IndexWriter*** is the central component of index process.
- The directory file tree where all the documents to be indexed are stored is recursively processed in order to add all documents sequentially to the existing index.
- The absolute path from the document is stored within the index. Moreover, all the relevant words from the text content are indexed, so later they can be searched.
- For performance reasons, in the very end of the process the index is optimized. Optimizing the index merges all index files together in order to minimize resource consumption and improve search performance. This task will benefit specially applications that handle large indexes.

As a note**,** *Lucene* allows running multiple concurrent searches against the same index. However, when performing operations that modify an index, only one index-modifying operation should be run on the same index at a time.

d) Searching terms within documents index

Having a repository of hundreds of documents indexed with the same search requirements; it is time to integrate search mechanisms to the model. Searching is the process of looking up words within an index headed for finding documents where these words appear. After the index file being created, ***Documents*** can be searched. Here are the steps for searching terms within documents index:

- ***Lucene.IndexSearcher*** class opens the index for searching. Basically, ***Lucene.IndexSearcher*** is a central pointer to the index database that implements several search methods.

- Then, ***Lucene.QueryParser*** parses a human-readable query into *Lucene*'s query internal representation. Typically, the parser splits the user query into single terms that are used as parameters for the search step. It should be noted that the same analyzers that run initially before indexing, also run on the search query, in order to uniform the query and match parts.

- A set of ***Terms*** is used as requirements for the search mechanism. The query ***Terms*** are the basic unit for searching. They correspond to the name of the field to be searched.

- Then, ***Lucene.IndexSearcher*** returns the relevant results in the form of a ***Lucene.Hits*** object. The hits collection that the searcher returns is ordered by ranking, in other words, by how well each document matches the query.

- Display the results through ***Lucene.Hits***. This structure only contains references to the underlying documents, that is, pointers to ranked search results. Therefore, instead of documents being loaded immediately upon search, matches are loaded from the index when requested. Since users usually access only the first few relevant documents, isn't necessary to retrieve all results; only the documents that will be presented to the user.

There are several types of queries that *Lucene* supports:

- Boolean Query,
- Phrase Query,
- Prefix Query,
- Range Query,
- Filtered Query,
- Span Query.

Moreover, there are several query expressions supporter by *Lucene*. The Table 3 shows some examples that ***Lucene.QueryParser*** handles [Luc10]:

Query Expression	Matched Documents
retail	Contain the term *retail* in the content field.
retail business ***retail* or *business***	Contain the term *retail* or *business*, or both in the content field
+retail +business ***retail* and *business***	Contain both *retail* and *business* in the content field
(***retail* or *business***) ***processes***	Contain *processes* and must also contain *retail* or/and *business*, all the content field
(retail^1.3) business processes	Give more relevance to *retail* keyword more that *business* and *processes* keywords in the content field
"retail business processes"	Contain exactly the phrase "*retail business processes*" in the content field
retail*	Contain terms that begin with *retail*, such as *retailing* and *retailer* in the content field.

Table 3 - *Lucene*'s Query Examples

After query the index with a query, ***Lucene.Hits*** object is available. ***Lucene.Hits*** provides efficient access to search results. Results are ordered by relevance, as in fact, *Lucene* internally computes a score, a numeric value of relevance for each document, regarding a user query. The ***Lucene.Hits*** object caches a limited number of documents and maintains a most recently-used list. In fact, the first 100 documents are automatically retrieved and cached initially, as only the best-scoring hits are the desired documents. *Lucene* uses the Equation to calculate the document score based on a query. This formula is based on a combination of the Vector Space Model along with the Boolean model to establish how relevant a user query is to a given document. The Boolean model is used to narrow down the documents following Boolean login in the query specification. "…*The idea behind is that is the more times a query term appears in a document relative to the number of times the term appears in all the documents in the collection, the more relevant that document is to the query*" [LSA10].

$$score(q,d) = coord(q,d) * queryNorm(q) * \sum_{t\ in\ q} \text{tf(t in d)} * \text{idf(t)}^2 * \text{t.getBoost( )} * \text{norm(t, d)}$$

Equation 5- Lucene's Scoring Formula, from [LSA10]

Lucene's scoring formula correlates the cosine-distance between document and the query vectors, using a Vector Space Model. A document whose vector is closer to the query vector in that model is scored higher. The following Table 4 summarizes all the factors from the score equation:

Factor	Description
tf(t in d)	Term frequency factor for the term (t) in the document (d).
idf(t)	Inverse document frequency of the term.
getBoost()	Field boost, as set during indexing.
*lengthNorm(t*field in d)*	Normalization value of a field, given the number of terms within the field.
coord(q, d)	Coordination factor, based on the number of query terms the document contains.
norm(q)	Normalization value for a query, given the sum of the squared weights of each of the query terms.

Table 4- Lucene's Scoring Factors description

3.2.2 Extending Information Retrieval to Passage Retrieval

In order to fulfil the project demands, retrieving the best document concerning a set of information requirements is not enough. Typical Information Retrieval systems locate and retrieve documents regarding some user need. On the other hand, Passage Retrieval (PR) Systems focus on modelling and retrieving relevant portions from documents. In fact, PR is concerned on finding passage-blocks that could provide better evidences to the users, rather than the full document text. Instead of documents as answer format, text snippets present itself as a more suitable approach within Question and Answering (QA) context. Moreover, evidence shows that users usually prefer passage sized answers over whole document retrieval, as passages provide context.

In order to fulfil QA demands, the tactic used was to exploit *Lucene*'s flexibility of the representation of documents within indexes. In fact, instead of indexing all the words from one document, documents are break up by segments. So, instead of the full document text being

indexed at once as a ***Document***, the content is pre-processed and divided into discourse passages.

Three types of discourse division were implemented within the IR system, as described in the Figure 13:

- Break the text by sentences
- Break the text by paragraphs
- Set passage window with a fixed number of sentences.

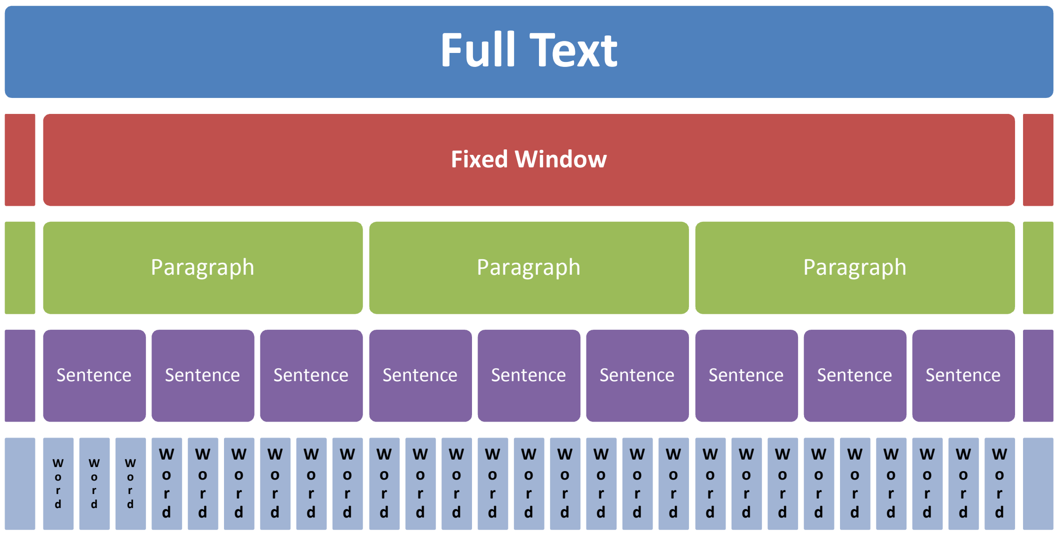

Figure 13 - Document Discourse Segmentation

Then, the passage is indexed as a document, that is, the passages virtually are independent documents. From there, every time a term is queried, despite of the system retrieve the original file, it retrieves only the passage snippet that corresponds to that query.

3.3 Genetic Algorithm for *Lucene*'s Query Optimization

The following sub-sections describe the development and integration of a Genetic Algorithm with an Information Retrieval framework, in order to improve Question and Answering performance.

3.3.1 Exploiting *Lucene*'s query features

The purpose of Information Retrieval is to fetch relevant information regarding some user's needs. Users, usually express such information's necessity by means of a query of words, that is, a full set of pertinent terms that users believe that are appropriated to express their information requisites in order to retrieve the most relevant results along with narrowing down most of the non-relevant ones. As experience shows, IR systems perform better as closer is the

relationship between the user input query and the content from the documents stored in the index.

Today's users of IR system are trained to fine-tune their keywords in order to faster acquire the desired result. User's queries are a careful combination of terms along with special operators that reduce the search space and improve retrieval performance. Figure 14 shows an example from one advanced query user using *Google* search engine. In this example, user queried the web IR tool in order to retrieve all the WebPages that contain the exact keywords "*information retrieval*" together, or the exact keywords "*search engine*" together, plus WebPages that not contain the keyword "*Wikipedia*".

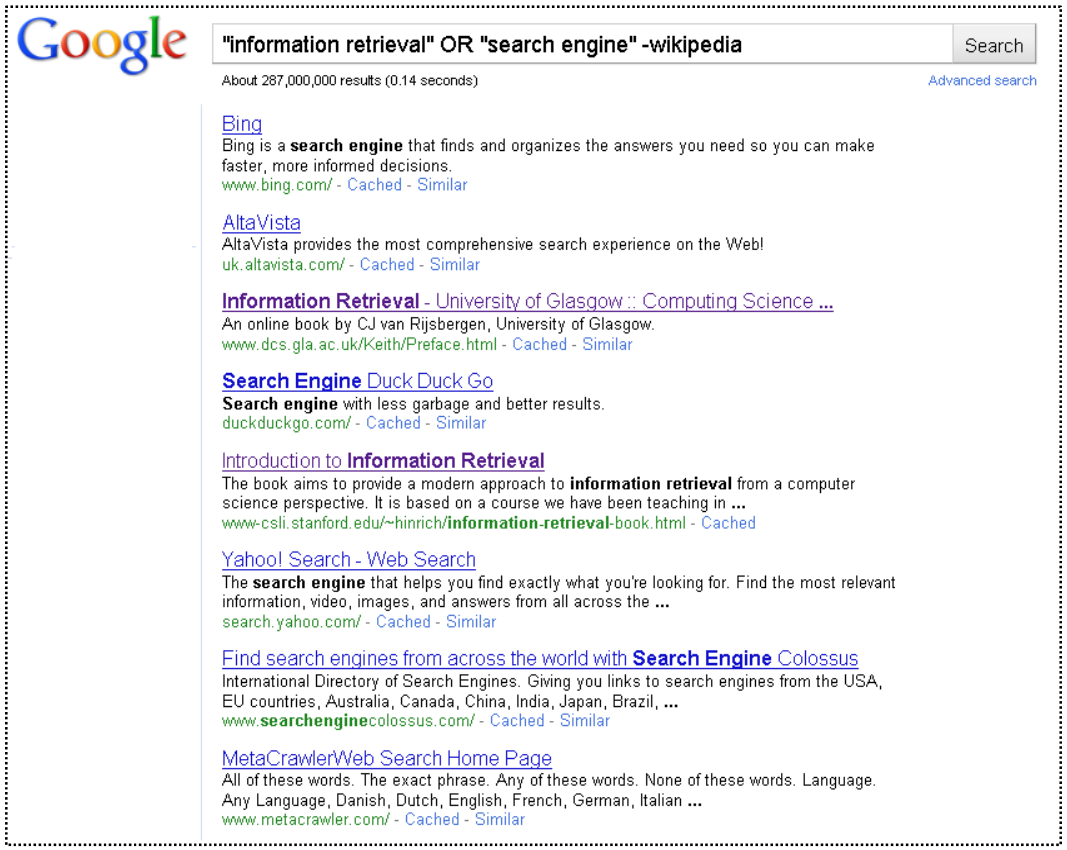

Figure 14 – Advanced query using tuned keywords with Google search engine

Moreover, another interesting paradigm within Information Retrieval is that different people use an amazingly wide variety of words to describe the same information need [Dum09]. This characteristic of human word usage set restrictions as how well a simple lexical matching system could do in satisfying users.

Hence, automatic optimization processes are needed in order to fit query parameters in such way that they produce an optimal retrieval performance. Genetic algorithms (GAs) represent an attractive Artificial Intelligence paradigm, as they provide an efficient searching methodology for the best possible plan in order to uncover good solutions to problems. Within

IR context, GAs can lend a hand to search mechanisms as they can draw optimization plans for input queries, progressively improving the retrieval of the most relevant documents. As a result, a genetic algorithm will be applied to the query terms, as they influence directly the retrieval results.

Fortuitously, *Apache Lucene* provides a rich array of query operators, such as the ones presented before on the Table 3. These operators can be integrated within a GA and consecutively be applied over the query terms in order to refine results.

Secondly, as the built IR system belongs to higher Question and Answering system, relevant linguistic clues and search heuristics could by devised from the Question Analyzer module. Linguistically analysing questions, several constrains can imposed in order to make a better fine-grained selection of keywords to generate relevant queries that will be further processed by the IR search engine.

As shown in the example described Figure 15 above, a set of keywords plus extra linguistic information is extracted from the original user question.

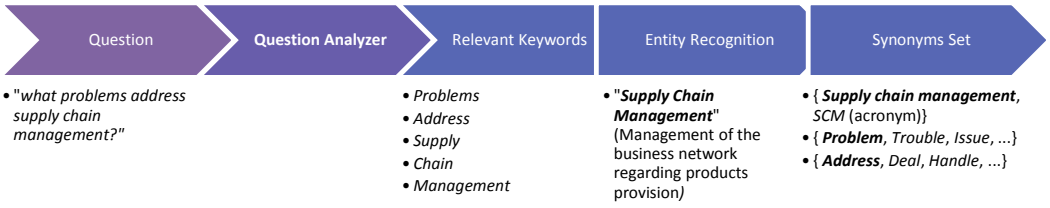

Figure 15 - Linguistic Features that can be extracted from one example question

Afterward, as the IR system is also linked to a Semantic Network, even more relevant features could be included in the query, as the network could retrieve equivalent semantic entities, such as supra-concepts.

Following the example shown on the Figure 16, *Wal-Mart* is a department store. So, when some random query concerns *Wal-Mart*, the keyword *department store* could be additionally added to the query engine. This will enhance the retrieved results, as *department store* is a super concept of *Wal-Mart*, providing further knowledge about it. If the user's information need was to know more about *Wal-Mart*, the semantic network already provided a very precise clue to the IR system, as it identifies that is a *department store*.

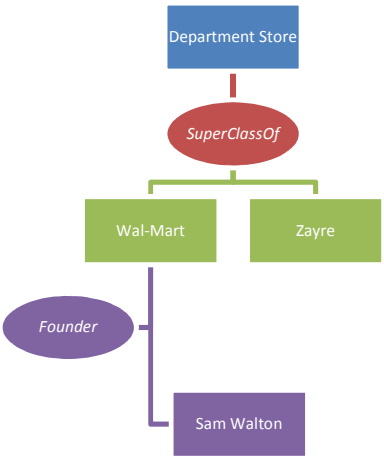

Figure 16 - Part of one example Semantic Network

As shown in the Figure 17, the ingredients for the genetic algorithm perform are the entities and the synonyms retrieved by the Question Analyzer module, the semantic related entities retrieved by the semantic network, and fundamentally user's information needs described as an array of terms where several operators can be applied.

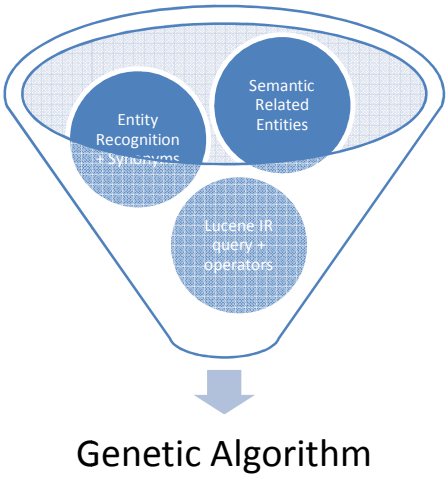

Figure 17 - Ingredients for the genetic algorithm

Applying GAs to this approach, the optimization criterion is the fitness applied over queries where the reference value is based on the *Lucene*'s internal scoring function. Then, GA takes care of the optimization process, as it will generate all the semantically equivalent plans for some given query. . This query expansion will allow providing passages which are no syntactically matched by the IR system using the original queries.

Figure 18 summarizes how the ingredients that will power the GA will be integrated within the *Lucene* IR framework.

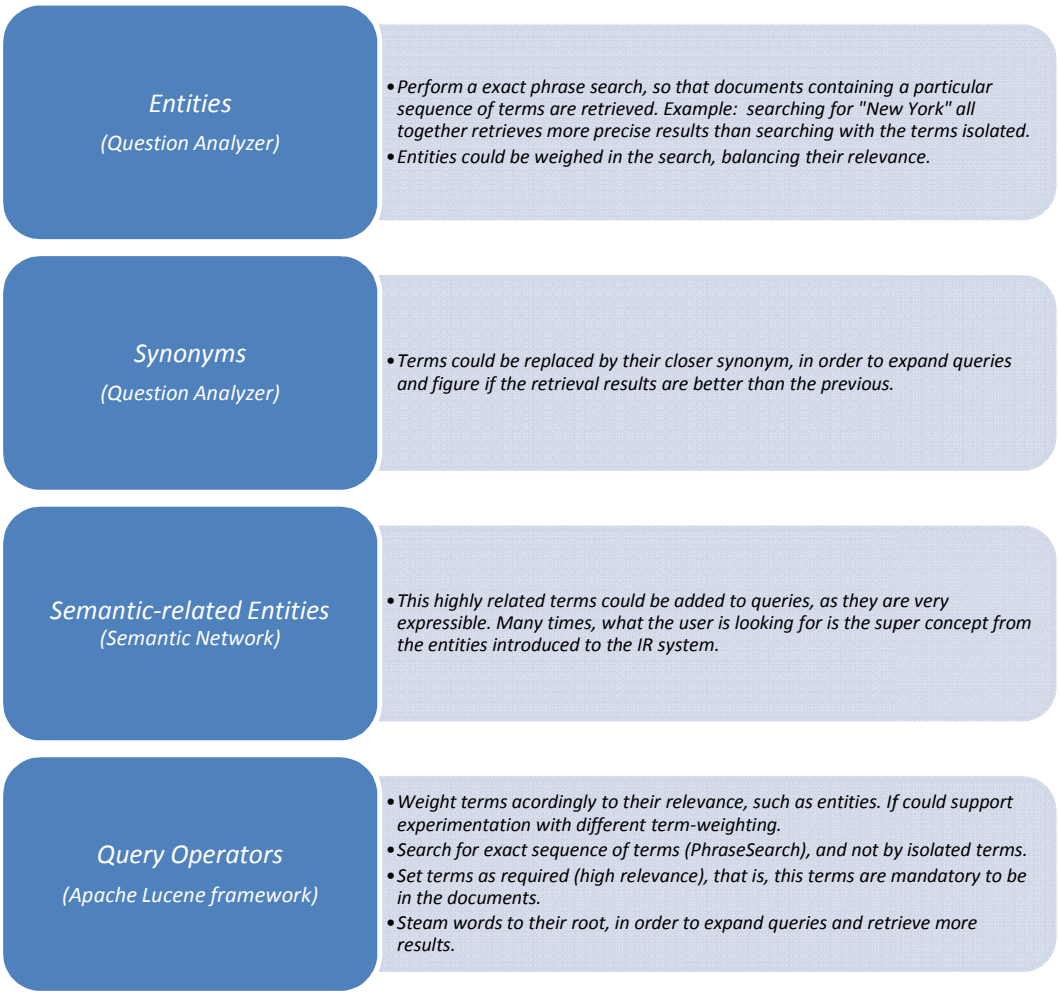

Figure 18 - How to use all the linguistic features with Lucene

As it's well known by the research community, query expansion improves the recall. However there are shortcomings such as decreasing the precision of the system. In fact, reformulating the query with synonyms, other weights and related terms have to be well considered.

3.3.2 Genetic Algorithm Implementation

Genetic Algorithms are robust in searching a multidimensional space to find optimal or near optimal solutions [Hol75]. This encouraged the use of GA in this project to search for such an optimal or near optimal combination of keywords, weights and operators.

This section is intended to make an analogy between the genetic algorithms mechanism and the developed implementation, within Information Retrieval context. Figure 19 shows the algorithm implemented:

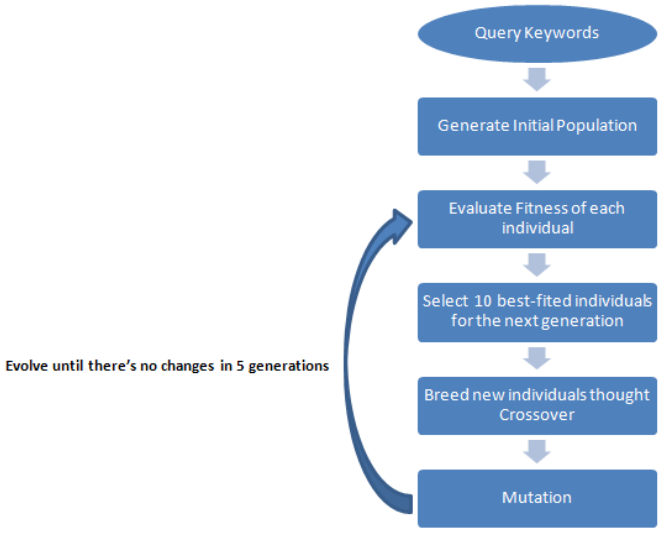

Figure 19 - Genetic Algorithm implementation

Representation of Individuals

Settings are directly encoded into individuals, without converting them into binary representation.

Thus, each *Lucene*'s query q is considered an individual. Each query q is constituted by a set of terms $t_1, t_2, t_3, ..., t_n$ plus a set of query operators $op_1, op_2, op_3, ..., op_n$ (*Lucene*'s query operators: required marker, weighting factors, phrase search marker, prefix marker) that can be applied to the terms, resulting $q = op_1(t_1), op_1(t_2), op_1(t_3), ..., op_n(t_n)$. It should be noted that not all parameters have to be present, and one operator is one or more *Lucene*'s query operators applied.

An example from one individual of length 5 is the following one represented on the Figure 20 bellow:

- *problems address supply chain management*

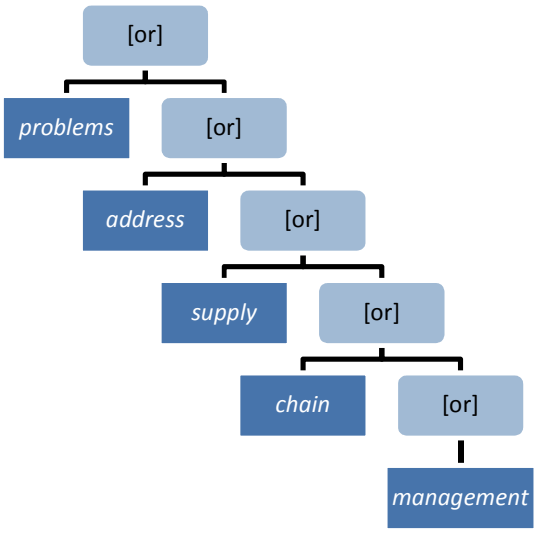

Figure 20 - Example of an individual

As we previously see, some special operators can be applied to the terms in order to bend the retrieval results. The next example described on the Figure 21 bellow, shows an example from one individual of length 3, with several operators applied over the terms.

- *(problems^1.9) +address "supply chain management"*

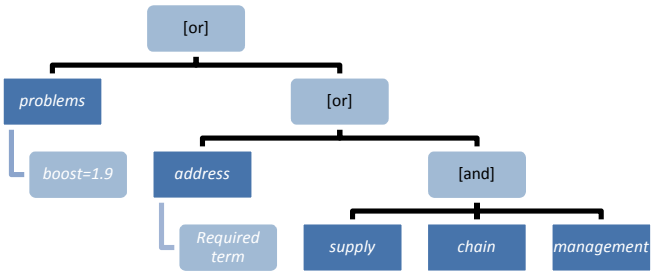

Figure 21 - Example of an individual with operations applied

Fitness Function

In order to measure the adaptability from one individual, that is, the relevance of a query given a set of information requirements and a collection of documents, it is computed the

arithmetic mean of *Lucene*'s score from the 10 best-ranked passages returned by the IR engine for some particular query in a given corpus, as described on the Figure 22 bellow: mean

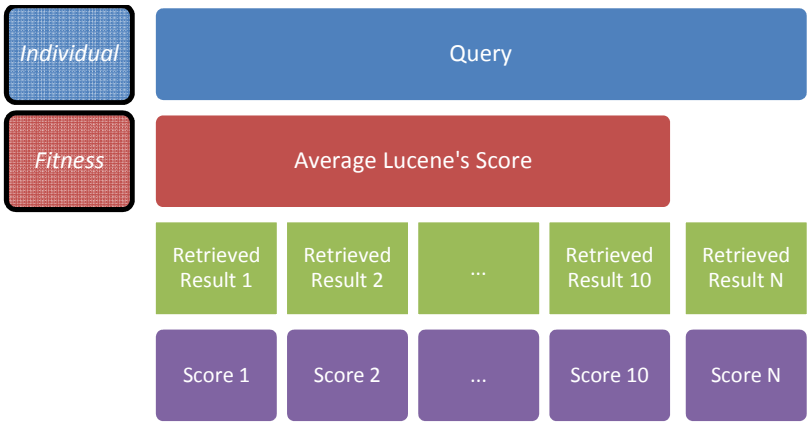

Figure 22 - Fitness from queries

Population Size

The population size is composed by 100 individuals, that is, 100 *Lucene*'s queries.

Population Initialization

The initial population is made following the following considerations:

- The original query is the first individual;
- If name-entities are recognised in the original query, the phrase-query operator is applied to all that terms;
- If the semantic network infers related entities, new individuals will be generated contemplating these new knowledge;
- All the combinations from the original query with synonyms are computed. That includes acronyms combinations also;
- Relations extracted from semantic network are also introduced;
- The rest of the population is created from the scratch by mutation operators, until make up the amount of 100 individuals;

One example from the first 10 individuals from one initial population is:

1. *problems address "supply chain management"*
2. *problems address SCM*
3. *issues address "supply chain management"*
4. *issues address SCM*
5. *(problems+) address "supply chain management"*
6. *problems (address^1.3) SCM*
7. *address SCRM*

8. *problems address "supply chain management" "product management"*
9. problems *"supply chain management"*
10. *problems address(SCM*^0.14)
11. *... (until 100)*

The formation of the initial population has a strong impact on the genetic approach.

Mutation

Mutations are used to maintain genetic diversity, preventing the population to become too similar to each other, thus slowing or even stopping evolution.

Mutation, as arbitrary perturbation in the chromosome representation, is required to guarantee that the in progress generations are still linked to the complete search space. Moreover, it's indispensable to bring in new genetic material into a population that has stabilized stage [Kra99].

Several mutations with fixed probabilities have been defined, as described on the Table 5:

Mutation	Probability	Description
Remove term	1%	Removes one random term from the individual;
Set term as required	1%	Puts the required marker into one random term from the individual; So, this term is mandatory.
Weighting term	2%	Change the boost factor from one random term from the individual;
Term stemming	3%	Stem one random term from the individual, so that terms with the same root could be retrieved.

Table 5 - Mutation operations

Crossover

Random selection supports genetic algorithm to avoid *local optima* by preventing the population of chromosomes from *crowding*, that is, certain individuals because of their superior fitness dominate the population, causing the algorithm to get trapped in a local maxima earlier.

- Select the best 10% of individuals from the current population to survive, that is, it goes straight to the next generation.
- The remaining part of the population will be replaced by new offspring created using crossover. Elements are selected from the remaining 90% of the population to be parents of new offspring to fill the next generation. The pairs of individuals

that are selected to crossover (the parents) are picked randomly, and from there, they generate two new individuals.

The Crossover works as it follows:

- One offspring is created by merging parameters from two parents.
- Each pair of parents generates two new chromosomes, one combining the average features from terms between the parents, and the other one, combining the maximum features from terms between the parents.
    - It should be noted that when combining terms with required mark, the required markers always overwrite the other features.

The Table 6 shows examples of how the crossover between terms is done:

	Example 1	***Example 2***	***Example 3***	***Example 4***	***Example 5***
Term 1 from Individual 1	*elegant*^1.4	*+elegant*	*elegant^1.6*	*elegant^0.3*	*"information retrieval"*
Term 2 from Individual 2	*elegant*^0.2	Ø (=*elegant*^0.0)	Ø (=*elegant*^0.0)	*delightful*^0.5	*IR*
Average Crossover	*elegant*^0.8	*elegant*^0.5	*elegant^0.8*	*delightful*^0.4	*"information retrieval"*
Maximum Crossover	*elegant*^1.4	*+elegant*	elegant^1.6	*delightful*^0.5	*"information retrieval"*

Table 6 - Examples from crossover with terms

The following *Table 7* shows the complete crossover process between two individuals:

1. *(primary^0.8) +purpose (address^0.6) "supply chain management"*
2. *Primary (purpose^0.1) scm*

	Keyword 1	***Keyword 2***	***Keyword 3***	***Keyword 4***
Individual 1	*primary*^0.8	*+purpose*	*address^0.6*	*"supply chain management"*
Individual 2	*primary* (=*primary* ^1.0)	purpose^0.1	Ø (=*address* ^0.0)	*scm*
New Individual 1 by Average Crossover	*primary* ^0.9	*+purpose*	*address ^0.3*	*"supply chain management"*
New Individual 2 by Maximum Crossover	*primary*	*+purpose*	*address* ^0.6	*"supply chain management"*

Table 7 - Crossover between two individuals

Then, mutation operations are applied to each new offspring, with fixed probabilities. Finally, the new population is ready to start a next generation.

Termination

The genetic algorithm repeats the process all over again, until the last 5 generations don't shown any improvements, that is, from there the overall fitness values are the same.

3.4 Summary

Prymas is a full-size Question and Answering system, extended with several layers. As we could see, Knowledge Retrieval layer is an important module from the *Prymas* Architecture. This level is in charge of storing all the text information, plus providing efficient mechanisms of knowledge recovery from the database of facts.

Apache Lucene is an Information Retrieval framework that allows building an application with indexing and searching capabilities. *Lucene* is one of the most popular IR libraries along the industrial context.

The high customization level that *Lucene* supports, suits the project goals better as precise custom-made strategies can be pursuit in order to better fulfil the business demands.

As a result of such flexibility, a standard *Lucene* IR system was extended in order to incorporate passage retrieval mechanisms. It is known that users prefer passage responses, instead of documents as it gives context.

Finally, a genetic algorithm was implemented within the query mechanism within the Passage retrieval system, in order to improve the optimization the user's answers regarding some question. This Query expansion will allow providing passages which are no syntactically matched by the IR system using the original queries.

In short, the design from the extended Information Retrieval system contemplates the following modules and the interactions shown by the Figure 23.

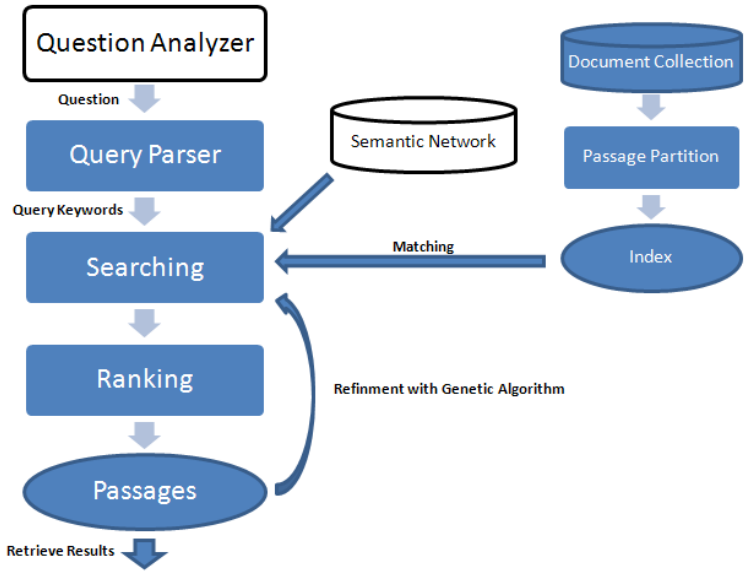

Figure 23 – Extended Information Retrieval implementation

Chapter 4

Experiments and Results

A genetic program was built over a testing environment to see the influence of Genetic Algorithms on Question and Answering (QA) context.

Thus, a standard Information Retrieval (IR) system built with Apache *Lucene* framework was compared to an equivalent one. Both of the two systems were evolved to Passage Retrieval (PR) systems to fulfil QA requirements, however, this last system was extended with a Genetic Algorithm (GA) to enhance retrieval task.

For the two versions of IR systems on trial, three models of PR were compared: sentence-based, paragraph-based, and a fixed-window-of-sentences-based.

4.1. Preliminary considerations

For the Information Retrieval system proposed, an input data set was used based on several document collections, such as functional reports regarding retail software projects, WebPages extracted from Wikipedia about basic retail concepts, retail handbooks and technical manuals about Oracle Retail Merchandising System[24]. The number of words, sentences and paragraphs from each document is variable. Using the paragraph passage retrieval indexer as a reference, the total amount of documents taken as input was 7575.

Secondly, a script of 60 questions was developed along with a retail expert. This compilation of questions is considered usual within retail world, especially for those who are inducted in the area. Besides, a set of results judgements for the questions was also shaped, that is, an assortment of suggested acceptable answers or guidelines regarding some user's

[24] ***Oracle Retail Merchandising System****: End-to-end integrated framework that supports retail industry.*

information need (See *Appendix A* Test Questions and Judgements).

Afterwards, the answers retrieved by the IR system were binary classified as relevant or non-relevant accordingly to the previous judgements. A comparison is made to the top 3 most relevant answers retrieved by each system. The system can answer one question if at least one passage retrieved from the three is evaluated as relevant.

Next, the standard Apache *Lucene* IR system extended to perform as a Passage Retrieval system is compared against the other analogous system that uses a Genetic Algorithm beneath. Both the systems were supplied with the same input keywords from the Question Analyzer module, a module from the higher Question and Answering system. Recall that *Lucene* framework performs ranked retrieval, that is, returns documents ordered by how close their meaning is to the input query. Three types of snippets as response were experimented: sentence-passage, paragraph-passage and a fixed window of 5-sentences-passage.

There's no fixed number of generations to be breed by the Genetic Algorithm. However, algorithm stops if the last 5 generated populations are the same.

4.2. Sentence Passage Retrieval Comparison

In this case study, a standard Apache *Lucene* Information Retrieval system, extended with Sentence Passage Retrieval mechanism was compared against the congener version with a Genetic Algorithm embedded in the search system.

The Figure 24 shows a chart comparing the rate of relevant answers retrieved, that is, the precision from first-to-third passages retrieved by each system.

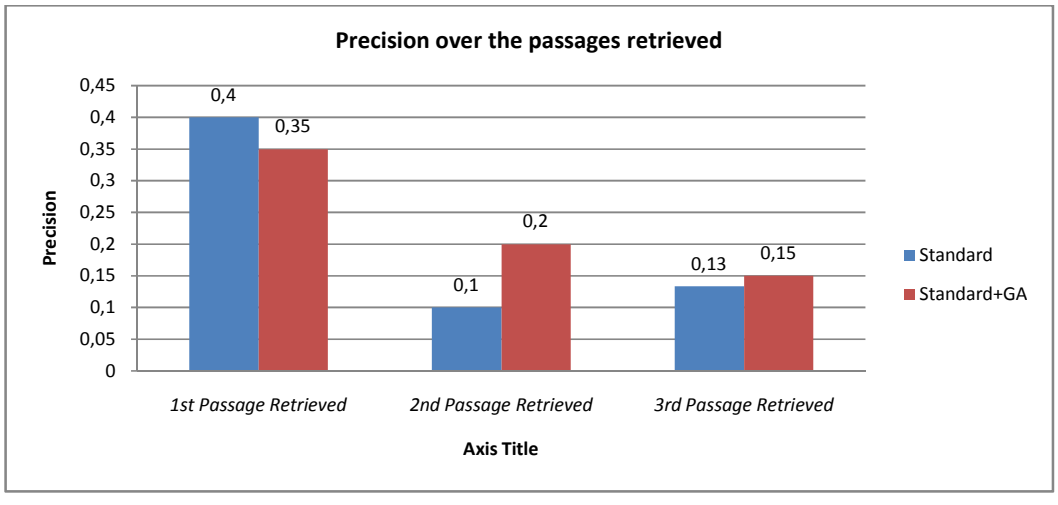

Figure 24 - Precision over the passages using sentence retrieval

Next, the Table 8 shows the number of relevant answers retrieved from first-to-third passages retrieved by each system.

	1st Retrieved Passage (Nº relevant answers)	2nd Retrieved Passage (Nº relevant answers)	3rd Retrieved Passage (Nº relevant answers)
Standard	24	6	8
Standard+GA	21	12	9
(2-1)	***(-3)***	***(+6)***	***(+1)***

Table 8 – Number of relevant answers using sentence retrieval

The average precision for each question using the sentence passage retrieval is shown on *Appendix H* Average Precision with Sentence Retrieval.

Afterwards, Table 9 presents a summary of the proportion of correct answers that each system can perform, along with their mean average precision.

	% Correct Answers	% Mean Average Precision
1. Standard	58,3333	47,36
2. Standard+GA	61,6667	46,80
(2-1)	***(+3,3334)***	***(-0,56)***

Table 9 – Fraction of correct answers and MAP using sentence retrieval

In short, despite of the Standard IR system with GA has less mean average precision than the standard version, the percentage of correct answers that it performs is bigger.

4.3. Paragraph Passage Retrieval Comparison

In this case study, a standard Apache *Lucene* Information Retrieval system, extended with Paragraph Passage Retrieval mechanism was compared against the congener version of the system with a Genetic Algorithm embedded in the search system.

The Figure 25 shows a chart comparing the rate of relevant answers retrieved, that is, the precision from first-to-third passages retrieved by each system.

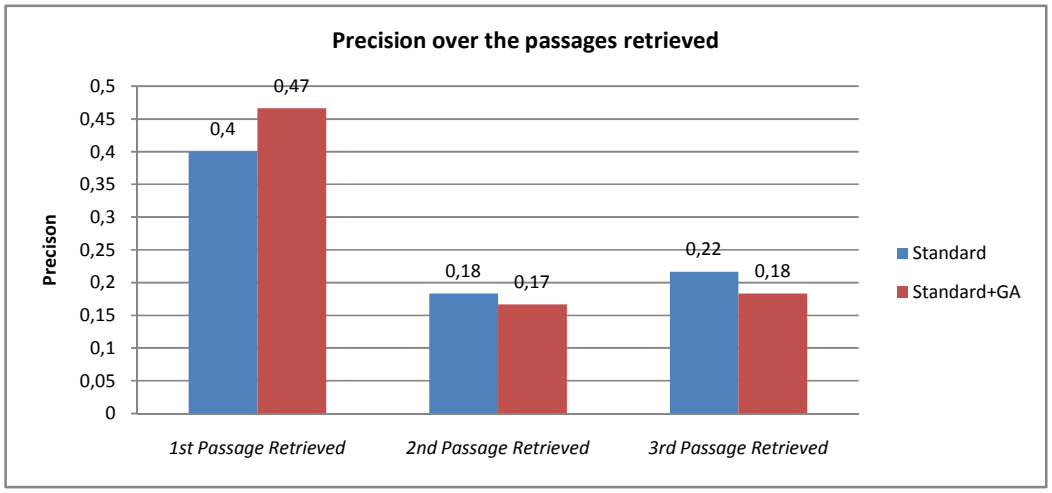

Figure 25 - Precision over the passages using paragraph retrieval

Next, the Table 10 shows the number of relevant answers retrieved from first-to-third passages retrieved by each system.

	1st Retrieved Passage (Nº correct answers)	2nd Retrieved Passage (Nº correct answers)	3rd Retrieved Passage (Nº correct answers)
1. Standard	24	11	13
2. Standard+GA	28	10	11
(2-1)	***(+4)***	***(-1)***	***(-2)***

Table 10 - Number of relevant answers using paragraph retrieval

The average precision for each question using paragraph passage retrieval is shown on *Appendix I* Average Precision with Paragraph Retrieval.

Afterwards, Table 11 presents a summary of the proportion of correct answers that each system can perform, along with their mean average precision.

	% Correct Answers	% Mean Average Precision
1. Standard	66,6667	50,69
2. Standard+GA	71,6667	56,38
(2-1)	***(+5,0000)***	***(+5,69)***

Table 11 - Fraction of correct answers and MAP using paragraph retrieval

In short, using paragraph segmentation, the IR version with GA surpasses the standard one, in terms of percentage of correct answers as well as in mean average precision.

4.4. Fixed-Window Passage Retrieval Comparison

In this case study, a standard Apache *Lucene* Information Retrieval system, extended with a fixed-5-sentences-window Passage Retrieval mechanism was compared against the other version of same standard Passage Retrieval system, but with a Genetic Algorithm embedded in the search system.

The Figure 26 shows a chart comparing the rate of relevant answers retrieved, that is, the precision from first-to-third passages retrieved by each system.

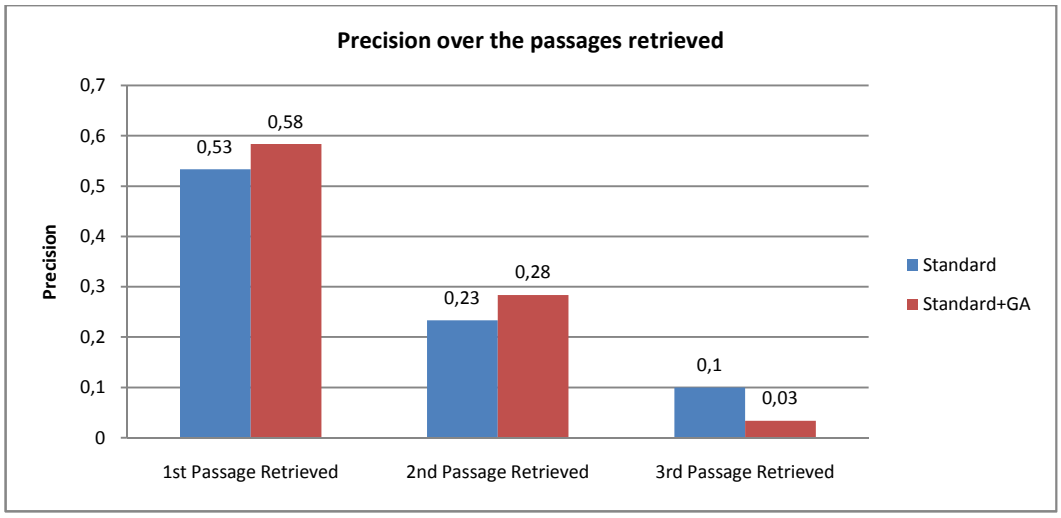

Figure 26 - Precision over the passages using sentence retrieval

Next, the Table 12 shows the number of relevant answers retrieved from first-to-third passages retrieved by each system.

	1st Retrieved Passage (# correct answers)	2nd Retrieved Passage (# correct answers)	3rd Retrieved Passage (# correct answers)
1. **Standard**	32	14	6
2. Standard+GA	35	17	2
(2-1)	***(3)***	***(3)***	***(-4)***

Table 12 - Number of relevant answers using fixed-window retrieval

The average precision for each question using the fixed-window passage retrieval is shown on *Appendix J* Average Precision with Fixed-Window Retrieval.

Afterwards, Table 13 presents a summary of the proportion of correct answers that each system can perform, along with their mean average precision.

	% Correct Answers	% Mean Average Precision
1. Standard	70,0000	60,42
2. Standard+GA	76,6667	67,36
(2-1)	***(+6,6667)***	***(+6,94)***

Table 13 - Fraction of correct answers and MAP using fixed-window retrieval

In short, using fixed-window segmentation, the IR version with GA outperforms the standard one, in terms of percentage of correct answers as well as in mean average precision.

4.5. Passage Retrieval Evaluation

The following Table 14 summarizes all the experimentations done with the several segmentation types, using the two IR systems: the standard one and another extended with GA. This summary shows the evolution of the number of relevant answers retrieved.

	Segmentation	Nº Correct Answers	Nº relevant answers (1st Retrieved)	Nº relevant answers (2nd Retrieved)	Nº relevant answers (3rd Retrieved)
1. Standard	**Sentences**	35	24	6	8
	Paragraphs	40	24	11	13
	Fixed-5-Window-Size	42	*32*	*14*	*6*
2. Standard+GA	**Sentences**	37	21	12	9
	Paragraphs	43	28	10	11
	Fixed-5-Window-Size	46	*35*	*17*	*2*

Table 14- Comparison between segmentation methods over passage retrieval

The following Figure 27 shows another overview over the evolution of the precision over the several models, considering the top-3 passages and the first passage retrieved.

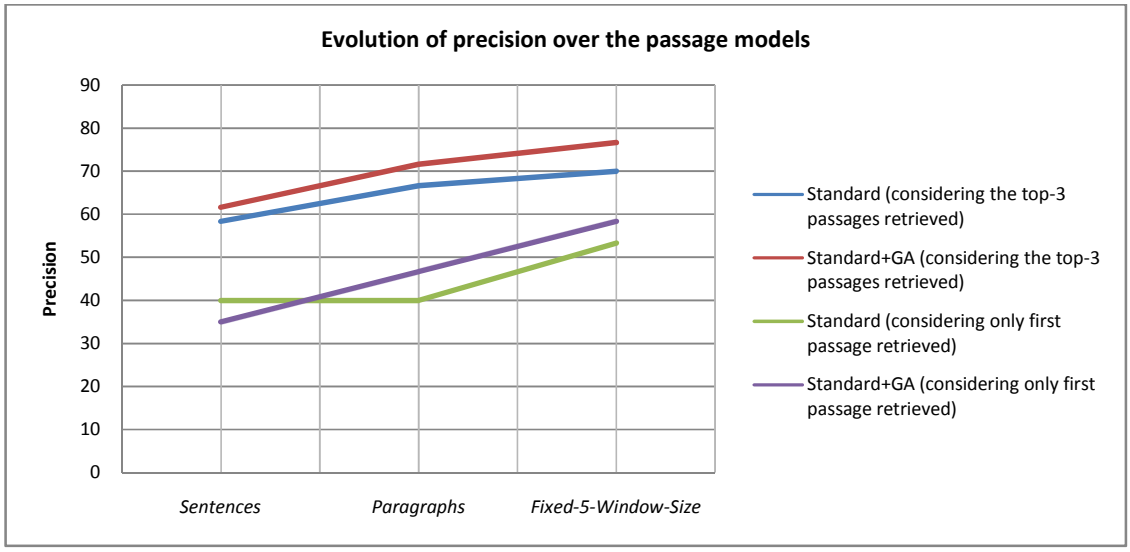

Figure 27- Comparison between precision on passage models

As supported by the graphic presented on *Appendix G Comparison between Passage Retrieval methods,* the genetic algorithm progressively improves the relevance of the several passage retrieval models. In fact, the top ten queries are increasingly enhanced over the generations, in order to be more relevant to the question posed.

In short we can see that generally, the bigger the passage size, the more precisely the IR systems will perform.

4.6. Discussion and Conclusions

In this section were evaluated three strategies of Passage Retrieval sentence-based, paragraph-based and fixed-window-based methods of defining passages. The experiments were conducted with two Apache *Lucene* Information Retrieval systems, one standard version, and another one extended with a Genetic Algorithm.

The experiments were conducted to measure the efficiency from the Genetic Algorithm applied to the passage retrieval mechanism, more precisely to evaluate if the query refinement done by the algorithm increases the precision. Moreover, the experimentations were indented to support understanding how passages should be defined and how they should be used in retrieval.

Hence, some general conclusions can be drawn from the results done here:

- IR plus Genetic Algorithm version of the system performs better that the standard version, as it takes more percentage of correct answers, using all the three models

of PR. This is thanks to its better average precision performance over the standard version.

- The first passage retrieved using IR plus Genetic algorithm version proves to be relevant more often, compared to the standard version. Retrieving relevant results in the beginning is very important, as users don't want to scroll down too much, in order to find the desired question.

- With the exception of sentence retrieval, IR plus GA version always has more mean average precision compared to the standard version. Mean average precision is a standard measure among TREC community and gives a good discrimination measure of quality across recall levels.

- The bigger is the passage size, the more relevant answers the IR systems obtain. Moreover, the number of first-retrieved and relevant answers also increases with bigger passage sizes.

In summary, the experiments shows that proper weighting, correct selection of keywords and linguistic features from the GA is significant to improve retrieval performance from question and answering systems, as the mean average precision is higher when comparing to the standard version.

On the other hand, as the size of the passage retrieved increases, systems perform higher mean-average-precision rates. In fact, bigger passages provide more context to users, better fulfilling their needs. Besides, it may be the case that there is no single point of division of text into passages that satisfies all queries. In short, there has to be good sense whenever choosing the appropriate passage segmentation. Thus, it is important to balance passage size along with user's requirements, having in mind that users don't want to browse large amounts of irrelevant information in order to accomplish their answers requests.

4.7. Summary

In this section were evaluated three strategies of Passage Retrieval sentence-based, paragraph-based and fixed-window-based methods of defining passages. The experiments were conducted with two Apache *Lucene* Information Retrieval systems, one standard version, and another one extended with a Genetic Algorithm.

The purpose of this study was to measure the efficiency from the Genetic Algorithm applied to the passage retrieval mechanism, and help to understanding how passages should be used in retrieval.

In short, the experiments show that appropriate weighting and selection of keywords and linguistic features by the GA are important to improve retrieval from QA systems. Moreover, even though enlarging the passage size increases the precision of the systems, segmentation should be done carefully.

Chapter 5

Conclusions and Future Work

This chapter identifies the main contributions for the research scheme, describing an outline from the developed work and the main goals accomplished, along with the foremost conclusions from the experiments done. Moreover, some guidelines for future work are also revealed.

5.1 Retrospective

Knowledge management is a breakthrough for organizations. Current corporations expect to increase their competitive advantages from the efficient and effective management of their knowledge assets. Without ready and easy access to knowledge, every issue is taken based on what the individual brings to the circumstance, instead of being the total sum of knowledge that everyone within the organization has ever learned about a problem of a similar nature.

Usually, non-technical persons don't want to rely on a particular technology to perform search tasks. Question and Answering systems provide an interesting interface to the base of facts, as users can express their needs into the system using their own natural language.

Information Retrieval is crucial for efficient access to large sorts of unstructured information. They lend a hand with the question of how the information should be organized so that queries can be resolved and relevant portions of information can be located and extracted.

With the overwhelming amount of information available, casual browsing by human means or even comprehensive searching by automatic programs is very costly. In short,

indexing mechanisms and proper methods to match user's information needs along with the sorted information are mandatory in this rapidly changing world.

Within this project an approach to tackle information overload issue was proposed. It was proposed an improved Information Retrieval (IR) system with Passage Retrieval that could provide short answers to queries expressed in natural language. The IR system is supported by Natural Language Processing techniques that analyze the collection of text documents as its main source of knowledge, and also by meaningful linguistic features extracted from the user's query.

In addition to that, in order to evolve traditional IR systems to a modern one who could better fulfil the user's information requisites, an evolutionary approach based on Genetic Algorithms was integrated in order to exploit query expansion. Hence, the algorithm combines several query keywords and experiments with various operators to progressively refine queries so that the retrieved results could be optimized along with the user's requests. Finally, this process is also enriched by linguistic features extracted by the question analyzer module and by relevant entities provided by semantic networks.

Within this project, some experiments were done in order to measure the efficiency from the Genetic Algorithm (GA) applied to the passage retrieval mechanism, more precisely evaluate if the query refinement done by the algorithm increases the precision. The experimentations were also intended to support better understanding about how passages should be defined and how they should be used in retrieval.

It was seen that as the passage size increased, the average precision of the system also increases. In fact, bigger passages provide more context to users, better fulfilling their needs. Besides, it may be the case that there is no single point of division of text into passages that satisfies all queries.

Furthermore, the trials also show that proper weighting; correct selection of keywords and linguistic features from the GA is very significant to improve retrieval performance from question and answering systems. In fact, an IR system extended with GA performed better mean average precision when comparing to the standard version, using sentence retrieval, paragraph retrieval and a fixed window of five sentences retrieval. I see the need to pursue more research in this promising area.

5.2 Goals Satisfaction

"I was really impressed when I first heard about this [Prymas] the concept…
We are doing a little bit ahead, comparing to what our competitors think they are doing...This represents the role that the innovation should have within organizations… Prymas is a different way to look at information…It's a very good concept. We want to work in that; we want to invest in that…"

(*Wipro Retail Manager*)

The area of research from this project is very large and being part of an innovation project is very motivating. Moreover, team work promotes the exchange and generation of new ideas. To the knowledge sharing problem posed by the company, the *Prymas* team was able to respond with success to the challenge, presenting a suitable solution. The receptivity from the proposal all over the company was really positive, reaching even top management who were truly surprised by this initiative. Presently, a market research is being carried out in order to adapt the architecture from the system to commercial purposes.

Moreover, following the spirit of open-innovation, the team has created a blog[25] with the purpose of staying abreast of the state-of-art technologies and obtain relevant feedback by experts in the area. The acceptance of this blog by the community was great, where we had the chance to get relevant insights from them. As measure of reference, over four months of the project, the blog has more than 160 posts and got more than 8,500 visits, it was often the most read blog weekly on Wordpress, and it was on top of the fastest growing blogs in April.

Besides, as this particular proposal addresses one important concern from today's world, I am more than pleased with my proposal. In fact Information Overload is a nightmare for any company, and contributing to the research field with a novel suggestion from an improved mechanism for matching information demands with relevant information is something that is worth running at. I had no previous knowledge on Information Retrieval, however I had the chance to understand its role in the current information economy and it's a field that I wish to work again in the future.

The inclusion in *Wipro Retail* environment was very satisfactory, even exceeding expectations. The working conditions offered, as well as the support by the company boosted the successful implementation of this project. Besides, having the chance to exchange ideas with people from different technological background and from other countries is really beneficial.

5.3 Future work

It is expected that this project in the future could lend a hand to training processes and consulting projects, providing untrained users with speedy access to knowledge that in many cases would take an expert some time to find. We anticipate that this project could also be applied outside knowledge management scope, like e-business for instance.

Following current trends, Information Retrieval will be one of the most important topics in the next decade, as it is growing beyond just searching for information. Nowadays IR provides important capabilities for Knowledge Management such as information sharing, as they start to take advantage from Web-based front ends. It is vital that IR tools can support collaborative work, so that users can contribute with knowledge to the corporate facts base. Furthermore, systems are no longer restricted to one single geographical area. Thus, network infrastructures

[25] *http://whatisprymas.wordpress.com/*

must be exploited, providing access to distributed databases and databases outside the scope (e.g. web repositories of Knowledge).

Another pertinent point to consider is that IR Systems can learn with the inputs from the users. If the retrieved results have a mechanism to classify the answers as relevant or not relevant, ranking functions could be adapted in order to better fulfil user's information demands. It was already shown that machine learning techniques could lend a hand in this task [UAG04].

Moreover, query refinement could also take advantage from a set of frequently asked questions (FAQ) table, removing ambiguities and driving the users to previous processed questions. Therefore, the system skips the process of looking up for relevant information in the database, as it has already a good answer considered by the users in the FAQ table.

Likewise, capturing the user profile, that is, information about the questioner such as context and domain of interest established from different dialogues between the user and the system could provide valuable information to narrow down search results whenever ambiguities have to be resolved.

Later, as the Questing and Answering system first analyses the question before running the IR system, question type can be known *à-priori*, so an expected type of answer can be modelled. However this feature is not exploited properly, as the indexed text is not annotated.

Better text segmentation into proper snippets should also be considered, combining document-level evidence and passage-level evidences. Moreover, overlapping text windows of varying sizes should also be thought-out.

Better query expansion is also a relevant feature to consider. Moldovan and Passça showed that their system fails to answer many questions solely because of the terminological gap [Mol02]. This increases the relevance of integrating similarity sets, relevance feedback, and thesaurus structures.

Studding term weighting approaches in automated retrieval is also worth to be investigated in the future, as the weighting algorithm used in this project is ad-hoc. Salton and Buckley already showed the important of good weighting [SB98].

The digital world is creating new challenge that currently all people have to deal with. Information Retrieval tools evolve to keep up with the rising demands. Exploiting media retrieval (e.g. music, images, and videos) would be also an interesting research topic to investigate in the future.

References

[**Acc07**] Accenture (2007) "Managers Say the Majority of Information Obtained for Their Work Is Useless." (http://www.metrics2.com/blog/2007/01/04/managers_say_the_majority_of_information_obtained.html).

[**Ala98**] Alander, Jarmo T. (1998) "An indexed bibliography of genetic algorithms in signal and image." *Technical Report 94-1-SIGNAL.*

[**Alf10**] Alfresco (2010) "Alfresco ECM." Alfresco. Retrieved June 8, 2010 (http://www.alfresco.com/).

[**All97**] Allee, Verna (1997) "Twelve principles of knowledge management." *Training & Development* Volume 51(Issue 11).

[**And04**] Andriessen, Erik (2004) *How to manage experience sharing: from organisational surprises to organisational knowledge*. Emerald Group.

[**Ask10**] AskJeeves (2010) "Ask Jeeves." Retrieved June 08, 2010 (http://uk.ask.com/).

[**Aut07**] Autonomy Agentware i3 (2007) "Autonomy Agentware i3." Agentware i3. Retrieved May 4, 2010 (http://www.autonomy.com/content/News/Releases/1997/0805.en.html).

[**BR99**] Baeza-Yates, Ricardo and Berthier Ribeiro-Neto (1999) *Modern Information Retrieval*. Addison Wesley Longman Publishing Co. Inc.

[**BM85**] Blair, D. and M. Maron (1985) "An evaluation of retrieval effectiveness for a full text document-retrieval system." *Communications of the ACM*.

[**BLI10**] BLIASoft Knowledge Discovery (2010) "BLIASoft Knowledge Discovery homepage." Retrieved March 3, 2010 (http://www.bliasoft.com/Eindex.html).

[**Bol07**] Bolebruch, Justin (2007) "Knowledge Management Using XML and Blogs - An Alternative Approach to Multimedia Tagging." Boston.

[**Box10**] Box (2010) "Box - Simple Online Collaboration." Box. Retrieved June 8, 2010 (http://www.box.net/).

[**BCM04**] Buitelaar, P., P. Cimiano, and B. Magnini (2004) "Ontology Learning from Text : An Overview." *Learning*.

[**BC02**] Burger, John, C. C. e. a. (2002) “Issues, Tasks and Program Structures to Roadmap Research in Question Answering (QA).”.

[**CD99**] Cagnoni, Stefano, A. B. Dobrzeniecki, and et al. (1999) “Genetic algorithm-based interactive segmentation of 3D medical images.” *Image and Vision computing* Vol 17(Issue 12).

[**Cal94**] Callan, James P. (1994) “Passage-Level Evidence in Document Retrieval.” *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[**CQL07**] Cao, Zhe, Tao Qin, and Tie-Yan Liu (2007) “Learning to rank: from pairwise approach to listwise approach.” *Proceedings of ICML*.

[**Che95**] Chen, H (1995) “Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms.” *Journal of the American Society for Information Science*.

[**Cla97**] Clarke, C., G. Cormack, and et al. (1997) “Relevance Ranking for One to Three Term Queries.” *Proceedings of RIAO-97, 5th International Conference "Recherche d'Information Assistee par Ordinateur"*.

[**CS89**] Cleveland, Gary and Stephen Smith (1989) “Using genetic algorithms to schedule flow shop releases.” *Proceedings of the third International Conference on Genetic*.

[**Com10**] comScore (2010) “comScore Reports Global Search Market Growth of 46 Percent in 2009.” Retrieved June 1, 2010 (http://www.comscore.com/Press_Events/Press_Releases/2010/1/Global_Search_Market_Grows_46_Percent_in_2009).

[**CB04**] Cox, Roger and Paul Brittain (2004) *Retailing: an Introduction*. 5th ed. Pearson Education.

[**CS05**] Cui, H., R. Sun, and et al. (2005) “Question answering passage retrieval using dependency relations.” *SIGIR 05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[**Cui05**] Cui, Hang, Renxu Sun, and et al. (2005) “Question Answering Passage Retrieval Using Dependency Relations.” *SIGIR’05*.

[**Dal09**] Dalton, Jeff (2009) “Open Source Search Engines, Retrieval Tools and Libraries.” Information Retrieval research and search engine development. Retrieved March 1, 2010 (http://www.searchenginecaffe.com/2007/03/open-source-search-engines-in-java-and.html).

[**Dee88**] Deerwester, S., e. a. . (1988) “Improving Information Retrieval with Latent Semantic Indexing.” *Proceedings of the 51st Annual Meeting of the American Society for Information Science* Volume 25.

[**Dum09**] Dumais, Susan (2009) “An Interdisciplinary Perspective on Information Retrieval.” *Salton Award Lecture*.

[**EWA10**] EWASystemsEnterpriseAnalytics (2010) “Enterprise Analytics.” EWA Systems. Retrieved March 10, 2010 (http://www.ewasystems.com/).

[**Fra06**] Frappaolo, Carl (2006) in *Knowledge Management*. 2nd ed. Capstone Publishing, Ltd.

[**Fra10**] Frappaolo, Carl (2010) “What Enterprise 2.0 Practitioners Should Know About KM Deployments.” Retrieved May 8, 2010 (http://www.cmswire.com/cms/enterprise-

20/what-enterprise-20-practitioners-should-know-about-km-deployments-007346.php).

[**Gal09**] Galago (2009) "Galago." Galago. Retrieved March 01, 2010 (http://www.galagosearch.org/).

[**Gar09**] Gardner, Dana (2009) "ZDNet - Enterprises seek better ways to discover, manage and master their information explosion headaches." Retrieved January 30, 2010 (http://www.zdnet.com/blog/gardner/enterprises-seek-better-ways-to-discover-manage-and-master-their-information-explosion-headaches/3091).

[**Gon01**] Gonzalez, J., A Rodriguez, and F. Llopis (2001) "University of Alicante at TREC-10." *Proceedings of TREC-10*.

[**Goo10**] Google (2010) "Currency Conversion." Retrieved April 01, 2010 (http://www.google.com/intl/en/help/features.html#currency).

[**Gor88**] Gordon, M (1988) "Probabilistic and genetic algorithms for document retrieval." *Communications of the ACM*.

[**GHM05**] Gospodnetic, Otis, Erik Hatcher, and Michael McCandless (2005) *Lucene in Action*. Manning Publications Co.

[**HAL02**] Hammo, Bassam, Ani Abu-Salem, and Steven Lytinen (2002) "QARAB: A Question Answering System to Support the Arabic Language.".

[**Hau04**] Haupt, Randy and Sue Haupt (2004) *Practical genetic algorithms*. John Wiley & Sons, Inc.

[**Hea97**] Hearst, Marti (1997) "TextTiling: segmenting text into multi-paragraph subtopic passages.".

[**Her09**] Hersh, William (2009) "A health and biomedical perspective." in *Information retrieval*. Springer.

[**HG01**] Hirschman, Lynette and Robert J. Gaizauskas (2001) "Natural language question answering: the view from here." *Natural Language Engineering* Volume 7(Issue 4).

[**Hol75**] Holland, John (1975) "Adaptation in Natural and Artificial Systems." *ACM SIGART Bulletin* (Issue 53).

[**IBM10**] IBM Lotus Notes (2010) "IBM Lotus software." IBM Lotus software. Retrieved April 2, 2010 (http://www-01.ibm.com/software/lotus/).

[**IDC07**] IDC (2007) "You are wasting time. Find out why." Retrieved April 1, 2010 (http://www.networkworld.com/news/2007/012307-wasted-searches.html).

[**IDC10**] IDC (2010) "The Digital Universe Decade, are you ready?" Retrieved June 21, 2010 (http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm).

[**Ind09**] Indri (2005) "Indri - Language modeling meets inference networks." Indri - Search engine from the Lemur project. Retrieved June 9, 2010 (www.lemurproject.org/indri/).

[**Irw79**] Irwin, Richard (1979) "The NPV Model of Strategy—The Shareholder Value Model." in *Financial Strategy: Studies in the Creation, Transer, and Destruction of Shareholder Value*.

[**IF00**] Ittycheriah, Abraham and Martin Franz (2000) "IBM's Statistical Question Answering System." *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*.

[**Jaw10**] Jaworski, Alexa (2010) "Insurance Networking News - You Store More and More Data. Can You Find It?" Retrieved May 1, 2010

(http://www.insurancenetworking.com/news/insurance_data_management_recovery_storage_ediscovery_compliance-24382-1.html).

[**KBH94**] Khuri, Sami, Thomas Back, and Jorg Heitkotter (1994) "The zero/one multiple knapsack problem and genetic algorithms." *Proceedings of the 1994 ACM Symposium on Applied Computing*.

[**Kno10**] Knorg (2009) "Knorg - knowledge management tool." Knorg. Retrieved April 3, 2010 (http://www.knorg.net/).

[**Kra99**] Kraft, D., G. Bordogna, and G Pasi (1999) *The Handbook of Fuzzy Sets Series*. MA: Kluwer Academic Publishers.

[**LCS97**] Lee, L., H. Chuang, and K. Seamons (1997) "Document Ranking and the Vector-Space Model.".

[**LF88**] Lee, W. and E. Fox (1988) "Experimental Comparison of Schemes for Interpreting Boolean Queries.".

[**Lem09**] Lemur (2002) "Lemur toolkit for language modeling and information retrieval." Retrieved June 09, 2010 (www.lemurproject.org/).

[**LM01**] Light, M., G. S. Mann, and et al. (2001) "Analyses for elucidating current question answering technology." *Natural Language Engineering* (Question Answering).

[**LQ03**] Lin, Jimmy, Dennis Quan, and et al. (2003) "WhatMakes a Good Answer? The Role of Context in Question Answering.".

[**Liu10**] Liu, Sunny (2002) "Introduction to Knowledge Management." Retrieved January 8, 2010 (http://www.unc.edu/~sunnyliu/inls258/Introduction_to_Knowledge_Management.html).

[**LC02**] Liu, Xiaoyong and Bruce Croft (2002) "Passage retrieval based on language models." *Conference on Information and Knowledge Management - Proceedings of the eleventh international conference on Information and knowledge management*.

[**LV01**] Llopis, Fernando and Jose L. Vicedo (2001) "IR-n, a passage retrieval system from University of Alicante, at Clef 2001.".

[**Luc00**] Lucene (2000) "Java search engine library." Apache Lucene. Retrieved June 09, 2010 (lucene.apache.org).

[**LSA10**] LuceneAPI (2010) "Lucene 3.0.1. API Similarity." Class Search.Similarity. Retrieved June 08, 2010 (http://lucene.apache.org/java/3_0_1/api/core/index.html).

[**Luc10**] LuceneQueryParser (2010) "Lucene API." QueryParser. Retrieved June 10, 2010 (http://lucene.apache.org/java/3_0_1/api/core/index.html).

[**Mal05**] Malhotra, Yogesh (2005) "Integrating knowledge management technologies in organizational business processes: getting real time enterprises to deliver real business performance." *Journal of Knowledge Management* Volume 9(Issue 1).

[**MRS08**] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008) *Introduction to Information Retrieval*. Cambridge University Press.

[**MT99**] Man, K. F., K. S. Tang, and at al. (1999) *Genetic Algorithms: Concepts and Designs*. London: Springer Verlag.

[**May02**] Maybury, Mark T. (2002) "Toward a Question Answering Roadmap.".

[**Meg10**] Megaputer (2010) "Megaputer - Data Mining and Text Mining." Megaputer Intelligence, Inc. Retrieved April 3, 2010 (http://www.megaputer.com/).

[**MG405**] MG4J (2005) "MG4J - Managing Gigabytes for Java." MG4J. Retrieved March 02, 2010 (http://mg4j.dsi.unimi.it/).

[**Mic96**] Michalewicz, Zbigniew (1996) "Genetic algorithms + data structures = Evolution Programs." 3rd ed. Springer.

[**MB07**] Middleton, Christian and Ricardo Baeza-Yates (2007) "A Comparison of Open Source Search Engines.".

[**Min08**] Minion (2008) "minion search engine." minion. Retrieved March 01, 2010 (https://minion.dev.java.net/).

[**Mit98**] Mitchell, Melanie (1998) *An Introduction to Genetic Algorithms*. The MIT Press.

[**Mol02**] Moldovan, D and D Passça (2002) "Performance issues and error analysis in an open-domain question answering system." *Proceedings of the 40th AnnualMeeting of the Association for Computational Linguistics (ACL)*.

[**Mur03**] Murphy, Rick (2003) "Structuring the unstructured (document)." Retrieved March 5, 2010 (http://www.allbusiness.com/management/1045200-1.html).

[**Nov98**] Novak, Joseph (1998) *Learning, creating, and using knowledge: Concept maps as facilitative tools in schools and corporations*. Lawrence Erlbaum Associates.

[**Nut09**] Nutch (2003) "Open source search engine based on Lucene Java for the search and index component." Nutch Serch Engine. Retrieved June 05, 2010 (nutch.apache.org/).

[**Nux10**] Nuxeo (2010) "Nuxeo Enterprise Content Management." Nuxeo. Retrieved June 8, 2010 (http://www.nuxeo.org/xwiki/bin/view/Main/).

[**Ope08**] OpenEphyra (2008) "OpenEphyra - QA system." OpenEphyra. Retrieved June 09, 2010 (http://www.ephyra.info/).

[**Par10**] Paradigm OpsLink (2010) "Paradigm OpsLink - WITSML Real Time Data Acquisition." Paradigm OpsLink. Retrieved June 08, 2010 (http://www.pdgm.com/products/opslink.aspx).

[**PGF00**] Pathak, Praveen, Michael Gordon, and Weiguo Fan (2000) "Effective Information Retrieval using Genetic Algorithms based Matching Functions Adaptation.".

[**Pow10**] PoweredBy (2010) "Applications and web applications using Lucene include." Retrieved June 22, 2010 (http://wiki.apache.org/lucene-java/PoweredBy).

[**Pul08**] Pulkkinen, Jyrki (2008) "Knowledge management and emerging technologies.".

[**Rap10**] Rapid-I (2010) "Rapid-I." Open-Source Data Mining. Retrieved April 6, 2010 (http://rapid-i.com/content/view/10/69/lang,en/).

[**Rob96**] Robertson and et al. (1996) "Sliding Window scored with Okapi BM25." *Okapi at TREC 4*.

[**RJ76**] Robertson, S. E. and K. S. Jones (1976) "Relevance weighting of search terms." *Journal of the American Society for Information Science*.

[**Roc71**] Rocchio, J. (1971) "Relevance feedback in information retrieval.".

[**Sal10**] Sales Force Content Library (2010) "Sales Content Manager Software." Sales Force. Retrieved June 8, 2010 (http://www.salesforce.com/crm/sales-force-automation/content-management/).

[**SAB93**] Salton, Gerald, James Allan, and Chris Buckley (1993) “Approaches to passage retrieval in full text information systems.”.

[**SB76**] Salton, Gerard and Christopher Buckley (1976) “Term-weighting approaches in automatic text retrieval.”.

[**SB98**] Salton, G. and C. Buckley (1998) “Term-weighting approaches in automatic text retrieval.” *IPM*.

[**SWY75**] Salton, G., A. Wong, and C. S. Yang (1975) “A Vector Space Model for Automatic Indexing.” *Communications of the ACM* Volume 18(Issue 11).

[**Sef10**] Self, Will (2010) “NewStatesman.” Retrieved June 22, 2010 (http://www.thisislondon.co.uk/standard/related-511-google-inc.do).

[**Sin09**] Singh, Vik (2009) “A Comparison of Open Source Search Engines.” Retrieved March 10, 2010 (http://zooie.wordpress.com/2009/07/06/a-comparison-of-open-source-search-engines-and-indexing-twitter/).

[**Sin01**] Singhal, Amit (2001) “Modern information retrieval: a brief overview.” *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.

[**Sol09**] Solr (2004) “Open-source search server based on the Lucene Java search library.” Solr. Retrieved June 03, 2010 (lucene.apache.org/solr/).

[**Spa71**] Sparck, Jones (1971) “Automatic Keyword Classification for Information Retrieval.”.

[**TK03**] Tellex, Stefanie, Boris Katz, and et al. (2003) “Quantitative Evaluation of Passage Retrieval Algorithms for Question.” *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*.

[**Ter04**] Terrier (2004) “Terrier IR Platform.” Terrier. Retrieved March 02, 2010 (http://terrier.org/).

[**Tie05**] Tiedemann, Jorg (2005) *Integrating linguistic knowledge in passage retrieval for question answering*. Vancouver: Association for Computational Linguistics.

[**Tie07**] Tiedemann, Jorg (2007) *A comparison of genetic algorithms for optimizing linguistically informed IR in Question and Answering*. Springer-Verlag.

[**Tik10**] Tika (2010) “Tika - Content Analysis Toolkit.” Retrieved June 22, 2010 (http://tika.apache.org/).

[**TRE10**] TREC (2010) “TREC QA track.” (http://trec.nist.gov/tracks.html).

[**Tru10**] Trueknowledge (2010) “trueknowledge.” Retrieved June 05, 2010 (http://www.trueknowledge.com/).

[**TC99**] Turtle, Howard and Bruce Croft (1999) “Inference Networks for Document Retrieval.”.

[**UKO10**] UKOM (2010) “UK Online Measurement.” Retrieved June 1, 2010 (http://googlebarometer.blogspot.com/2010/06/latest-insights-trends-from-google.html).

[**UAG04**] Usunier, Nicolas, Massih Amini, and Patrick Gallinari (2004) “Boosting Weak Ranking Functions to Enhance Passage Retrieval for Question Answering.” *SIGIR04*.

[**Wek10**] Weka-3 (2010) “Weka 3: Data Mining Software.” Weka 3. Retrieved March 10, 2010 (http://www.cs.waikato.ac.nz/ml/weka/).

[**WMB99**] Witten, Ian, Alistair Moffat, and Timothy Bell (1999) “Managing Gibabytes: Compressing and Indexing Documents and Images.” 2nd ed. Morgan Kaufmann Publishing.

[**Wol10**] Wolframalpha (2010) “wolframalpha.” Retrieved June 08, 2010 (http://www.wolframalpha.com/).

[**Wra07**] Wray, Richard (2007) “How one year's digital output would fill 161bn iPods.” Retrieved June 22, 2010 (http://www.guardian.co.uk/media/2007/mar/06/newmedia.business).

[**Wra10**] Wray, Richard (2010) “Goodbye petabytes, hello zettabytes.” guardian.co.uk. The Guardian. Retrieved May 10, 2010 (http://www.guardian.co.uk/technology/2010/may/03/humanity-digital-output-zettabyte).

[**Wum05**] Wumpus (2005) “wumpus.” wumpus. Retrieved March 01, 2010 (http://www.wumpus-search.org/).

[**Xap03**] Xapian (2003) “xapian - Open Source Search Engine Library.” xapian. Retrieved January 03, 2010 (http://xapian.org/).

[**YK93**] Yang, J. and R. Korfhage (1993) ““Query Optimization in Information Retrieval Using Genetic Algorithms.” *Proceedings of the fifth International Conference on Genetic Algorithms*.

[**YM94**] Yuret, D. and M. Maza (1994) “A genetic algorithm system for predicting the oex.” *Technical Analysis of Stocks & Commodities*.

[**Zac99**] Zack, Michael H. (1999) “Managing Codified Knowledge.” *Sloan Management Review* Volume 40(Issue 4).

[**Zet04**] Zettair (2004) “Zettair - compact and fast text search engine.” Zettair. Retrieved March 01, 2010 (http://www.seg.rmit.edu.au/zettair/).

Appendix A

Test Questions and Judgements

This section contains all the questions that were used to test the Information Retrieval system, along with the suggestions to the answer. In order an answer be considered relevant, it has to follow the given guidelines.

1. **what is all commodity volume value**
   All commodity Volume (value) (ACV) represents the total annual sales volume of retailers.

2. **what elements can create visual merchandising**
   Displays, colour, lighting, space, product information, sensory inputs such as smell, touch, and sound as well as technologies such as digital displays and interactive installation.

3. **what are the techniques for trade sales promotions**

- Trade allowances: short term incentive offered to induce a retailer to stock up on a product.
- Dealer loader: An incentive given to induce a retailer to purchase and display a product.
- Trade contest: A contest to reward retailers that sell the most product.
- Point-of-purchase displays: Extra sales tools given to retailers to boost sales.
- Training programs: dealer employees are trained in selling the product.
- Push money: an extra commission paid to retail employees to push products.

4. **where is located the largest shopping mall**
   The Dubai Mall, located in Dubai, United Arab Emirates

5. **how Tesco uses technology, in order to innovate**
   Tesco implemented self-service till and cameras to reduce queues

6. **what is a supply network**
   Processes carried out at facility nodes and over distribution link.

7. **examples of supply network**
   Supply chain is a special instance of a supply network in which raw materials, intermediate materials and finished goods are procured exclusively as products through a chain of processes that supply one another.

8. **what is a supply chain**
   Supply chain is a system of organizations, people, technology, activities, information and resources involved in moving a product or service from supplier to customer.

9. **what problems address supply chain management**
   Distribution Network Configuration, Distribution Strategy, Trade-Offs in Logistical Activities, Integration of processes through the supply chain to share valuable information, Inventory Management, Cash-Flow.

10. **what is a manufacturer's suggested retail price**
    Price which the manufacturer recommends that the retailer sell the product.

11. **what are the functions of a store manager**
    Store manager is the person responsible for the day-to-day operations and management of a retail store. (It deals with factors such as Human resources, store business operations, product management, team development, problem solving)

12. **in what countries operates SPAR**
    32 countries: Austria, Italia, UK, South Africa, Germany, Spain, Ireland, Norway, Hungary, Denmark, France, Japan, Belgium, Greece, Slovenia, Czech Republic, Netherlands, Switzerland, Russia, Zimbabwe, Poland, Australia, Botswana, Swaziland, Namibia, Croatia, China, Mauritius, Zambia, Ukraine, Romania, India.

13. **what factors involve service delivery**
    Equipment used to provide the service (e.g. vehicles, cash registers, technical systems, and computer systems), physical facilities (e.g. buildings, parking, waiting rooms), requesting service consumer, other customers at the service delivery location, Customer contact.

14. **what are the characteristics of services**
    The characteristics of services are: Intangibility, Perishability, Inseparability, Simultaneity, and Variability.

15. what are second-hand goods

Goods that are being purchased-by or otherwise transferred to a second or later end user.

16. what is a consignment shop

"Consignment shop" is an American English term for second-hand stores that offer used goods at a lower cost than new.

17. what are the sales techniques

The sale process can be made through: Direct sales (involving person to person contact), pro-forma sales, agency-based, travelling salesman, and request for proposal (an invitation for suppliers, through a bidding process, to submit a proposal on a specific product or service).

18. what are the key performance indicators of the sales stores used for

The Key Performance Indicators indicate whether or not the sales process is being operated effectively and achieves the results as set forth in sales planning. It should enable the sales managers to take timely corrective action deviate from projected values. It also allows senior management to evaluate the sales manager.

19. what is a sales promotion

Media and non-media marketing communication are employed for a pre-determined, limited time to increase consumer demand, stimulate market demand or improve product availability.

Sales promotions targeted at the consumer are called consumer sales promotions. Sales promotions targeted at retailers and wholesale are called trade sales promotions. Some sale promotions, particularly ones with unusual methods, are considered gimmick by many.

20. what is sales planning

Sales planning involve predicting demand for the product and demand on the sales assets (machines, people, or a combination of both). Planning insures that when a consumer wishes to purchase the product, the product is available, but it also means opportunities for additional sales are presented and the sales assets are available to exploit these opportunities. Planning should allow for meeting increasing customer demand for more products, services and/or customization as the business is growing, but also react quickly when demand decreases. Sales planning improve efficiency and decreases unfocused and uncoordinated activity within the sales process.

21. how the sales process is planned

Sales forecasting (gather data on past sales, analyze trends and report forecasts), demand planning (validate forecasts, revise inventory and customer service policies), supply planning (ability to meet demand by reviewing available capacity and scheduling required operations), match supply and demand plans with financial considerations, finalize the plan and release it to implementation.

22. what is a transaction broker

This is where the salesperson doesn't represent either party, but handles the transaction only.

23. what is RFID

Radio-frequency identification or RFID is the use of an object/tag applied to or integrated into a product, animal, or human being with the intention of identification and tracking using radio waves.

24. what is retailing business

Retailing consists of the sale of goods or merchandise from a fixed location, such as a department store, boutique or kiosk, or by mail, in small or individual lots for direct consumption by the purchaser.

25. what is a reseller

A reseller is a company or individual that purchases goods or services with the intention of reselling them rather than consuming or using them.

26. what is psychological pricing

Psychological pricing or price ending is a marketing practice based on the theory that certain prices have a psychological impact. The retail prices are often expressed as "odd prices": a little less than a round number, e.g. $19.99 or $2.98.

27. in retailing, how products are called

At a retail in-store level, merchandising refers to the variety of products available for sale.

28. what is product placement

Product placement, or embedded marketing, is a form of advertisement, where branded goods or services are placed in a context usually devoid of ads (movies, story line of television shows, news programs). The product placement is often not disclosed at the time that the good or service is featured.

29. what is a POS

Point of sale (POS) or checkout is the location where a transaction occurs. A "checkout" refers to a POS terminal or more generally to the hardware and software used for checkouts, the equivalent of an electronic cash register.

A POS terminal manages the selling process by a salesperson accessible interface. The same system allows the creation and printing of the voucher.

30. what are the objectives of packing and labelling

Physical protection, Barrier protection, Containment or agglomeration, Information transmission, Marketing, Security, Convenience, and Portion control.

31. what is merchandising

Merchandising is the methods, practices, and operations used to promote and sustain certain categories of commercial activity. In the broadest sense, merchandising is any practice which contributes to the sale of products to a retail consumer.

32. what is flow operations positions

Flow Operation Positions are positions used to store pallets that arrive to the warehouse.

33. give me an example of inventory strategy

a) **Just-in-time** (**JIT**) is an inventory strategy that strives to improve a business's return on investment by reducing in-process inventory and associated carrying costs. To meet JIT objectives, the process relies on signals between different points in the process, which tell production when to make the next part. Implemented correctly, JIT can improve a manufacturing organization's return on investment, quality, and efficiency.

b) **Just in Sequence** (**JIS**) is an inventory strategy that matches JIT and complete fit in sequence with variation of assembly line production. Components and parts arrive at a production line right in time as scheduled before they get assembled. Feedback from the manufacturing line is used to coordinate transportation to and from the process area. When implemented successfully, JIS improves a company's return on assets (ROA), without loss in flexibility, quality or overall efficiency.

34. Where inventory management is required and what's for?

Inventory management is required at different locations within a facility or within multiple locations of a supply network to protect the regular and planned course of production against the random disturbance of running out of materials or goods. The scope of inventory management also concerns the fine lines between replenishment lead time, carrying costs of inventory, asset management, inventory forecasting, inventory valuation, inventory visibility, future inventory price forecasting, physical inventory, available physical space for inventory, quality management, replenishment, returns and defective goods and demand forecasting.

35. what is B2C

Business-to-consumer (B2C) describes activities of businesses serving end consumers with products and/or services.

36. what is the main wholesaling process

Wholesaling is the sale of goods to retailers, to industrial, commercial, institutional, or other professional business users, or to other wholesalers and related subordinated services. Wholesalers frequently physically assemble sort and grade goods in large lots, break bulk, repack and redistribute in smaller lots.

37. what is wardrobing

Practice of purchasing an item, using it, and then returning it to the store for a refund.

38. what is ARI

Oracle Retail Active Retail Intelligence (ARI) is an exception management and workflow tool driven by business rules. It spans the Oracle Retail Merchandising Operations Management applications and provides a central repository for alert notifications and associated actions across all Oracle-based applications.

39. what is oracle retail allocation

Oracle Retail Allocation helps retailers determine the inventory requirements at the item and location level, which results in an inventory allocation that optimizes the supply across all locations.

40. what is oracle retail invoice matching

Oracle Retail Invoice Matching is made for retailers who want to better manage reconciliation and payment of supplier invoices.

41. what applications ARI uses

ARI is a monitoring system that interacts with any applications database. As such it does not use any information from other retailing applications; rather, it monitors the retailing application databases for events defined by a client and notifies the client when said events occur.

42. what is an enterprise process modification

A business example of an Enterprise Process Modification is a client changing its transfer approval process from analysts being able to approve transfers to only managers being able to approve transfers.

43. what is RTM

Oracle Retail Trade Management (RTM) is a management system designed to integrate and streamline the international trade transaction process. RTM links multiple departments together for all import functions. RTM provides immediate online visibility to the status, location, and disposition of products as they move through the import cycle.

44. what is ReSA

Oracle Retail Sales Audit (ReSA) is an auditing system that provides a simplified sales audit process that accepts raw point of sale (POS) data and provides clean data to downstream applications, while ensuring integrity of audited data. The application is designed to focus on exception conditions, while allowing clean data to flow through thus increasing productivity.

45. what RPM supports

Oracle Retail Price Management (RPM) is a pricing and promotions execution system.

RPM supports pricing strategies you can use to define how item retails are proposed when pricing worksheets are generated. The strategies are defined at department, class, or subclass to represent which items are affected. RPM also supports creation of vendor-funded promotions, by either associating an existing deal from RMS with the promotion, or by creating a new deal in RMS based on the information provided for the promotion.

46. RMS is web-based?

Yes, RMS is a Web-based forms application that provides direct database access.

47. What is RMS

Oracle Retail Merchandising System (RMS) is the foundation system that records and controls virtually all data in the retail enterprise and ensures data integrity across all integrated systems.

48. How oracle retail supports the invoice verification process

Oracle Retail Invoice Matching (ReIM) module supports the invoice verification process with accuracy and efficiency, focusing resources on exception management.

The application helps identify, review, and resolve errors and irregularities in a timely manner.

49. How to integrate oracle retail allocation with other applications

Oracle Retail Allocation uses a Java architecture built on a layered model. Layers of the application communicate with one another through an established hierarchy and are only able to communicate with neighbouring layers. The application is divided into a presentation layer, a middle tier consisting of services and business objects, and a database access/driver layer.

50. What is the name of the database to manage users, on the oracle retail allocation

Users are managed via a user table on the Oracle Retail Allocation database called ALC_USERS.

51. How to do a pre-pack allocation

A pre-pack is a package containing multiple items for distribution. Oracle Retail Allocation has the capability to define optimum pre-packs for clients in addition to allocating them. This function is used to minimize shipping and handling costs.

52. How to integrate RMS and RPM

RPM uses a Java architecture built on a layering model. RPM exists on the same database schema as RMS which allows information to be shared between applications through direct database reads, package calls, and batch processes.

53. What is the module where we create purchase orders

Purchase Order module allows creating and maintaining purchase orders in a variety of ways.

54. What mass return transfers are used for

Mass return transfers are used to reallocate merchandise to locations or to return merchandise to the supplier.

55. Where the inventory adjustments are created

Inventory adjustments are created in RMS or imported from an external system (store or warehouse application). Inventory adjustments can also be created by locations for multiple items, by item for multiple locations, or through a product transformation for a specific location.

56. RMS supports single sign on?

Yes, Oracle currently provides three different implementations: Oracle Single Sign-On (SSO), Java SSO and Oracle Access Manager.

57. What are the levels of security of RMS

The three levels of security offered by RMS are:

- Database level security, a built in feature of Oracle Database, based on database roles
- Application level security, form or screen level security based on database roles
- Data level security, built into RMS to give a client the ability to further limit user access to information.

58. How to handle invoice discrepancies

ReIM resolution dialog offers an approach to handling invoice discrepancies where reviewers can disposition a discrepancy based on a set of user-defined reason codes.

59. Where do we deal with taxes and duties on import merchandising

For calculation of tax and duties applicable on import merchandise, the Harmonized Tariff Schedule (HTS) files need to be uploaded into ORTM system.

60. Where do we deal with replenishment

RMS includes key functions such as item maintenance, inventory management, and replenishment.

Appendix B

Test Queries to the Information Retrieval system

This section contains all the queries used to test the Information Retrieval system.

1. is “all commodity volume” value
2. elements create visual merchandising
3. techniques trade sales promotions
4. located largest shopping mall
5. Tesco uses technology innovate
6. “supply network”
7. examples “supply network”
8. is supply chain
9. problems address “supply chain management”
10. manufacturer’s suggested retail price
11. functions “store manager”
12. countries operates SPAR
13. factors involve service delivery
14. characteristics services
15. is second-hand goods
16. is “consignment shop”
17. is sales techniques

18. key performance indicators sales stores
19. is sales promotion
20. is sales planning
21. sales process planned
22. is transaction broker
23. is RFID
24. is retailing business
25. is reseller
26. is psychological pricing
27. retailing products called
28. is product placement
29. is POS
30. objectives packing labelling
31. is merchandising
32. is flow operations positions
33. example inventory strategy
34. inventory management required
35. is B2C
36. is main wholesaling process
37. is wardrobing
38. is ARI
39. is “oracle retail allocation”
40. is “oracle retail invoice matching”
41. applications ARI uses
42. is enterprise process modification
43. is RTM
44. is ReSA
45. RPM supports
46. RMS is web-based
47. is RMS
48. “oracle retail” supports invoice verification process
49. integrate “oracle retail allocation” other applications
50. is name database manage users “oracle retail allocation”
51. do pre-pack allocation
52. integrate RMS RPM
53. is module create “purchase orders”
54. mass return transfers used
55. inventory adjustments created
56. RMS supports single sign on
57. levels security RMS
58. handle invoice discrepancies
59. deal taxes duties import merchandising
60. deal replenishment

Appendix C

Initial Population and First Generation

This section contains one example for the initial population and the first generation from the genetic algorithm applied to the question "*what problems address supply chain management*".

Initial Population

0 {1.1642439246177674} problems address "supply chain management"
1 {0.5979545950889588} problems address scm
2 {1.1642439246177674} (problems+) address "supply chain management"
3 {1.1642439246177674} problems (address+) "supply chain management"
4 {1.1642439246177674} problems address "supply chain management"
5 {1.7064804673194884} address "supply chain management"
6 {1.6695516586303711} problems "supply chain management"
7 {0.8858792901039123} problems address
8 {1.2245667040348054} problem* address "supply chain management"
9 {1.1642439246177674} problems address "supply chain management"
10 {1.1642439246177674} problems address "supply chain management"
11 {1.2228458404541016} (problems^0.06) address "supply chain management"
12 {1.188597273826599} problems (address^0.85) "supply chain management"
13 {1.1642439246177674} problems address "supply chain management"
14 {1.2190988540649415} (problems^0.44) address "supply chain management"
15 {1.2318716764450073} problems (address^0.38) "supply chain management"

16 {1.1642439246177674} problems address "supply chain management"
17 {1.0673438727855682} (problems^1.66) address "supply chain management"
18 {1.2201648831367493} problems (address^0.59) "supply chain management"
19 {1.1642439246177674} problems address "supply chain management"
20 {1.081891030073166} (problems^1.57) address "supply chain management"
21 {1.2218770146369935} problems (address^0.57) "supply chain management"
22 {1.1642439246177674} problems address "supply chain management"
23 {1.167803168296814} (problems^0.97) address "supply chain management"
24 {1.01723655462265} problems (address^1.75) "supply chain management"
25 {1.1642439246177674} problems address "supply chain management"
26 {1.0930389583110809} (problems^1.5) address "supply chain management"
27 {1.2192655682563782} problems (address^0.6) "supply chain management"
28 {1.1642439246177674} problems address "supply chain management"
29 {1.0770652770996094} (problems^1.6) address "supply chain management"
30 {0.9993495166301727} problems (address^1.86) "supply chain management"
31 {1.1642439246177674} problems address "supply chain management"
32 {1.0608203887939454} (problems^1.7) address "supply chain management"
33 {1.0554223954677582} problems (address^1.56) "supply chain management"
34 {1.1642439246177674} problems address "supply chain management"
35 {1.1618127942085266} (problems^1.02) address "supply chain management"
36 {1.2273653030395508} problems (address^0.16) "supply chain management"
37 {1.1642439246177674} problems address "supply chain management"
38 {1.0131011843681335} (problems^1.99) address "supply chain management"
39 {1.2146865844726562} problems (address^0.02) "supply chain management"
40 {1.1642439246177674} problems address "supply chain management"
41 {1.2163266837596893} (problems^0.49) address "supply chain management"
42 {1.2286136269569397} problems (address^0.47) "supply chain management"
43 {1.1642439246177674} problems address "supply chain management"
44 {1.2213737547397614} (problems^0.02) address "supply chain management"
45 {1.0493529260158538} problems (address^1.59) "supply chain management"
46 {1.1642439246177674} problems address "supply chain management"
47 {1.1070854187011718} (problems^1.41) address "supply chain management"
48 {1.2075899362564086} problems (address^0.71) "supply chain management"
49 {1.1642439246177674} problems address "supply chain management"
50 {1.0147467017173768} (problems^1.98) address "supply chain management"
51 {0.5979545950889588} (problems+) address scm
52 {0.5979545950889588} problems (address+) scm
53 {0.5979545950889588} problems address (scm+)
54 {0.9537012279033661} address scm
55 {0.6898506283760071} problems scm
56 {0.8858792901039123} problems address

57 {0.6619910359382629} problem* address scm
58 {0.5979545950889588} problems address scm
59 {0.5979545950889588} problems address scm
60 {0.6265807509422302} (problems^0.76) address scm
61 {0.6921885848045349} problems (address^1.69) scm
62 {0.5975716650485993} problems address (scm^1.53)
63 {0.5628164649009705} (problems^1.28) address scm
64 {0.5002392143011093} problems (address^0.03) scm
65 {0.5944427758455276} problems address (scm^1.2)
66 {0.5585358917713166} (problems^1.45) address scm
67 {0.655838206410408} problems (address^1.37) scm
68 {0.6437590301036835} problems address (scm^0.17)
69 {0.5612210869789124} (problems^1.35) address scm
70 {0.5501150012016296} problems (address^0.7) scm
71 {0.6130264699459076} problems address (scm^1.84)
72 {0.6140198111534119} (problems^0.87) address scm
73 {0.6939595758914947} problems (address^1.71) scm
74 {0.6260355144739151} problems address (scm^0.64)
75 {0.5550522863864898} (problems^1.59) address scm
76 {0.6108743667602539} problems (address^1.07) scm
77 {0.6442616671323776} problems address (scm^0.21)
78 {0.6441254556179047} (problems^0.58) address scm
79 {0.6690300583839417} problems (address^1.47) scm
80 {0.630799463391304} problems address (scm^0.58)
81 {0.6594694972038269} (problems^0.02) address scm
82 {0.49853086471557617} problems (address^0.02) scm
83 {0.6048111140727996} problems address (scm^0.9)
84 {0.6187169492244721} (problems^0.83) address scm
85 {0.6545880675315857} problems (address^1.36) scm
86 {0.5940454006195068} problems address (scm^1.38)
87 {0.556365692615509} (problems^1.52) address scm
88 {0.539523133635521} problems (address^0.52) scm
89 {0.6441456794738769} problems address (scm^0.28)
90 {0.556059592962265} (problems^1.66) address scm
91 {0.538047480583191} problems (address^0.5) scm
92 {0.6422104686498642} problems address (scm^0.11)
93 {0.6588261365890503} (problems^0.35) address scm
94 {0.5066847771406173} problems (address^0.07) scm
95 {0.5966213047504425} problems address (scm^1.03)
96 {0.6481529653072358} (problems^0.53) address scm
97 {0.5339502364397049} problems (address^0.35) scm

98 {0.6090540200471878} problems address (scm^0.84)

99 {0.6615726232528687} (problems^0.09) address scm

Generation number 1

0) {1.2228458404541016} (problems^0.06) address "supply chain management"

(1) {1.2273653030395508} problems (address^0.16) "supply chain management"

(2) {1.2286136269569397} problems (address^0.47) "supply chain management"

(3) {1.2318716764450073} problems (address^0.38) "supply chain management"

(4) {1.2201648831367493} problems (address^0.59) "supply chain management"

(5) {1.2218770146369935} problems (address^0.57) "supply chain management"

(6) {1.7064804673194884} address "supply chain management"

(7) {1.6695516586303711} problems "supply chain management"

(8) {1.2213737547397614} (problems^0.02) address "supply chain management"

(9) {1.2245667040348054} problem* address "supply chain management"

(10) {0.5339502364397049} (problems^1.0) (address^0.675) (scm^0.5)

(11) {0.6422104686498642} problems address scm

(12) {0.6422104686498642} (problems^1.33) (address^1.0) (scm^0.555)

(13) {1.0673438727855682} (problems^1.66) address

(14) {0.5585358917713166} (problem*^1.72) (address^1.0) (scm^1.0)

(15) {1.2218234300613404} (problem*^1.99) address

(16) {0.5628164649009705} (problems^1.1400000000000001) (address^0.51) (scm^0.5)

(17) {0.49853086471557617} (problems^1.28) address scm

(18) {1.1642439246177674} (problems^0.5) (address+) "supply chain management"

(19) {1.7064804673194884} problems (address+) "supply chain management"

(20) {1.0673438727855682} (problems^1.8199999999999998) (address^1.0) "supply chain management"

(21) {1.0147467017173768} (problems^1.98) address "supply chain management"

(22) {0.6260355144739151} (problems^1.0) (address^0.515) (scm^0.32)

(23) {0.5002392143011093} problems address (scm^0.64)

(24) {1.0554223954677582} (problems^1.0) (address^1.56) "supply chain management"

(25) {1.0554223954677582} problems (address^1.56) "supply chain management"

(26) {1.1642439246177674} (problems^0.5) (address^1.0) "supply chain management"

(27) {0.97644402384758} problems address "supply chain management"

(28) {0.556365692615509} (problems^1.26) (address^0.76) (scm^0.5)

(29) {0.539523133635521} (problems^1.52) address scm

(30) {1.2163266837596893} (problems^0.745) (address^1.0) "supply chain management"

(31) {0.6442616671323776} problems address "supply chain management"

(32) {0.6615726232528687} (problems^0.545) (address^0.675) (scm^0.5)

(33) {0.5339502364397049} problems address scm

(34) {0.5979545950889588} (problems^1.0) (address^0.515) (scm+)

(35) {0.5561325341463089} problems address (scm+)
(36) {0.6265807509422302} (problems+) (address^1.0) (scm^1.0)
(37) {1.1642439246177674} (problems+) address
(38) {0.6626725286245346} (problems^1.0) (address^0.71) (scm^1.0)
(39) {1.6695516586303711} problems (address^1.42)
(40) {0.4985308647155761 7} (problems^1.0) (address^0.51) (scm^0.5)
(41) {0.5966213047504425} problems address scm
(42) {0.6130264699459076} (problems^0.5) (address^1.0) (scm^0.92)
(43) {1.9560451745986938} problems address (scm^1.84)
(44) {0.6594694972038269} (problems^0.51) (address^1.0) (scm^0.5)
(45) {0.8858792901039123} problems address scm
(46) {0.6187169492244721} (problems^0.915) (address^1.0) (scm^0.5)
(47) {0.5966213047504425} problems address scm
(48) {0.6843425184488297} (problems^0.765) (address^1.495) (scm^0.5)
(49) {0.650012880563736} problems (address^1.71) scm
(50) {1.1070854187011718} (problems^1.205) (address^0.8) "supply chain management"
(51) {1.2192655682563782} (problems^1.41) address "supply chain management"
(52) {0.5940454006195068} (problems^1.0) (address^1.2349999999999999) (scm^0.69)
(53) {0.6690300583839417} problems (address^1.47) (scm^1.38)
(54) {0.655838206410408} (problems^1.0) (address^1.185) (scm^0.5)
(55) {0.5975716650485993} problems (address^1.37) scm
(56) {1.01723655462265} (problems^1.0) (address^1.375) "supply chain management"
(57) {0.5975716650485993} problems (address^1.75) "supply chain management"
(58) {0.8858792901039123} (problems^1.0) (address+)
(59) {0.638615220785141} problems (address+)
(60) {0.5975716650485993} (problems^1.0) (address^1.0) (scm^1.53)
(61) {0.5975716650485993} problems address (scm^1.53)
(62) {0.6265807509422302} (problems^0.88) (address^1.425) (scm^0.5)
(63) {0.7050789892673492} problems (address^1.85) scm
(64) {0.5585358917713166} (problems^1.555) (address^1.0) (scm^1.0)
(65) {1.0673438727855682} (problems^1.66) address
(66) {1.1618127942085266} (problems^0.995) (address^1.0) "supply chain management"
(67) {1.167803168296814} (problems^1.02) address "supply chain management"
(68) {0.5940454006195068} (problems^1.0) (address^1.0) (scm^0.69)
(69) {0.8858792901039123} problems address (scm^1.38)
(70) {0.630799463391304} (problems^1.0) (address^1.0) (scm^0.43)
(71) {0.6441456794738769} problems address (scm^0.58)
(72) {0.6619910359382629} (problem*^1.225) (address^1.0) (scm^0.5)
(73) {0.5585358917713166} (problem*^1.45) address scm
(74) {0.8858792901039123} (problems+) (address^1.0)
(75) {0.9537012279033661} (problems+) address

(76) {0.5066847771406173} (problems^1.0) (address^0.535) (scm^0.5)
(77) {0.6441456794738769} problems address scm
(78) {0.5002392143011093} (problems^1.0) (address^0.515) (scm^0.5)
(79) {0.6437590301036835} problems address scm
(80) {0.6090540200471878} (problems^1.0) (address^0.85) (scm^0.42)
(81) {0.5501150012016296} problems address (scm^0.84)
(82) {0.5944427758455276} (problems^1.0) (address^0.7949999999999999) (scm^1.1)
(83) {1.2201648831367493} problems address
(84) {0.5501150012016296} (problem*^1.0) (address^0.85) (scm^0.5)
(85) {0.6619910359382629} problem* address scm
(86) {0.539523133635521} (problems^1.25) (address^0.76) (scm^1.0)
(87) {1.0930389583110809} (problems^1.5) address
(88) {0.6545880675315857} (problems^1.0) (address^1.1800000000000002) (scm^0.5)
(89) {0.6422104686498642} problems (address^1.36) scm
(90) {0.5585358917713166} (problems^1.225) (address^1.0350000000000001) (scm^0.5)
(91) {0.6108743667602539} (problems^1.45) (address^1.07) scm
(92) {0.4985308647155761 7} (problems^0.765) (address^0.51) (scm^0.5)
(93) {0.6481529653072358} problems address scm
(94) {1.0673438727855682} (problems^1.555) (address^1.0) "supply chain management"
(95) {0.5585358917713166} (problems^1.66) address "supply chain management"
(96) {1.1642439246177674} (problems+) (address^1.0) "supply chain management"
(97) {0.9537012279033661} (problems+) address "supply chain management"
(98) {0.8858792901039123} (problems^1.46) (address^1.0350000000000001)
(99) {0.5610016077756882} (problems^1.92) (address^1.07)

Best Top 10 Chromosomes from the first Generation:

0 (problems^0.06) address "supply chain management" [1.2228458404541016]
1 problems (address^0.16) "supply chain management" [1.2273653030395508]
2 problems (address^0.47) "supply chain management" [1.2286136269569397]
3 problems (address^0.38) "supply chain management" [1.2318716764450073]
4 problems (address^1.42) [1.6695516586303711]
5 problems address (scm^1.84) [1.9560451745986938]
6 address "supply chain management" [1.7064804673194884]
7 problems "supply chain management" [1.6695516586303711]
8 problems (address+) "supply chain management" [1.7064804673194884]
9 problem* address "supply chain management" [1.2245667040348054

Appendix D

Fitness Evolution using Sentence Retrieval

The following Figure 28 shows the evolution of the score fitness function from the Genetic Algorithm, considering the top 10 individuals from each generation. It was used Sentence Passage Retrieval and the question analyzed used as a test was "*what problems address supply chain management*".

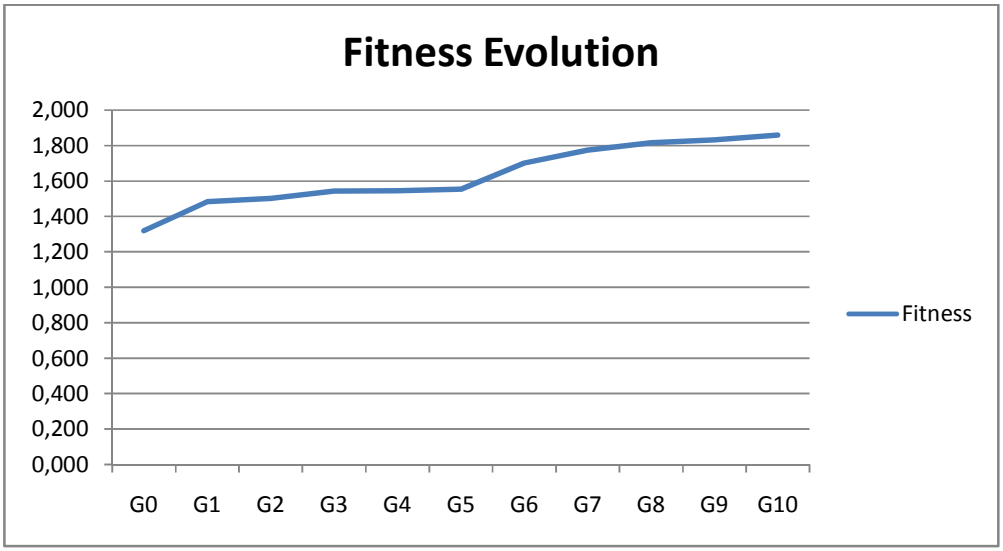

Figure 28 - Fitness evolution for the top 10 individuals using sentence retrieval

Appendix E

Fitness Evolution using Paragraph Retrieval

The following Figure 29 shows the evolution of the score fitness function from the Genetic Algorithm, considering the top 10 individuals from each generation. It was used Paragraph Passage Retrieval and the question analyzed used as a test was "*what problems address supply chain management*".

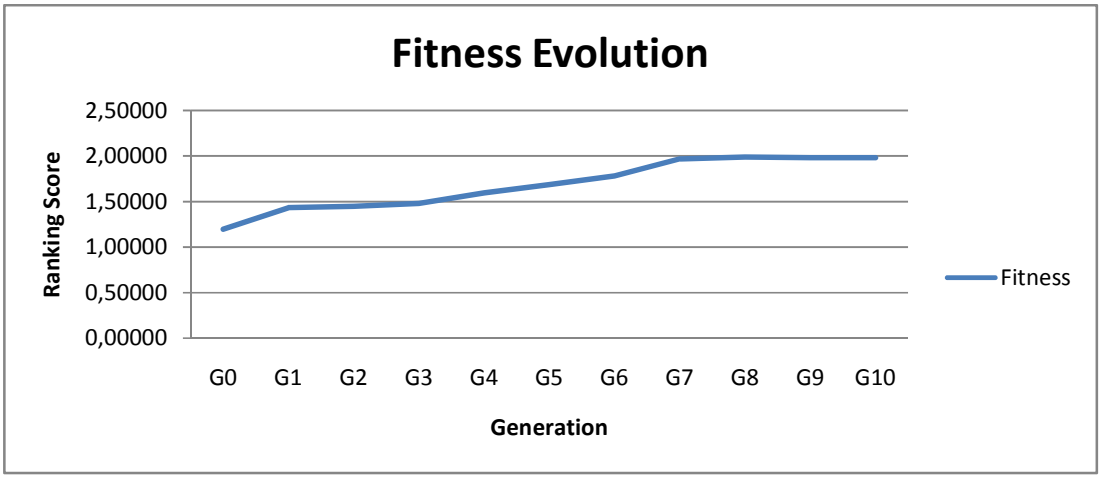

Figure 29 - Fitness evolution for the top 10 individuals using paragraph retrieval

Appendix F

Fitness Evolution using Fixed-Window Retrieval

The following Figure 30 shows the evolution of the score fitness function from the Genetic Algorithm, considering the top 10 individuals from each generation. It was used a Fixed-Window of five sentences Passage Retrieval and the question analyzed used as a test was "*what problems address supply chain management*".

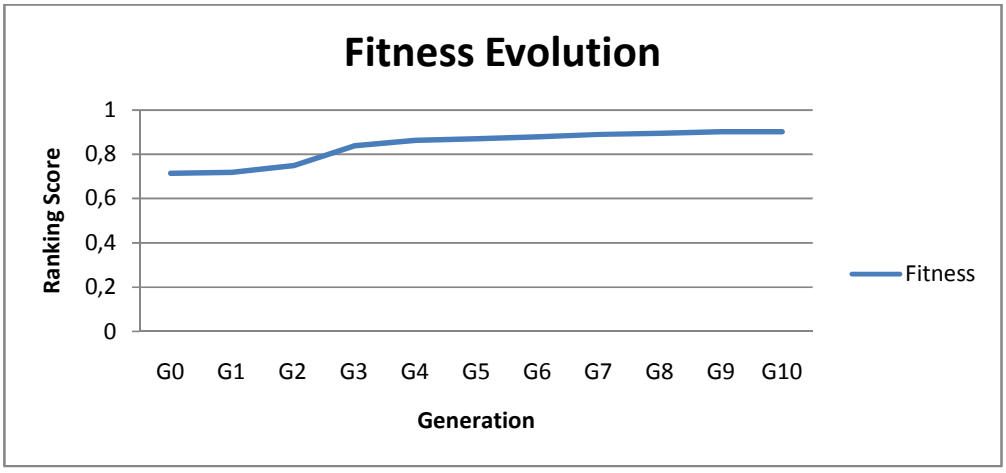

Figure 30 - Fitness evolution for the top 10 individuals using fixed-window retrieval

Appendix G

Comparison between Passage Retrieval methods

The following Figure 31 shows the comparison from the evolution of the score fitness function from the Genetic Algorithm, considering the top 10 individuals from each generation. It was compared: Sentence Passage Retrieval, Paragraph Passage Retrieval, and a Fixed-Window of five sentences Retrieval. The question analyzed used as a test was "*what problems address supply chain management*".

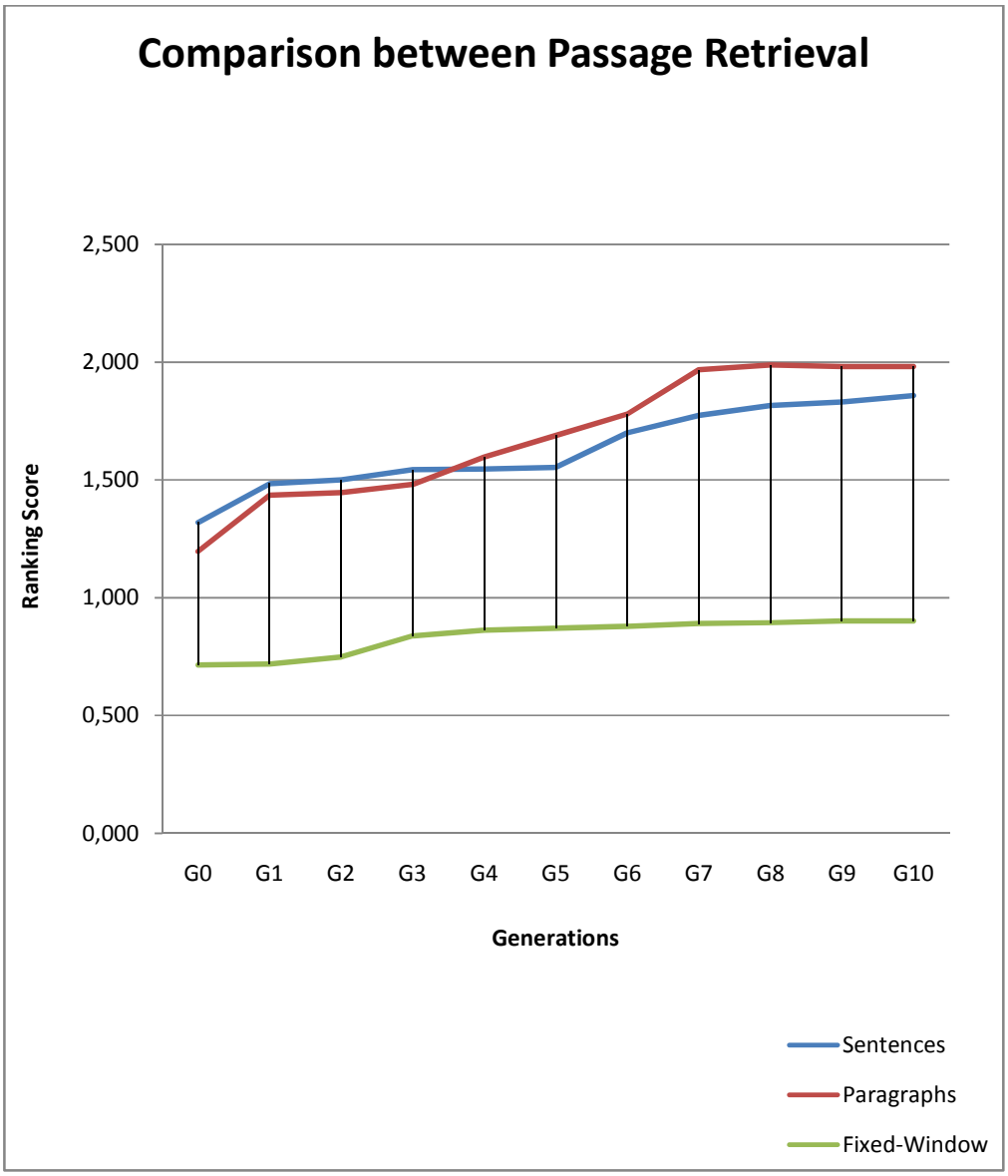

Figure 31 – Fitness evolution, comparing3 different Passage Retrieval models

Appendix H

Average Precision with Sentence Retrieval

This section shows a chart presenting the average precision in each question, using Sentence Passage Retrieval.

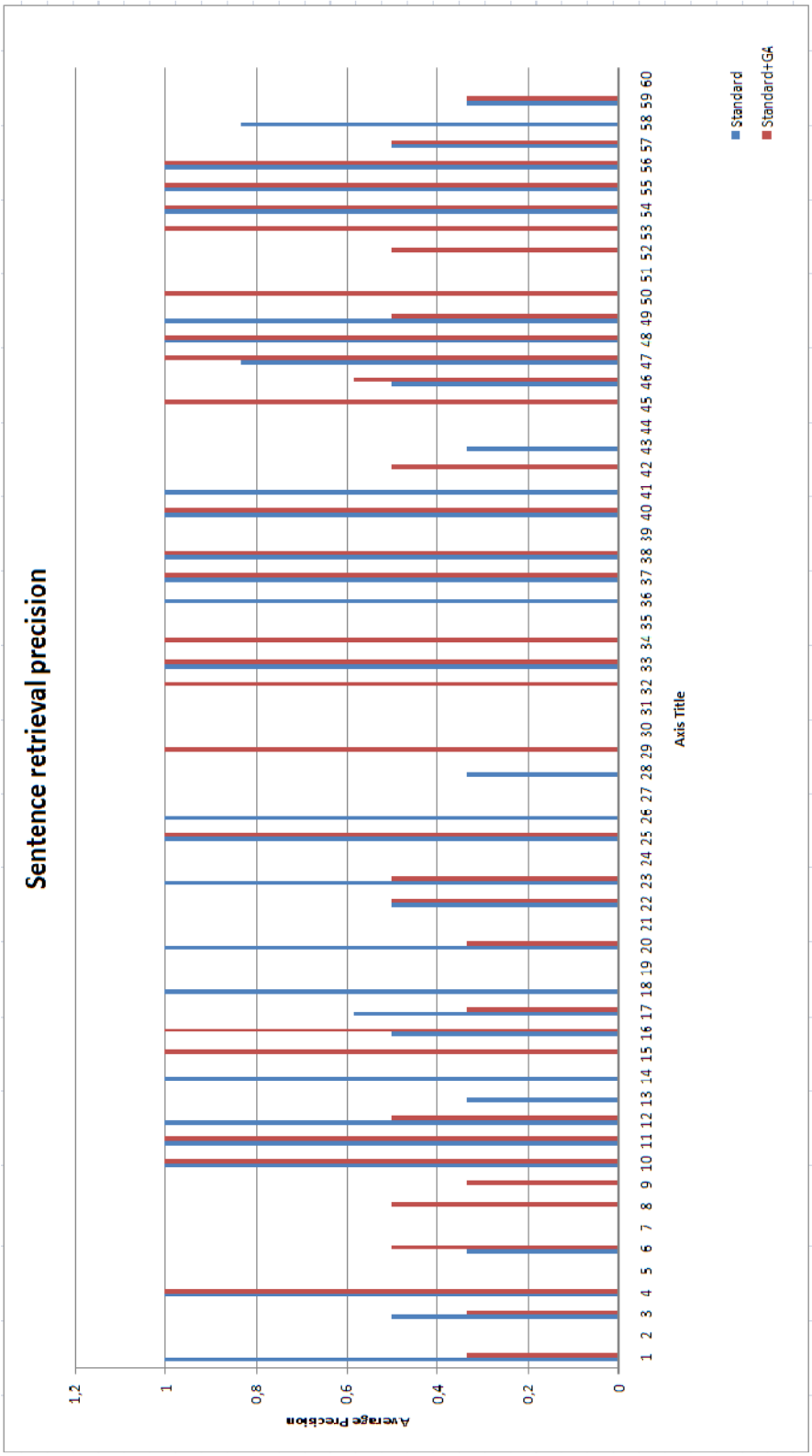

Figure 32 - Average precision using sentence passage retrieval

Appendix I

Average Precision with Paragraph Retrieval

This section shows a chart presenting the average precision in each question, using Paragraph Passage Retrieval.

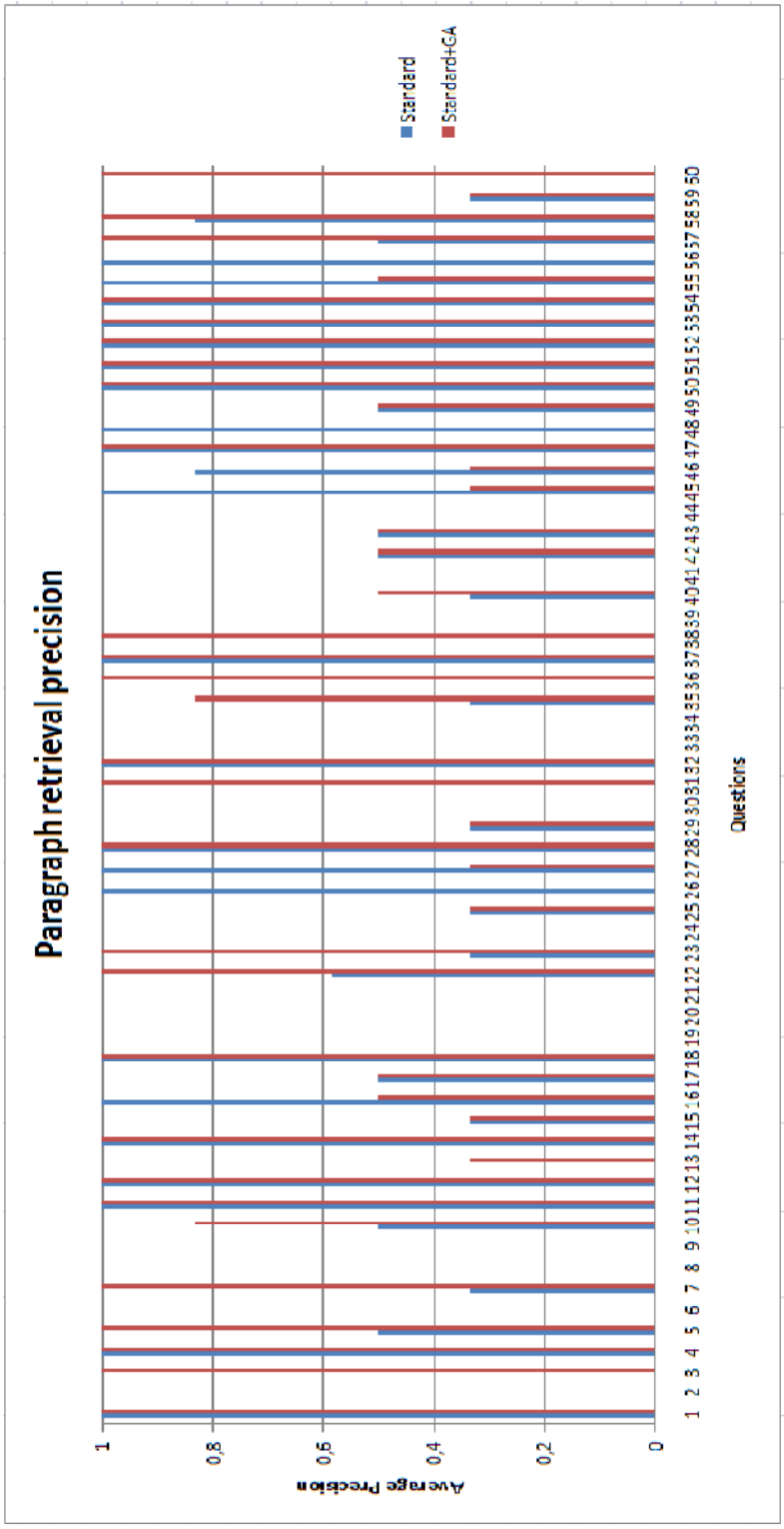

Figure 33 - Average precision using paragraph passage retrieval

Appendix J

Average Precision with Fixed-Window Retrieval

This section shows a chart presenting the average precision in each question, using a Fixed-Window of five Sentences, Passage Retrieval.

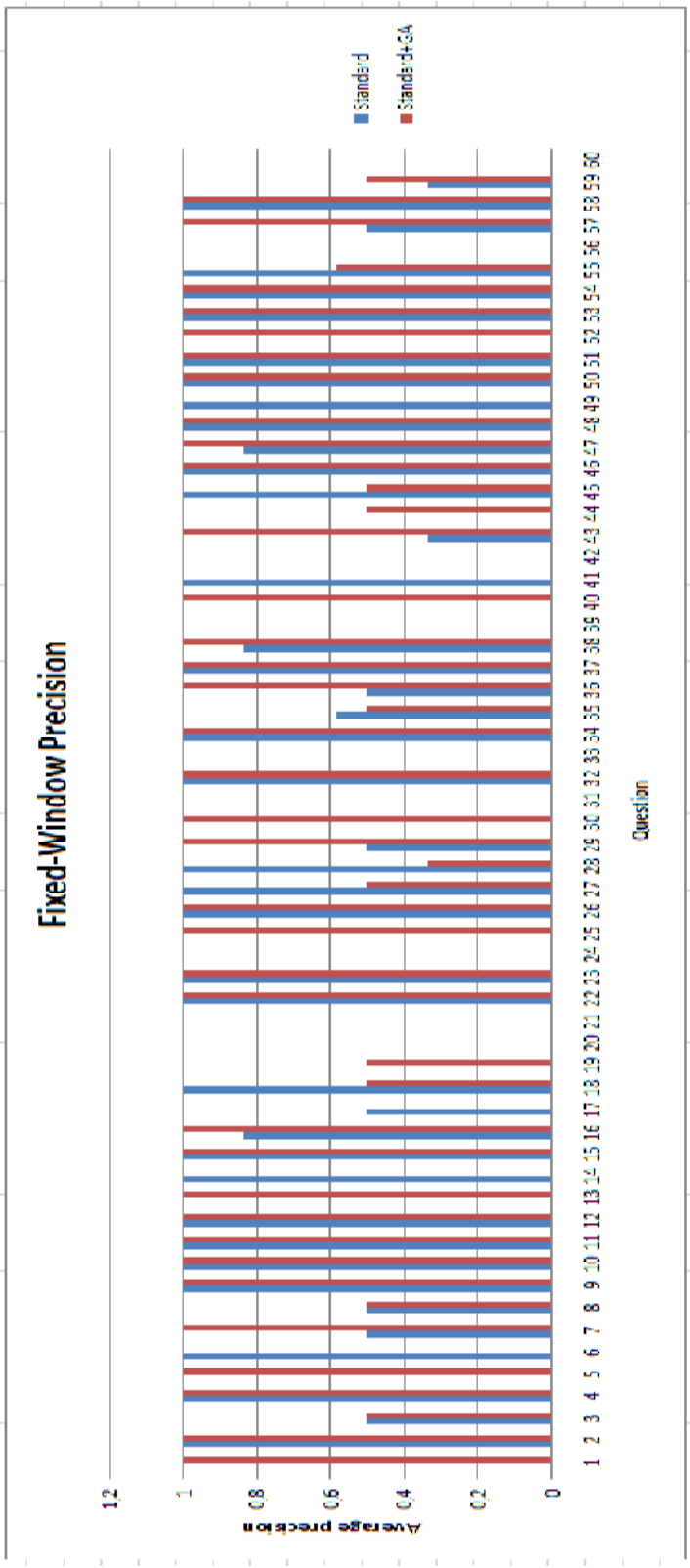

Figure 34 - Average precision using fixed-window passage retrieval