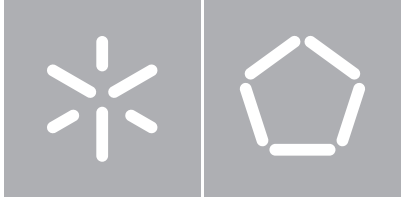
Universidade do Minho
Escola de Engenharia

Mauro José Ferreira de Freitas

Personalized approach on a smart image search engine, handling default data

Outubro de 2014



Universidade do Minho
Escola de Engenharia

Mauro José Ferreira de Freitas

Personalized approach on a smart image search engine, handling default data

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de

Professor Cesar Analide

Outubro de 2014

Name: Mauro José Ferreira de Freitas

E-mail: maurojffreitas@gmail.com

Number Id: 13952947

Title: Personalized approach on a smart image search engine, handling default data

Title PT: Abordagem de Personalização num motor de busca de imagens, tratamento de informação desconhecida ou incerta

Supervisor: Cesar Analide

Conclusion Year: 2014

Master Degree Designation: Master's Degree in Informatics Engineering

Minho's University 29/10/2014

Signature: Mauro José Ferreira de Freitas

Acknowledgements

I would like to thank my supervisor Cesar Analide who gave me the opportunity to develop this project under his guidance and helped me on my academic growth in the last two years during my master's degree journey.

I would also like to show my appreciation for professors José Maia Neves and Henrique Vicente who helped me in the development of scientific contribution.

To Bruno Fernandes I would like to thank the support provided during the course of this project. His contribution was essential in the past few months during the completion of this project and also in carrying out the scientific contributions.

To all my friends, thanks for everything we've been through in the last five years since we were admitted in Informatics Engineering. Your support was very important to complete this stage of my academic life.

Finally I thank my parents and sisters who were always available to help me, providing all conditions for the achievement of all my objectives.

Abstract

Search engines are becoming a tool widely used these days, there are different examples like Google, Yahoo!, Bing and so on. Adjacent to these tools is the need to get the results closer to each one of the users. Is in this area that some work has been developed recently, which allowed users to take advantage of the information presented to them, with no randomness or a sort of generic factors.

This process of bringing the user closer to the results is called Personalization. Personalization is a process that involves obtaining and storing information about users of a system, which will be used later as a way to adapt the information to present. This process is crucial in many situations where the filtering of content is a requirement, since we deal daily with large amounts of information and it is not always helpful.

In this project, the importance of personalization will be evaluated in the context of intelligent image search, making a contribution to the project CLOUD9-SIS. So, it will be evaluated the information to be treated, how it will be treated and how it will appear. This evaluation will take into account also other examples of existing search engines. These concepts will be used later to integrate a new system of searching for images, capable of adapting its results depending on the preferences captured from user interactions. The usage of the images was only chosen because CLOUD9-SIS is intended to return images as a result, it was not developed or used any technique for image interpretation.

Keywords: Personalization; Case-Based Reasoning; Intelligent Systems; Search Engines; Handling Default Data

Resumo

Os motores de busca estão a tornar-se uma ferramenta bastante utilizada nos dias de hoje, existindo diferentes exemplos, tais como Google, Yahoo!, Bing, etc. Adjacente a essas ferramentas surge a necessidade de aproximar cada vez mais os resultados produzidos a cada um dos utilizadores. É nesta área que tem sido desenvolvido algum trabalho recentemente, o que permitiu que os utilizadores, pudessem tirar o melhor proveito da informação que lhes é apresentada, não havendo apenas uma aleatoriedade ou factores de ordenação genéricos.

A este processo de aproximação do utilizador aos resultados dá-se o nome de Personalização. A Personalização é um processo que consiste na obtenção e armazenamento de informações sobre os utilizadores de um sistema, para posteriormente serem utilizadas como forma de adequar a informação que se vai utilizar. Este processo é determinante em várias situações onde a filtragem dos conteúdos é um requisito, pois lidamos diariamente com grandes quantidades de informação e nem sempre esta é útil.

Neste projecto, vai ser avaliada a preponderância da Personalização no contexto da pesquisa inteligente de imagens, dando um contributo ao projecto CLOUD9-SIS. Assim, será avaliada a informação a ser tratada, a forma como será tratada e como será apresentada. Esta avaliação terá em consideração também exemplos de outros motores de busca já existentes. Estes conceitos serão posteriormente utilizados para integrar um novo sistema de procura de imagens que seja capaz de adaptar os seus resultados, consoante as preferências que vão sendo retiradas das interacções do utilizador. O uso das imagens foi apenas escolhido porque o projecto CLOUD9-SIS é suposto retornar imagens como resultado, não foi desenvolvida nem utilizada nenhuma técnica de interpretação de imagens.

Palavras-chave: Personalização; Raciocínio Baseado em Casos; Sistemas Inteligentes; Motores de busca; Tratamento de informação incompleta ou desconhecida

Contents

List of Figures

Abbreviations

ANN	Artificial Neural Network
CBR	Case-Based Reasoning
GEA	Genetic and Evolutionary Algorithms
IS	Intelligent Systems
API	Application Programming Interface
DoC	Degree of Confidence
QoI	Quality of Information
DFT	Density Functional Theory
CD	Constitutional Molecular Descriptor
QD	Quantum-Chemical Molecular Descriptor
BP-ANN	Back Propagation Artificial Neural Network
EA	Evolutionary Algorithms
GA	Genetic Algorithms

1. Introduction

Search engines are technologies that have long appeared in our day-to-day. However, they are still evolving, searching for a solution adapted to the concept of each user.

This issue has been a concern for the major search engines that gradually turned their results more targeted to each user. This means that the information itself is not the only factor influencing the results.

This connection of the information with the user gains a new dimension, contributing greatly to the efficiency of the search. This improvement was confirmed by the research made by Yandex, which is the most used search engine in Russia. They include countries such as Ukraine, Belarus and Kazakhstan only by the use of Personalization. Results showed that users click 37% more on the first result when personalized, making the process of finding the information the user is looking for 14% faster (Kruger A., 2012).

The scope of this project is to take a non-personalized search engine and manipulate it using techniques of Intelligent Systems, this manipulation is intended to make user's will a filtering parameter.

This issue has been addressed in various search engines, however there is little information on the techniques used, so this project shows a way to use the power of intelligent systems techniques to create a support system for a search engine.

Almost all search engines are beginning to use the personalization, however they use it in different ways. This is because personalization can be based in Q&A forms or in users historic. Within these two categories, personalization can be based only on the responses to forms or else the combination of the user with the preferences of similar users (Analide C. and Novais P., 2012).

Besides personalization there are other techniques related to the extraction of information that have to be studied with the intention to create a more reliable system. In this subject there are a few possibilities however this project will explore the capacities of the CBR. The decision for this technique will be explained next.

The problem described above can be partially solved using the previous searches, which are stored on a kind of database. This procedure is very connected to the philosophy of CBR

systems, which can solve a problem by getting information from previous experiences kept on a knowledge base.

An example is the medical health screening which in most cases are based on previous experiences. According to Aamodt and Plaza this technique is a possibility for problems where the amount of information is wide and where the past experiences could lead to a possible solution. These last experiences provide helpful information needed to solve the current case.

However CBR can only be applied when there's something similar between cases that can be kept and then used as a first step to achieve a solution to a new problem (Balke T. et al, 2009).

This technique has a wide range of application, and this was expressed by Khan, Hoffmann and Kolodner, which uses CBR for medical problems or by Davide Carneiro who came up with a solution to the law problem of dispute resolution (Carneiro D., 2009).

Besides all this great features, the availability of data and the cost of obtain it, continues to be the major problem within this technique.

Aamodt and Plaza in 1994 defined a model that is nowadays widely accepted when speaking about CBR.

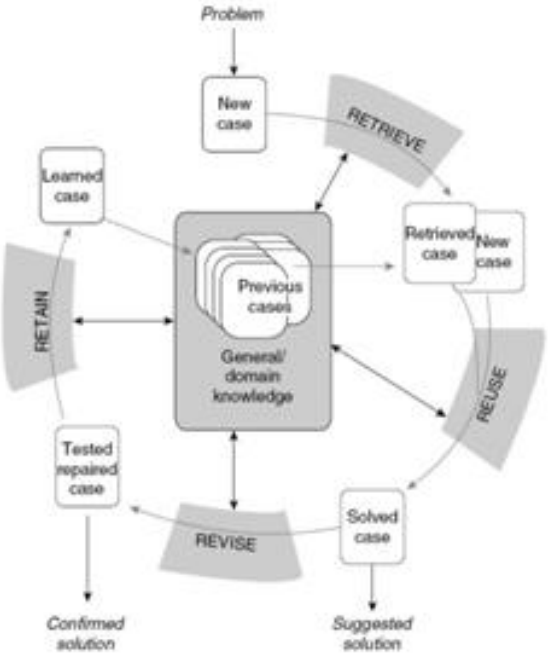


Figure 1 - Typical CBR Cycle (Aamodt and Plaza, 1994).

As seen on Figure 1 there a few vocabularies that were adopted. For example, a problem is expressed as a case and the previous cases are kept on a domain knowledge. They also clarify that, in order to work with CBR, the first step is to define the structure of the case. This structure is almost always composed by the problem, the description and the solution however it is possible to be adapted according the scope of the problem. This structure will be important from now on, it will have to be kept for every cases and will have to include every information helpful for the search of a solution.

Having the new case structured correctly the following step is to retrieve from the knowledge base the cases more similar to the new case. This measure of similarity is limited by a threshold that can be suited to each case study.

With the retrieved cases and the new case obtained, both could be combined on the reuse phase to provide a solution. After this step the solution is confirmed and can be tested and adapted before being established as a final solution, which would be stored on the domain knowledge.

The revise phase is one of the most important because is where the case is tested and where the user can interact with the suggested solution. At this stage the solution can be adapted or corrected providing more reliability to the cases that are stored. This interaction can solve some problems that automatic adaptation can't. Kolodner in 1992 defined the main adaptation techniques as the adaptation of attributes of the case, the usage of rules and the analog transfer (Kolodner J., 1992).

While a valuable solution, that will be stored on the domain knowledge, were not achieved, the process can become iterative.

1.1 Motivation

Based on an existing search engine, the motivation lies in the possibility of creating an intelligent system able to capture information from users so it can be used later.

Many search engines use personalization, however there are few references on the techniques used and they are often related to the location and previous searches of the user.

This gap has been detected and became the biggest motivating factor, since some new paths can be explored in regards to this issue.

Another motivation lies on the possibility to create a new CBR model able to deal with unknown information in a much effective way than the already developed systems. This new approach is being studied and now it can be improved and implemented to retrieve better results. There is also the fact that this approach was never applied to this area, which represents an additional challenge to this work.

Finally this work is intended to be applied on a mobile application, so another motivation is the development of the application itself however in regards to the knowledge extraction there's the motivation of creating a system able to communicate with many other systems.

1.2 Objectives

The goal of this project is to develop a mobile application that provides a search engine with the possibility to perform a regular search and with an intelligent and customizable module, able to provide to the users some content perfectly suited to his particular taste, his preference, i.e., his wills. This work is intended to provide a new approach to a search engine with new techniques, using the images only as the result to the analysis of the query performed. The images won't be handled by any particular system that could help when trying to find some matches to the query presented to the system. However, the system will be developed on a way that some additional modules could be added on future to enhance the solution we want to create. This will give the possibility to establish a connection between cases also by analyzing some features of the images such as colors or shadows.

To do this, techniques of knowledge analysis and extraction were studied, in order to create models able to determine the most effective way to collect information from the user and the searches. These techniques must be capable of handling large amounts of different kinds of information, such as uncertain or unknown information, and still be able to produce a result in good time.

In regards to the uncertain or unknown information there's also the objective of adapting a new approach that will be studied to the context of this project. This approach will have to be capable of dealing with the storage of large amounts of data.

Related to the knowledge extraction, in this work it will communicate with a mobile application, however since the beginning an objective is to create a cross platform solution, so

the knowledge extraction system will be able to communicate with web applications for instance, once the search engines are mostly used on web (Google, Yahoo, etc.).

Finally the objective to the mobile application is to create a user friendly environment with the simplicity of many other search engines, that allows the user to instinctively reach his purposes of finding the right content.

1.3 Structure of the document

This document will be split in seven parts. The first one where will be explained the project, establishing a context, the motivation to explore this path and which objectives are intended to be reached.

Then there's the state of the art where is described how search engines are dealing with this subject, showing when possible examples of how it's made. It is also made an analysis of some intelligent systems techniques that were thought to be used in this project. This analysis contains also some advantages, disadvantages and application areas for each technique.

The following chapter is related to the technologies that were used in the implementation proposed in this project. Here there are some explanations of how the technologies operate and how they could be useful in our implementation. There is also a conclusion about the analysis performed on the previous chapter in regards of the intelligent system technology to use.

The fourth chapter is one of the most important chapters of this thesis. The Implementation chapter is divided in three sub chapters where is detailed the major decisions taken in the creation of the back office, the data model and the mobile application.

To complement this new approach and in regards of the handling default data technique, it was created a case study that was described on chapter five. This case study helps to understand how the CBR system will handle the repository and how the new cases will be handled when presented to the system.

The second last chapter is where the results of the work performed over this project are described. Here some conclusions are taken in regards of the techniques that were chosen.

Finally the last chapter is the conclusion and future work. This chapter is divided in three major topics, the synthesis of the work performed, the relevant work such as scientific contributions and some guidelines about what is being done and what could be done in future in order to take this project to the next level.

2. State of the Art

After choosing the theme, a research was made in order to understand what has been done in this area. It is well known that the major search engines such as Google, Yahoo!, Bing and so on, use personalization in their searches. This factor is also used in other systems that are not exactly a search engine such as Amazon or eBay. However it is unclear which technique is used to filter and suggest the results.

It is known however that the vast majority of users of the systems mentioned above are against the collection of information during their use. Yet only 38% of users know how to work around this situation (Barysevich A., 2012).

2.1. Intelligent Systems

Since the purpose is the development of an intelligent system, there are some possibilities to be considered related to the existing techniques.

There are many techniques and each one has its pros and cons, so the choice of the technique required a reflection to realize which one suited well in this project. After this reflection it was chosen the technique considered closer to the project requirements.

There are 3 main techniques used with regards to intelligent systems, they are the Artificial Neural Networks, Case-Based Reasoning and Genetic and Evolutionary Algorithms.

2.1.1 Artificial Neural Network

Artificial neural networks are computational systems that seek for a problem solution using, as base structure, the central nervous system of humans. It is also considered an artificial neural network any processing structure whose paradigm is connectionism.

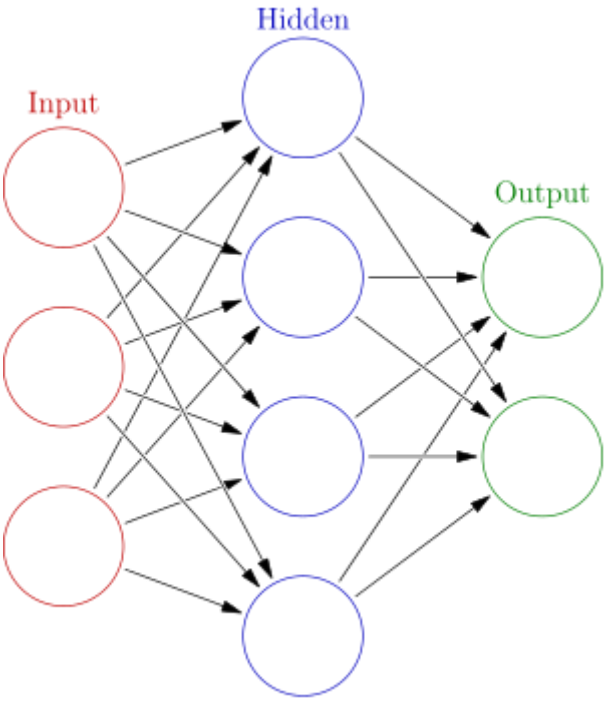


Figure 2 - Example of a ANN structure, showing a possibility of interconnections between neurons

2.1.2 Case-Based Reasoning

The case-based reasoning is a paradigm that consists in solving problems, making use of previous cases that have occurred. When a problem is proposed, a similarity to those observed earlier is calculated in order to see which one is closest. Discovered that similar case the same solution is applied.

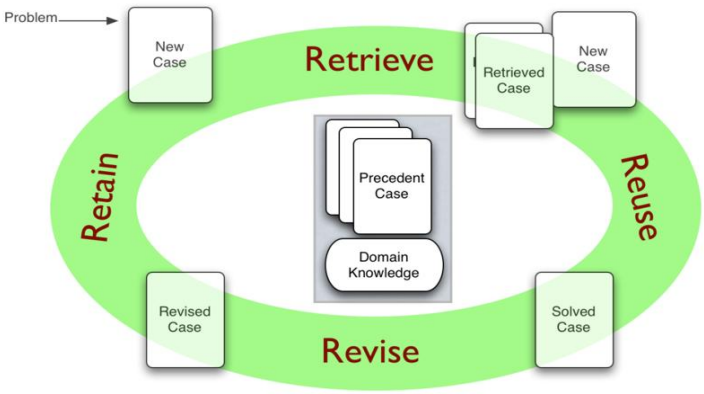


Figure 3 - CBR paradigm illustration.

When applying this type of paradigm, first step consists in creating a structure for each case, which is usually composed of the problem itself, the solution and the justification.

The choice for this kind of reasoning is usual in areas where knowledge from experience is vast, when there are exceptions to the rule and when learning is crucial (Analide C., et al., 2012;Kolodsen J., 1992;Aamodt A. and Plaza E., 1994).

As it was said before, the range of a CBR system is extremely wide. A few examples are medical diagnosis, law, assisted learning, diets and so on.

According to Riesbeck and Shank or Tsinakos, CBR systems can improve assisted student learning. SYIM_ver2 is a software developed to monitor the student's performance, saving his learning needs or even answering the student's questions. The reasoning of this system is based on a CBR (Riesbeck C. and Schank R., 1991;Tsinakos A., 2003).

This system identifies similar questions and uses them to answer a new one. The search of similar questions is made according to a key search feature. The PISQ (Process of Identification of Similar Questions) is responsible to find and provide a solution on Educational knowledge base.

SYIM_ver 2 allows the user to make two kinds of searches, a free text search and a controlled environment search, which was an advantage when compared with the previous version.

In some countries is now becoming possible to solve law problems using previous experiences. This fact opened new paths for CBR systems to explore the law issue and there are many examples, one of them offers a solution to the online dispute resolution. This problem is recent since we now face a new era with a huge technology focus, which became the traditional courts not capable to deal with that amount of cases.

With the acceptance of this process new systems start to appear. In 2009/2010, Davide Carneiro presented a solution that uses past experiences to achieve a quicker solution to the real problem (Carneiro D., et al., 2009; Carneiro D., et al., 2010).

Previously in 1989 Rissland and Ashley developed a model based on hypothetical and actual cases which gave him the name HYPO. (Rissland E. and Ashley K., 1989) The input is from the user and is called Current Fact Simulation (CFS), where each case includes a description, the arguments, the explanation and justification.

The output is provided as a case-analysis-record with HYPOS analysis; a claim lattice showing graphic relations between the new case and the cases retrieved by the system; 3-ply arguments used and the responses of both plaintiff and defendant to them; and a case citation summary (Rissland E. and Ashley K., 1989). Related to the domain knowledge, it is structured in three places: the cases-based of knowledge (CBK), the library of dimensions and the normative standards. CBK is where the cases are stored and ,the dimensions encode legal knowledge about a certain point of view.

HYPO was a start for other systems that took advantage of the great results on legal reasoning. TAX-HYPO on tax law (Rissland E. and Skalak D., 1989), CABARET for income tax law (Skalak D. and Rissland E., 1992) and IBP for predictions on argumentation concepts (Brüninghaus S. and Ashley K., 2003) are some of the examples that derived from HYPO.

Kolodner in 1992 and Khan and Hoffmann in 2002 presented two solutions for two different problems such as medical diagnosis and diet recommendations, respectively (Kolodner J., 1992; Khan A. and Hoffman A., 2002). In case of medical diagnosis is important to filter from a huge amount of diseases and symptoms, to achieve the better treatment, as quickly as possible. The first reasoning within doctors is to use previous cases solved, to select the best solution to the current case. Making comparison with the CBR reasoning, the doctor receives a new case (patient) with a specific disease and symptoms. After achieving a solution, this new case became part of the doctor knowledge. Future cases can be easily and quickly solved based on the knowledge, improving the speed of the solution and decreasing the expenses. The CBR system will also improve the reliability and the speed of the process. In 1993 Kolodner suggested that cases should be presented in a form of a triple <problem, solution, justification>.

Besides that, the only purpose of the system is to "suggest", the final decision should be validated and adapted, if needed, by the doctor, giving more quality to the cases retained.

Regarding to the diet recommendation, this system is able to provide a menu specific to some personal features such as medical conditions, calorie restrictions, nutrients requirements or even culture issues. In 2002 Khan and Hoffman proved that a system using CBR is able to recommend a diet, fitting the patient needs.

Once more the solution is only a recommendation the expert should adapt, if needed, and then evaluate the solution provided by the available cases of the knowledge base.

On the other hand, there are several CBR software tools developed and tested, which are being applied in real situations, namely Art*Enterprise, developed by ART Inference Corporation

(Watson I., 1996). It offers a variety of computational paradigms such as a procedural programming language, rules, cases, and so on. Present in almost every operating system, this tool contains a GUI builder that is able to connect to data in most of proprietary DBMS formats. It also allows the developer to access directly the CBR giving him/her more control of the system, which in most situations may turn into a drawback.

The CBR itself is also quite limited since the cases are represented as flat values (attribute pairs). The similarity strategy is also unknown, which may represent another constraint since there is no information about the possibility of adapting it. These features are presented on Figure 4.

CasePower which is a tool developed using CBR. It was developed by Inductive Solutions Inc.(Watson I., 1996) and uses the spreadsheet environment of Excel. The confines of the Excel functionalities are echoed on the CBR framework, making it more suitable for numerical applications. The retrieval strategy of CasePower is the nearest neighbor, which uses a weighted sum of given features. In addition to this strategy the system attaches an index to each case in advance, improving the system performance in the retrieval process. The adaptation phase in this system may be made through formulas and macros from Excel but the CBR system in itself cannot be adapted as explained in Figure 4.

CBR2 denotes a family of products from Inference Corporation, and may be the most successful CBR. This family is composed by CBR Express which stands for a development environment, featuring a customer call tracking module; the CasePoint which is a search engine for cases developed by CBR Express; the Generator that allows the creation of cases-bases through MS Word or ASCII files; and finally the Tester which is a tool capable of providing metrics for developers. In this tool the cases structure comprises a title, a case description, a set of weighted questions (attribute pairs), and a set of actions. Such as CasePower, CBR2 uses the nearest neighbor strategy to retrieve the cases initially. Cases can also be stored in almost every proprietary database format. However, the main and most valuable feature of CBR2 is the ability to deal with free-form text. It also ignores words like and, or, I, there, just to name a few. It also uses synonyms and the representation of words as trigrams, which makes the system tolerant to spelling mistakes and typing errors.

The EasyReasoner that is a module within Eclipse, being extensively regulated since it is available as a C library, so there is no development interface and is only suitable for experienced C programmers. This software developed by Hayley Enterprises (Watson I., 1996) is very similar

to ART and also ignores noise words and use trigrams to deal with spelling mistakes. Once more, these tools are not likely to be adapted because it is an API, and the similarity strategy is not even known.

The ESTEEM software tool, which was developed by Esteem Software Inc., has its own inference engine, a feature that allows the developer to create rules, providing a control over the induction process. Moreover, cases may have a hierarchy that can narrow the searches, which is very useful when accessing multiple bases and nested cases. The matching strategy is also the nearest neighbor. The advantages of this tool are the adaptability of the system and the usage of two different similarity strategies (nearest neighbor and induction). A disadvantage is the fact that it only runs on Windows (Watson I., 1996).

jCaBaRe is an API that allows the usage of Case-Based Reasoning features. It was developed using Java as a programming language, which gives this tool the ability to run in almost every operating system. As an API, jCaBaRe has the possibility to be adapted and extended providing a solution for a huge variety of problems. The developer can establish the cases attributes, the weight of each attribute, the retrieval of cases, and so on. One of its main limitations is the fact that it still requires a lot of work, by the developer, to achieve an working system. Several features must be developed from scratch.

The Kate software tool that was produced by Acknosoft (Althoff K-D., et al., 1995) and is composed by a set of tools such as Kate-Induction, Kate-CBR, Kate-Editor, and Kate-Runtime. The Kate-Induction tool is based on induction and allows the developer to create an object representation of cases, which can be imported from databases and spreadsheets. The induction algorithm can deal with missing information. In these cases, Kate is able to use the background knowledge. The Kate-CBR is the tool responsible for case comparison and it uses the nearest neighbor strategy. This tool also enables the customization of the similarity assessments. Kate-Editor is a set of libraries integrated with ToolBook, allowing the developer to customize the interface. Finally, Kate-Runtime is a set of interface utilities connected to ToolBook that allows the creation of an user application.

Another CBR system is the ReMind, a software tool created by Cognitive Systems Inc. (Watson I., 1996). The retrieval strategies are based on templates, nearest neighbor, induction and knowledge-guided induction. The templates retrieval is based on simple SQL-like queries, the nearest neighbor focus on weights placed on cases features and the inductive process can be made either automatically by ReMind, or the user can specify a model to guide the induction

algorithm. These models will help the system to relate features providing more quality to the retrieval process. The ReMind is available as a C library. It is also one of the most flexible CBR tools, however it is not an appropriate tool for free text handling attributes. The major limitations of this system are the fact that cases are hidden and cannot be exported, and the excessive time spent on the case retrieval process. The nearest neighbor algorithm is too slow, and besides the fact that inductive process is fast, the construction of clusters trees is also slow (Watson I., 1996).

The characteristics of all these tools are compiled on Figure 4, where the implemented features are represented by "+" and the features not implemented by "—". Figure 4 provides an overview of the main features existing in each tool.

	CBR Software Tools Characteristics					
CBRs/Features	Type	Similarity Strategy	Database Platforms	Cross Platform	Free Text Handling	Adaptability
ART*Enterprise	User Application	Unknown	Several	+	-	-
CasePower	User Application	Nearest Neighbour	Unknown	+	-	-
CBR Express	User Application / API	Nearest Neighbour	Several	+	+	-
Easy Reasoner	API	Unknown	Unknown	+	+	-
ESTEEM	User Application / API	Nearest Neighbour / Induction	Several	-	-	+
jCaBaRe	API	To be developed	To be developed	+	To be developed	+
KATE	User Application	Nearest Neighbour / Induction	Several	-	-	-
ReMind	User Application / API	Nearest Neighbour / Induction	Unknown	+	-	+

Figure 4 - Characteristics of existent CBR Software Tools.

2.1.3 Genetic and Evolutionary Algorithms

Genetic and evolutionary algorithms are computational models based on the concept of inheritance and evolution, such as the evolution of species. This algorithms use concepts of evolution and selection, mutation and reproduction operators. All those processes are dependent on the population, in a particular environment.

These algorithms are probabilistic and seek for an optimal local solution. First is established a population, where each individual represents a quality to the solution, and then the population will suffer several recombinations and mutations.

On the process of solving a problem, first a random population is established and the elements closer to the solution are chosen, meanwhile, crosses between individuals are made to generate new points of development.

Finally the mutation is performed and new elements are created through random changes. With this new population, the previous steps are repeated until you reach the optimal solution.

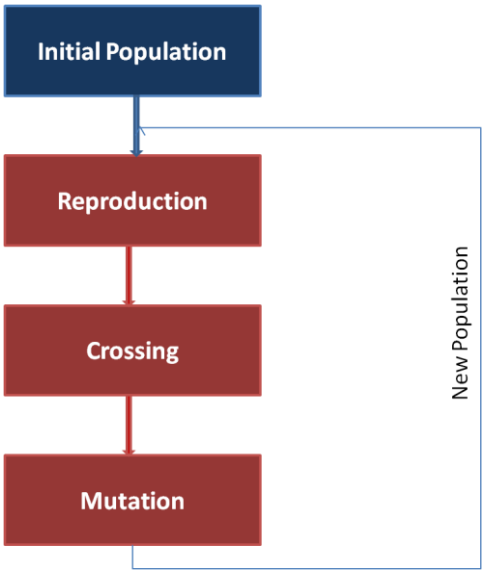


Figure 5 - GEA process of getting a solution to a problem.

Genetic and evolutionary algorithms are perfect for optimization problems, problems where the solution is difficult to find and when finding the solution requires parallelism (Analide C. et al., 2012; Back T., 1996; Linden R., 2006).

The GEA Systems have many applications such as Computer-Aided molecular design (Clark D. and Westhead D., 1996), molecular recognition (Willet P., 1995), drug design (Parrill A., 1996), protein structure prediction and chemistry (Pedersen J. and Moult J., 1996).

In molecular design there is an important activity related to the creation of new molecules, with designed qualities, in this area EA (Evolutionary Algorithms) proved to be an alternative. Venkatasubramanian et al. developed an algorithm where the molecules are converted to strings using a symbolic representation and then they are optimized to have the

desired properties. Glen and Payne invented also a GA (Genetic Algorithm) that allows the creation of molecules but within constraints.

Another application is the handling of the chemical structure, in this case the GA was used to determine the minimum chemical distance between two structures. The GA solved this problem by mapping one structure into another.

In quantum mechanics the EAs have been used to obtain approximate solutions to the Schrödinger equation. Zeiri et al, use an EA to calculate the bound states in a double potential and in the non-linear density function calculation. To achieve that, they use real-value encoding.

Related to Macromolecular structure prediction, protein tertiary structure is one of the most common challenges in computational chemistry because the mechanisms of folding are understood and the conformational space is enormous. EA helped on the problem of the search space, however the main concern was to find a suitable fitness function, because that is what defines the objective of the system. Once more EAs proved to be a reliable solution.

Finally in the field of protein similarity, a GA for protein structure comparison was developed by May and Johnson.

This system is a binary string GA that encodes three translations and three rigid body rotations. During the encoding, a transformation is generated to overlap one protein on another. The result is given by the fitness function that uses dynamic programming to determine the equivalent.

In general, GEA is used to get optimal results in cases where the domain is too vast or there is a need to merge several techniques, because the solution is to difficult to achieve. It is also used when the parallelism is needed to get a solution.

2.2. Personalization over search engines

In the past few years the search engines have been evolving. Some years ago every search performed returns the exactly same result to any user, however nowadays that doesn't happen anymore. The results didn't become completely different they become adapted, with some commonality.

According to the search, Engine Journal of article "Dealing with personalized search [Infographic]", Google searches are shaped by four main factors, which are the location, the

language, the search history and the personal user settings. These are the main concerns of Google to try to get the results closer to the users.

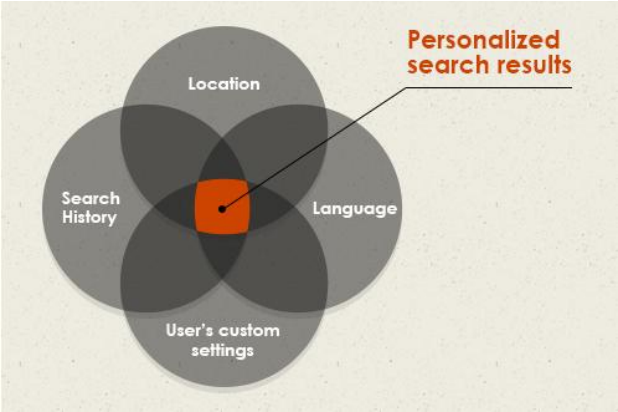


Figure 6 - Four main factors that personalize searches on Google.

However there other factors for instance, regarding the location, some systems split the location in many levels such as country, city or metropolitan areas. In this case Google, when receive a search query about a restaurant the results are ordered by distance, letting the user know where they can find the nearest restaurant of that kind.

Some systems also differentiate the searches performed on web from the searches from mobile. Regarding to the Personal History, Bing started to adapt their results not only by previous searches but also by likes or shares on social network, such as Facebook or Twitter.

These two also helped on personalization of the results providing information about social connections, where the results get influenced by user's friends (Search Engine Land, 2013).

In 2004 Sugiyama, Hatano and Yoshikawa proposed a solution that tried to created a profile of the user without any effort from the user. They evaluate existing systems, which require the user to fulfill questionnaires or to rate the results provided, and decided to create a solution able to retrieve and rank the results without asking anything to him. This solution provides results to the user, and monitors the history based on clicks of the user, adapting the user profile on the fly. This way the next query performed on the system will take into account the current profile of the user. The results collected from the analysis of the system revealed that this approach is reliable to the personalization issue on search engines (Sugiyama K., et al., 2004).

Danny Sullivan also introduces three further factors to the Personalization. In first place he explains that in location it is possible to specify the city or metropolitan area, as a way to suggest results. Second factor lies on fact of searches done by desktops present different results

from mobile. Finally, he talks about the social connections, since he considers the content viewed by friends a way to customize the search on a search engines (Sullivan D., 2011).

Other example of Personalization usage is Yandex, the Russian search engine, where a personalized search is considered useful only if the results are related to a context and user interests collected through previous searches.

An example is Figure 7:

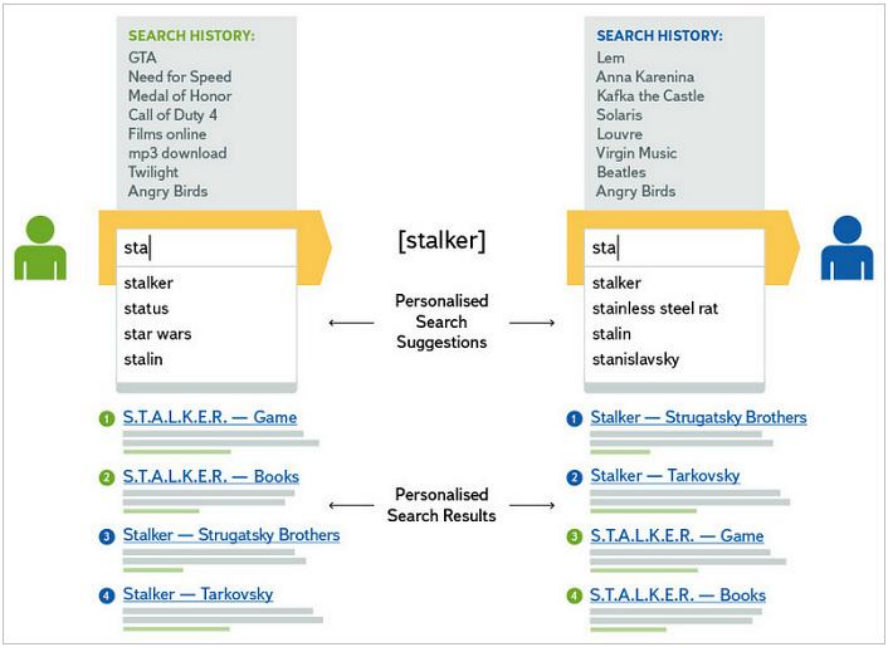


Figure 7 - Example of Yandex suggestions and results.

Figure 7 illustrates the fact that, if a user searches a lot about games, on his search he will see hints of games first, but if instead the user is interested in culture, then the suggestions will be cultural. However the formula used to calculate the relevance of the document to the search does not allow the use of factors such as personal information. The information used by the search engine to personalize search only takes into account the history and the clicks on results (Yandex about page, 2014).

In Personalization, Bing gave the first steps recently and opted to use a strategy based on two scenarios, which are the location and history. Besides that, they are trying to develop a new feature called adaptive search. This search consists on contextualizing the results, i.e., trying to understand what the user is looking for.

As an example, we have the “Australia” query. In this case you need to understand if the user is looking for information of a possible trip for holidays, or else if it is a movie buff interested on "Australia" movie (Bing Blogs, 2011; Sullivan D., 2011).

Focusing on the subject oriented to the concept of personalized images search, this is something new and with short information. With technological advances it becomes possible for the user to access an endless amount of pictures. However, there are some cons, because it is harder to find what we really want.

One possible approach is related to the image interpretation using its elements or colors, after that the image is categorized. Usually this concept becomes associated with systems that evolve through the similarities that are being found (Bissol S., et al., 2004).

Another approach is to contextualize the images using tags. This concept arises normally associated to social network or photo sharing sites and consists in assigning tags to images on publication. This will group the images into categories, allowing them to be available on other searches.

This method is quite popular since it only requires the system to collect the tags, building your base of knowledge (Borkar R., et al. ; Lerman K. and Plangprasopchok A., 2010).

2.2.1 Benefits

One of the main benefits of personalized search is related to the improvement of the quality of decisions made by consumers. According to Diehl, the internet reduced the cost of obtaining information however the capacity of dealing with that amount of information didn't evolve. This fact reduced the quality of consumers decision and provided a new path to be explored, developing new tools to help the consumer on decision process. The analysis of personalized results when taking decisions proved a positive correlation between personalization and the quality of the decisions. In the previous study it was concluded, that lower costs on search lead to no personalization, making consumers choose, options of lower quality (Diehl K., 2003).

Another study conducted by Haubl G. and Benedict G.C. also revealed that personalized searches reduces the quantity of options inspected (Benedict G. and Haubl G, 2012).

2.2.2 Disadvantages

A disadvantage that is always present when talking about personalization is the fact that personalization limits the user experience over the internet. According to Eli Pariser personalization lacks user possibility to meet new information (Pariser E., 2011). In 2002, Thomas W. Simpson said that "objectivity matters little when you know what you are looking for, but its lack, is problematic when you do not" (Simpson T., 2012).

Another disadvantage is the circulation of private information between companies. Nowadays, many systems gather the information about the users without their consent or knowledge, and that information is valuable for internet companies creating a privacy issue that doesn't benefit personalization.

Related to this work some researches were made and some achievements were reached in areas like elderly healthcare, where there has been some issues related to privacy and data protection (Costa A., et al., 2013).

The problem is connected to Ambient Assisted Living where the patients and doctors could be helped by technology. However the doctors have a deontological code to follow and respect, something that have to be kept by the ones who manage the system.

However there's no legislation on the computer scientific area that protects the information managed by the systems that are developed.

Something that also happens is that our information we share on internet or even information that is collected by our actions are gathered by companies that gets a huge profit when handling it.

3. Technologies

As software product, the solution of this project is trying to provide some technologies, each one trying to solve the problems that were described before. In this chapter it will be explained the technologies that were closer and the explanation of why it was chosen.

3.1. Programming Language

The programming language chosen to be used within this project was Java. This language was first thought to be used when a mobile application became part of a solution. This fact reduced largely the wide range of programming languages available to develop this solution. However this only narrows the development of the mobile application itself, the choice of the programming language for the web server were not affected however for consistency it was maintained the same on server side.

That was, one of the reasons although there were others, for example Java is a versatile and modular programming language which is good it is possible to develop some modules that can be adopted without changing all the structure of the solution. This fact helped to create a generic solution before adapt it to this particular case. Giving the possibility to cover several case studies James Gosling also developed java to class-based and object oriented being released in 1995 (Gosling J., McGilton H., 1996).

The ability of java to deal with concurrency, as it was said before CBR deals with huge amount of information which can cause a delay retrieving cases, with concurrency this problem can be circumvented (Goetz B., Peierls T., 2006).

Other fact was the highly usage by other users which provided some background with some frameworks that were needed and helped with the portability of the system.

Finally java is well-funded and supported which makes it a choice to long-term since it's not expected that with so many followers and so many frameworks it will disappear soon.

Although all these advantages it is clear that it's not perfect there are some problems such as memory consumption or performance (Jelovic D., 2012). However they were not a sufficient to change the decision of using Java.

3.2. Android

The platform used to develop the mobile application was Android. The choice was made due to the amount of users that everyday join this platform, over 1.5 million every day. The programming language that is based was also a factor to take into account, without forgetting the easy way to test and distribute.

When speaking about the mobile platform to achieve the results pretending, there were two possibilities IOS and Android. These two are the most common and more supported due to the amount of users that everyday requests new features (NetMarketShare, 2014). These origins the creation of new frameworks, everyday helps to achieve some base features, such as http request or image loading features bases on URL searches.

However between these two, there's one that stands out. That's Android.

Developed by one of the most reliable trademarks on market, Google inc., it was released as an open source project since 2008 and then a lot of users started to develop many other frameworks to interact with Android.

This open source philosophy helped the exponential growth that occurred in the last years but also became the system more robust and fault tolerant.

In sum, this framework is a reliable choice because it is used by many users which potentiate its growth and improvement.

Furthermore as an open source operating system, the cost of development of an app is much lower when compared with iOS.

Finally Android is based in Java which provides a huge adaptation and modularity to the software developed to run its devices.

This modularity enables our solution to be part of a single solution or for instance become a module within other softwares that look at this approach as a potential help to bigger problems.

3.3. Case-Based Reasoning - Decision

Analyzed those three techniques (ANN, CBR and GEA), it was possible to realize a bit in which consisted and what is their most common uses. Established the scope of this project as something that need to evolve, but without the need of parallelism or an optimal solution, it was clear from the beggining that genetic and evolutionary algorithms would not be the path to follow.

In addition, the use of previous cases is a factor that is intended to be exploited. This field suited well either in ANN, through the cases used on network training, either the RBC, with the experience through previous cases (Analide C., et al., 2012; Aamodt A. and Plaza E., 1994).

Since it is intended that the user searches are recorded, to be used later as a suggestion or filter on searches carried out, it becomes evident the need to go consuming a knowledge base, built on previous experience of the user and not so much in need to "teach" a structure to respond more effectively to the problem (Kolodsen J., 1992). Both paths would be viable, however the choice was made for the CBR technique, because it is intended that, during use, each user can check its knowledge base so it can "see" if some of the previous cases are closer to any search already done.

This is a case where the CBR can be used effectively to suggest, through the similarity of cases. Using this technique it also becomes easier to share the cases between users. Using ANN the system would have to add cases to train the network, and only then have improvements on search results (Analide C., et al, 2012).

Finally this project can contain exceptions to the rule, an example is the Australia query, explained before. Many users can run this query but they have not the same intent.

3.4. Bing API

This project has an objective related to the creation of an image search engine, so there is a need to work over an already developed framework that can provide results about a query. The creation of a system from scratch was not an option once there are some search engines already working, and has more than satisfying results. These results had to be reliable in order to provide to the system, sets of images that are connected to the subject searched. Analysing those factors and looking for a solution with huge acceptance and simplicity, Bing was the best choice (Rao L., 2012).

Bing is a search engine such as Google or Yahoo, from Microsoft, released in 2009. This search engine is growing since the beginning gaining lots of users and becoming the second choice of the search engines, only being overcome by Google. More than the fact of being used by lots of users is the fact that Bing is present in more than 30 languages which helps on personalization regarding the location of the users (Rao L., 2012).

As it was said before Bing is improving its search system trying to integrate personalization on the results, however they features started by the most common like location, search history, social connections and so on. This fact along with the possibility to filter the results only to images also helped to decide for the usage of Bing API.

Bing API is very simple once it is only necessary to make an http request to a specific address, and than convert the result to the objective we want. In this project for instance the results are converted on a set of urls to the images that are provided to an Image loader. That image loader uses the urls to get the images placing them on a view of the Android application. But there is also another feature given by Bing API that led to its choice, the feature "Related Search". This feature suggests some queries related to the issue searched that could be interesting to the user. Those related searches, are one of the steps taken into account to the personalization process of our solution once this project did not reject the techniques already used, this project is trying to evolve the personalization analyzing some new paths that were not explored yet.

Regarding the results, they are easily converted once they are returned in XML or JSON, which are notations commonly used. The usage of those notations helps on getting the information needed from images since there are lots of libraries that handle with notations, providing results in good time. The time spent on getting the results is important since one of the objectives of the project is the retrieval of results in good time.

Finally regarding the results they are used twice on this, they are used without any filter to give the user the possibility to face new information not focusing its knowledge only in things from its interest. And they are also used on personalization process being presented to the system that will determine some related content to be suggested to the user based on the query and results.

4. Implementation

In this chapter will be explained, the most relevant decisions made during the development of this solution and it will also be explained with more detail the architecture of the solution and its components. This chapter is also important to give a general overview about the workflow of all process and components such as the Android application, the CBR system of the server and the communication between them.

The Figure 8 provides an overview about this solution in regards three-tier architecture:

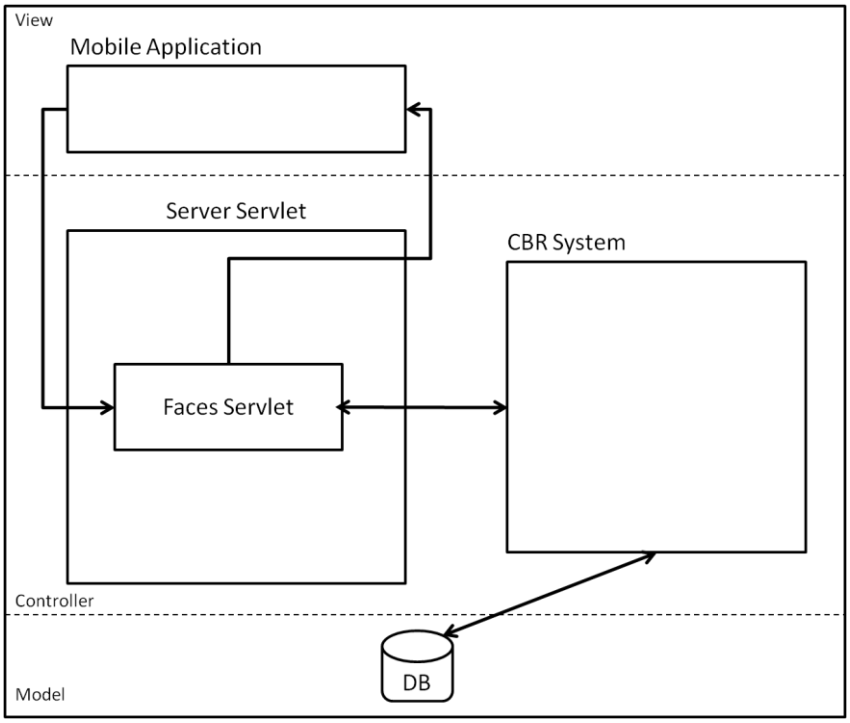


Figure 8 - Architecture of the Solution.

This architecture showed on Figure 8 clearly defines three layers as it was said before. Starting from the bottom, the Model layer is where the data base is located, this layer is responsible for providing and storing the information that will be manipulated by the system.

The middle layer named Controller is where the information given by the user is used by the system to find appropriate solution, based on the system itself and based on the information from the model layer. Basically this layer is the intermediary between the other two. In this system the user needs some information from the database (Model layer) however that

information cannot be given directly, there is a need for some computation before the information retrieval. This computation is performed on the controller layer.

Finally on the top there is the presentation layer, here is where the user of the platform can request information and where after the computation the information is shown.

With the objective of providing an overall perspective of the workflow of our solution, it was created a sequence diagram (Figure 9) that show how the user interacts with the system performing a query and how the servlet handles the information received, with the purpose of converting it into something the CBR system understands.

It is also explained how the CBR system communicates with the database and finally it is represented all the way back into the result presented to the user.

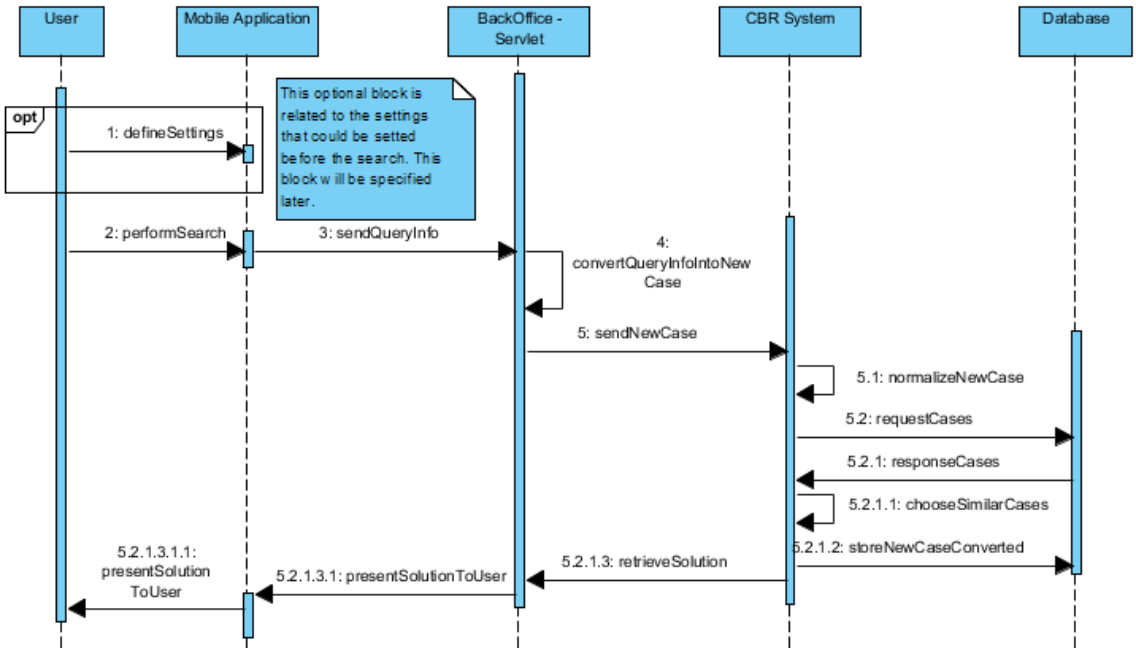


Figure 9 - Sequence Diagram of the Solution

4.1. BackOffice

The back office of this solution is a main component, because it represents all the logic behind the results that are presented to the user. These components contains the CBR System, which is responsible for receiving a case, convert it into the new structure, collect similar cases, choose a solution, revise the solution and finally retain the new case. In this CBR System it was introduced

two new techniques one for handling default data and another one for string comparison that will be explained better on the next chapters.

Still on the server there is a servlet responsible for the communication between the user request (Presentation Layer) and the CBR System.

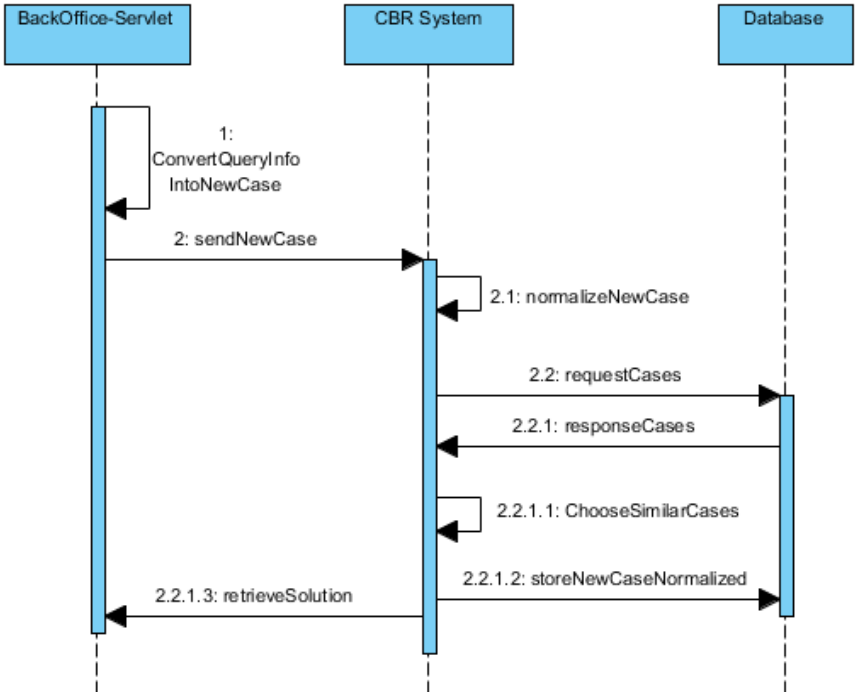


Figure 10 - Sequence Diagram of the Server.

4.1.1 String Comparison

One of the big concerns about the problem presented on this work was the string comparison, because the problem itself is related to the query the user wants to be answered, the images are only the way to present the results. The string comparison is what will be used together with the handling default data technique to achieve the results closer to user's will. Once more it is important to clarify that no processing is applied to the image with the intention to enhance the search of similar results.

Many other systems deal with string fields however there's almost always a guide to be followed that narrows the search, for instance the fields have some possibilities to be chosen by the user. This is a good practice to enhance the speed of case retrieval and the similarity between the results. However not only the search on the knowledge base is narrowed, the input

and the possibility to enhance the query are also narrowed and that limits user experience within a system. Besides that, there are some cases where the free text input fields are needed. For example a search engine, we want to write what we are looking for and it is not possible to have search filters that match all the user needs. On those cases, string comparison become part of the solution with some metrics.

The metrics used within this project were Jaro-Winkler distance, Dice's coefficient and Levenshtein distance.

Jaro-Winkler

As it was said before the Jaro-Winkler distance is the measure that results from the comparison of two strings. It was extended from the Jaro Distance and its usage was planned for the detection of duplications. In this metric the two strings are more similar as higher as the distance between them. It is more suitable for short strings and its similarity is between 0 and 1, where 0 represents no similarity at all and 1 represents a perfect match.

To calculate the similarity there a formula expressed on two concepts, the number of matching characters (m) and half of the number of transpositions (t). This formula represents something simple, if the number of the matching characters between the two strings (s1 and s2) is 0, so the distance is also 0, otherwise:

$$d_j = \begin{cases} 0 & , \quad m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right), & otherwise \end{cases} \tag{1}$$

The transpositions are taken into account since the order of the matching characters, are important to the matching measures.

With this process every character of first string (s1) are compared with every matching characters of second string(s2).

Levenshtein

Considered as valid in 1965 by Vladimir Levenshtein this measure is different from the previous. That's because Levenshtein distance calculates the differences between two words. This distance is measured looking into any insertion, deletion or substitution between first string (s1) and second string (s2).

It is recommended for situations where is intended to find short matches in long texts or situations where only a few differences are expected, such as spell checkers for example. When used for two long strings, the cost of calculating the distance is very high, almost the product of both string lengths.

Compared with Jaro-Winkler this distance respects the triangle inequality, this is the sum the Levenshtein distance between s1 and s2 is equal or lower than the sum of Levenshtein distance between s1, s3 and s2, s3, just like on a triangle where the sum of two sides are always greater than or equal to the remaining side.

The formula to calculate the distance is:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{a_{i \neq b_j}} \end{cases} \end{cases} \quad (2)$$

The result will always be at least the difference between sizes and at most the length of the biggest string.

Dice's Coefficient

Dice's Coefficient is a measure of similarity between two sets. This similarity is based in the number of common bigrams which are pairs of adjacent characters. This metric was developed by Thorvald Sorensen and Lee Raymond Dice and can be applied in several areas with different needs. For instance it can be used for sets, vector or this case for string.

The formula to calculate Dice's Coefficient is :

$$s = \frac{2n_t}{n_x + n_y} \quad (3)$$

4.1.2 Handling Default Data

The second concern about this project is related to the fact of handling default data. This project is intended to deal with different kinds of information or even deal with missing or incomplete information. But the objective is create a solution, to retrieve a valid solution or those cases too.

To become this solution reliable it was applied two new concepts, the DoC and the QoI. The DoC is the confidence applied to each term of extension of a predicate which helps on the normalization of the cases and on the improvement of speed of cases retrieval.

This process of normalization based on the calculation of DoC can be expressed on a new CBR cycle.

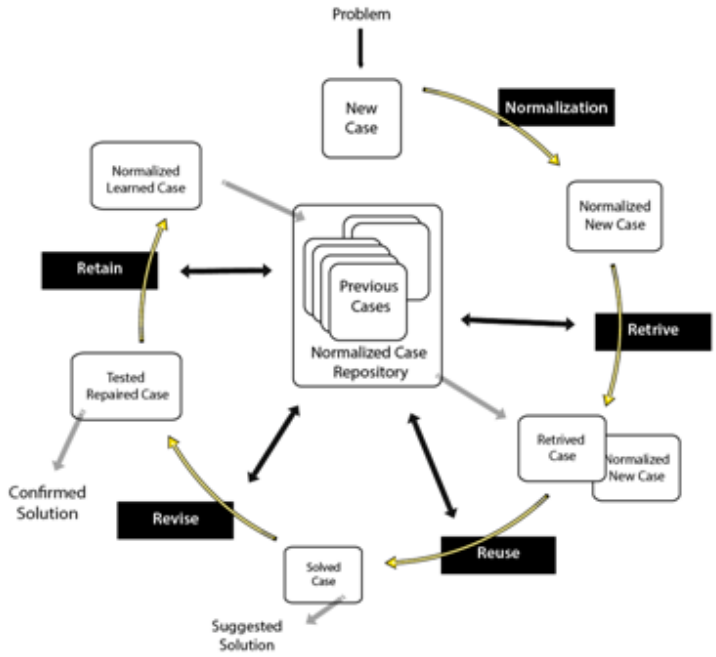


Figure 11 - Adapted CBR cycle taking into consideration the normalization phase.

This new cycle introduced a new phase called normalization where the new case introduced to the system is converted into the new structure to enhance the quality of the case retrieval. This conversion improves the quality of the similarity measures and makes all the system operate with normalized cases making the retain phase a process that add an already normalized case to the knowledge base. The DoC is also used to calculate the similarity between cases. This approach was previously used on real cases with good results, it was applied on

schizophrenia diagnosis (Cardoso L., et al., 2013) and asphalt pavement modeling (Neves J., 2012).

To fulfill this implementation some changes need to be performed related to the knowledge representation and reasoning.

Those changes use concepts of Logic Programming (LP) paradigm that were previously used for other purposes such as Model Theory (Kakas A., et al., 1998; Gelfond M. and Lifschitz V., 1988; Pereira L. and Anh H., 2009) and Proof Theory (Neves J., 1984; Neves J., et al., 2007).

In 1984 J. Neves created a new way to present knowledge representation and reasoning, which was used in this project. This new new representation is ELP (Extended Logic Program) which is a proof theoretical approach using an extension to an LP language. This extension is composed by a set of clauses in the form.

$$p \leftarrow p_1, \cdots, p_n, not\ q_1, \cdots, not\ q_m \tag{4}$$

$$?\,(p_1, \cdots, p_n, not\ q_1, \cdots, not\ q_m)\ \ (n, m \geq 0) \tag{5}$$

In the previous formulas ? is a domain atom representing falsity, p_i, g_j and p are classic literal, can be a positive atom or a negative atom, if preceded of negation sign $\neg$. Following this kind of representation a program is also associated to a set of abducibles (Kakas A., et al., 1998; Pereira L. and Anh H., 2009). Those abducibles are exceptions to the extensions of predicates.

Finally the purpose of this representation is to improve the efficiency of search mechanisms helping on the solution of several problems.

The decision making processes still growing and some studies were made related to the qualitative models and qualitative reasoning in Database theory and in Artificial Intelligence research (Halpern J., 2005; Kovalerchuck B. and Resconi G., 2010).

However specifically related to the knowledge representation and reasoning in LP, there were some other works that presented some promising results (Lucas P., 2003; Machado J., et al., 2010).

On those works was presented the QoI(Quality of Information) measure, that were related to the extension of predicate i and gives a truth-value between 0 and 1. Where 1 represents

known information (positive) or false information (negative). When the information is unknown, the QoI can be measure by:

$$QoI_i = \lim_{N \to \infty} \frac{1}{N} = 0 \qquad (N \gg 0) \tag{6}$$

in situations where there is not set of values to be taken by the extension of predicates. In this cases N denotes the cardinality of the set of terms or clauses of extension of predicate.

When there's a set of values to be taken the QOI will consider the set of abducibles and use its cardinality. If the set of abducibles is disjoint the QOI is given by:

$$QoI_i = {}^{1}/_{Card} \tag{7}$$

otherwise it will be handled a Card-combination subset with Card elements (C_{Card}^{Card}):

$$QoI_i = \frac{1}{C_1^{Card} + \cdots + C_{Card}^{Card}} \tag{8}$$

Another element taken into account is the importance given each attribute of the predicate. It is assumed that the weights of all attributes are normalized

$$\sum_{1 \le k \le n} w_i^k = 1, \forall_i$$
$$\forall \rightarrow universal\ quantifier \tag{9}$$

with all the previous elements it is now possible to calculate a score for each predicate given by:

$$V_i(x) = \sum_{1 \le k \le n} w_i^k \times QoI_i\,(x)/n \tag{10}$$

this way it is possible to create a universe of discourse based on the logic programs with the information about the problem. The result are productions of the type:

$$predicate_i(x_1, \cdots, x_n) :: QoI \tag{11}$$

The DoC is given by $DoC = V_i(x_i, \ldots, x_n)/n$ representing the confidence in a particular term of the extension of a predicate.

Summarizing the Universe of discourse can be represented by the extension of the predicates:

$$a_1(\cdots), a_2(\cdots), \cdots, a_n(\cdots) \ \ where \ \ \ (n \geq 0) \tag{12}$$

but this new concept does not simply changes the structure of the cases, it also changes the way of calculation of the similarity between them.

So analyzing each case there's an argument for each attribute of the case.

This argument can have several types of values, it can be unknown, member of a set, may be in the scope of an interval or can qualify an observation. The arguments compose a case, that is mapped from a clause.

For instance we can consider a clause where the first argument value fits within the interval [35,65] which the interval [0,100] as a domain. The second argument is an unknown value (¬) and its domain ranges the interval [0,20]. And finally the third argument is the certain value 1 within the interval [0,4].

Considering the case data as the predicate extension:

$$f_1 : x_1, x_2, x_3 \rightarrow \{True, False\} \tag{13}$$

One may have the following representation:

$$\begin{cases} \neg f_1(x_1, x_2, x_3) \longleftarrow not\ f_1(x_1, x_2, x_3) \\ f_1(\underbrace{[35, 65], \quad \perp, \quad 1}_{attribute`s\ values\ for\ x_1, x_2, x_3}) :: 0.85 \\ \underbrace{[0, 100][0, 20][0, 4]}_{attribute`s\ domains\ for\ x_1, x_2, x_3} \end{cases} \tag{14}$$

The following step after setting the clauses or terms of the extension of the predicates is the transformation of the arguments into continuous intervals, using its correspondent domains. The first argument is already an interval, so this step is only applied for the following arguments. The second argument has an unknown value so it's correspondent interval matches it's domain. The third argument is a certain value so its interval is within its value ([1,1]).

Finally the 0.85 value represents the QoI of the term $f_1([35,65],\ \perp, 1) :: 0.85$. This step able the determination of the attributes values ranges for x_1, x_2, x_3 as explained next:

$$\begin{cases} \neg f_1(x_1, x_2, x_3) \longleftarrow not\ f_1(x_1, x_2, x_3) \\ f_1(\underbrace{[35,65],[0,20],[1,1]}_{attribute`s\ values\ ranges for\ x_1,x_2,x_3}) :: 0.85 \\ \underbrace{[0,100]\ [0,20]\ [0,4]}_{attribute`s\ domains\ for\ x_1,x_2,x_3} \end{cases} \tag{15}$$

At this step it is possible to calculate the Degree of Confidence of each attribute. For instance the DoC of the first argument ([35,65]) will demote the confidence that the attribute fits the interval [35,65]. But first a normalization where every domain have to fit a common interval [0,1].

The normalization procedure is according to the normalization $\frac{Y - Y_{min}}{Y_{max} - Y_{min}}$

$$\begin{cases} \neg f_1(x_1, x_2, x_3) \longleftarrow not\ f_1(x_1, x_2, x_3) \\ x_1 = \left[\frac{35-0}{100-0}, \frac{65-0}{100-0}\right], \\ x_2 = \left[\frac{0-0}{20-0}, \frac{20-0}{20-0}\right],\ \ x_3 = \left[\frac{1-0}{4-0}, \frac{1-0}{4-0}\right] \\ f_1(\underbrace{[0.35,0.65],[0,1],[0.25,0.25]}_{attribute`s\ values\ ranges\ for\ x_1,x_2,x_3}) :: 0.85 \\ \underbrace{[0,1]\quad [0,1]\quad [0,1]}_{attribute`s\ domains\ for\ x_1,x_2,x_3} \end{cases} \tag{16}$$

Once the domains are normalized the equation $DoC = \sqrt{1 - \Delta l^2}$ is applied to each attribute, where Δl is the range of normalized interval.

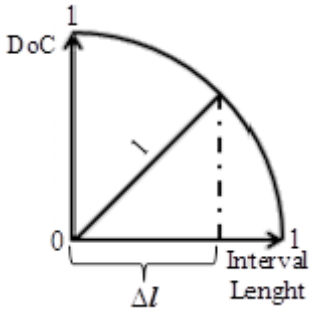


Figure 12 - Evaluation of the Degree of Confidence.

Below, it is shown the representation of the extensions of predicates within the universe of discourse. At this stage all the arguments are nominal and represent the confidence that attribute fit inside the intervals referred previously.

$$\begin{cases} \neg f_1(x_1, x_2, x_3) \longleftarrow not\ f_1(x_1, x_2, x_3) \\ f_{1_{doc}} \underbrace{(0.95, \quad 0, \quad 1)}_{attribute`s\ confidence\ values\ for\ x_1,x_2,x_3} :: 0.85 \\ \underbrace{[0.35, 0.65][0, 1]\ [0.25, 0.25]}_{attribute`s\ values\ ranges\ for\ x_1,x_2,x_3} \\ \underbrace{[0, 1] \quad [0, 1] \quad [0, 1]}_{attribute`s\ domains\ for\ x_1,x_2,x_3} \end{cases} \tag{17}$$

With this representation it is now possible to represent all cases of the knowledge based in a graphical form with Cartesian plane in terms of QoI and DoC of each case.

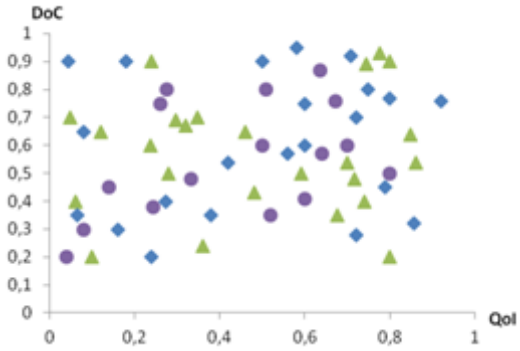


Figure 13 - Case base represented in the Cartesian plane.

To select, the best solution of new case is represented in the graphic according to its DoC and QoI and then it is selected a suitable duster which is measured according to a similarity measure given in terms of the modulus represented below. In the following figure it is represented the duster (red circle) and the new case (red square).

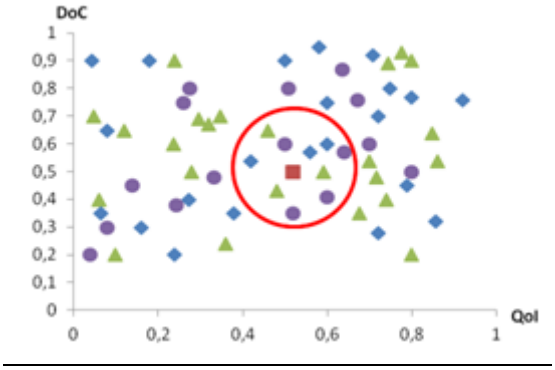


Figure 14 - Case base represented in the Cartesian plane.

4.2. Data Model

Starting from the lower level of this solution, we will explain the data model. The data model represents the structure of the database that will be used by the CBR system. As it was explained on chapter 2 the CBR system has a repository where the cases are stored. This repository is also used by the CBR system to find the best solution to a new case. So the database should follow a structure that enables the system to retrieve the result that is intended to be on this project and, at the same time follow some guidelines from the CBR system. For that reason the data model was one of the components that were more discussed, because it has to respect some requisites and it has to be adapted, since our system is supposed to grow and new features could appear.

One of the requisites was to respect the structure of the cases of CBR systems that was normally used. When studying the concept of the CBR, the structure that was mostly used to explain the case structure, was a structure where is defined the problem, the solution to the problem and a justification to the solution presented (Analide C., et al., 2012).

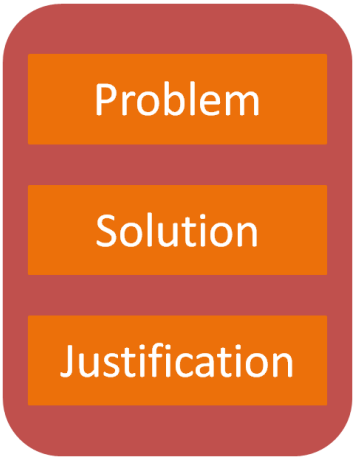


Figure 15 - Structure of the CBR Cases.

So with this structure in mind and with the features and requisites of our initial problem, the solution that were thought, needed to collect the information about the query to be searched, which represents the problem, needed to collect the information of the URLs that represent the solution and finally collect some tags that can be seen as a justification. However in this case the tags are not intended to justify the solution, they represent additional information about the query that could be used to find a better match to a new problem presented to the CBR system.

But this structure didn't have some of the new features that appeared next such as the age group and the location, so the new structure was the following:

Figure 16 - Structure of the CBR Cases Adapted.

This new structure is the base of our solution, however it suffered a little adaptation because of the usage of techniques for handling default data. The handling of default data as it was explained before gives this system the capability to deal with uncertain and unknown information, so in order to fulfill the requisites explained on the previous chapter all the cases have to be normalized according to those concepts. In addition any new case presented to the system has to be normalized first. This normalization would involve the first two attributes of the case since the other three are free input text fields, something that the handling default data technique doesn't know how to proceed.

So the chosen database structure that respect those requisites described above was the following:

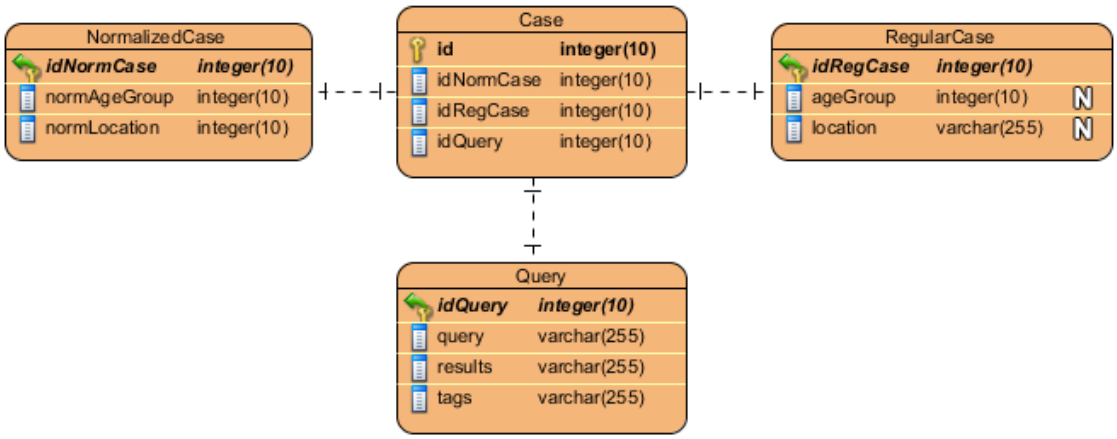


Figure 17 - Data Model of the Solution.

In this model it is possible to see that the case itself is structured around three tables, the main table is the "Case" which is composed by an id of the case, an id of the normalized fields (idNormCase), an id of the regular field (idRegCase) and an id of the query (idQuery). The table "Regular Case" contains the field "idRegCase" to match with the case that represents, contains the Age Group (ageGroup) and Location (location) provided by the search. The table "Normalized Case" will contains the same fields that the "Regular Case", however how it's name suggests they are normalized. This normalization converts all the regular fields to integers, even if the originals are not integers so this table will contain the idNormCase field to connect to the case table and then contains the two fields normalized from the "Regular Case" table, the fields "normAgeGroup" and "normLocation" respectively. Finally the table "Query" stores the “idQuery” to be related to case table and also stores the query itself, the results and the tags.

We believe that this model, according to the features considered at this stage, represents a reliable and fast way to store all the information involved in this system. We also believe that this approach is structured according to the principles explained on the handling default data chapter.

4.3. Mobile Application

The mobile application, as it was explained before, represents the presentation layer of the architecture of this project. It was developed under the platform of Android due to the fact that nowadays most of the mobile users have an Android and it is easier to develop a fully functional application. However this application can be developed in other platforms, such as iOS or Windows Phone in regards to the mobile development. This application can even be adapted to be a web application, giving to the user the possibility to perform their searches on a web browser. To achieve one of these alternative solutions and many others that could appear in the future, the only requisites needed to be filled by the new platform are the connection to internet, the possibility to handle http requests which is related to the connection with the internet, and the capability to provide to the user the images returned from the system.

Once again it is important to understand that the images are only the result of the queries, and no further analysis, is performed with them to achieve a better match of the results. This means that this project can be easily adapted to a normal search engine.

It is also important to understand, that this application is where each user, can interact with the system performing their searches, and achieving the results that the system select as the most relevant to the current will of the user.

The first step performed after completing the CBR system and the Servlet was the development of some screens of the application and the definition of the structure of the application, in order to fulfill the objectives and requisites of the system.

The structure that was chosen was created following the structure of other search engines, since it has already been performed complex studies to understand what is more user friendly and what could not be used. Those facts lead to an application with three screens, one to perform the search, another one to define some settings about the search and finally one to present the results. The last one was chosen not to be divided in two (normal results,

personalized results) because, it would be necessary one more click to achieve one of the results and Google, Yahoo! and Bing are only one click away from the results.

Related to the screens some mockups were created and some improvements were made until the reach of the results that will be presented next.

4.3.1 Search

The search screen is the first to be presented to the user and it was one of the main concerns about the screens of the mobile application. When analyzed other search engines, such as Google or Bing it was perceived that the main screens were simple, discreet and user friendly. Those screens were normally composed by, an input box and a search button that lead the user to the result with only one click. This strategy was followed strictly by this solution that maintain the idea of the input box with search button, however we wanted to add some options to help the user to achieve the results he want. Those options needed to be chosen before the search being performed, but they don't need to be changed in order to take advantage of the system. So it was decided to add another button on the top right corner where the user could access those options. This solution was the best way to not disrupt the simple layout, but at the same time allow the user to access some extra options.

The Figure 18 represents the result that was achieved with the specification above.

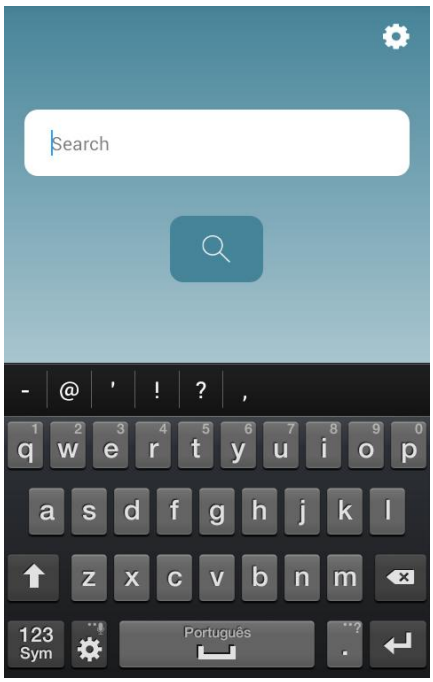


Figure 18 - Search Screen.

At this screen the user is able to perform the search he wants by typing the query on the input box and clicking the search button. As an option the user can define some extra features to access them he can click the settings button on the top right corner.

The diagram of Figure 19 exemplifies the actions (use cases) that can be performed by the user.

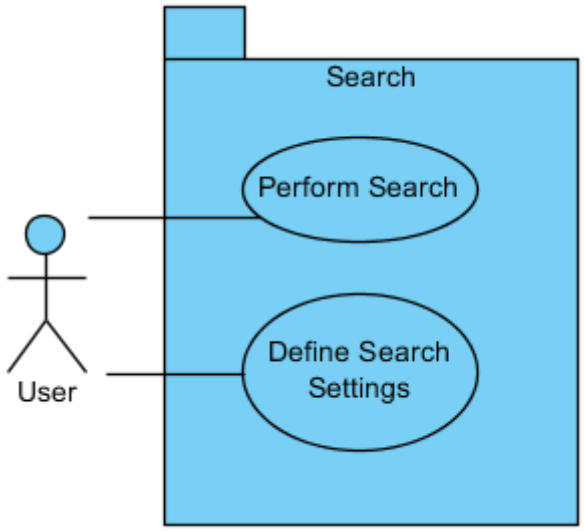


Figure 19 - Use Case Diagram of Search.

This diagram represents a simple way to explain which actions can be performed at this screen, however analyzing all the systems that compose this system it is possible to better detail the interaction that happens on this screen.

In the diagram of Figure 20 it is possible to understand some interactions that happen between the user, the mobile application and the servlet. There are more systems to evaluate, however in this diagram, they will not be explored.

Those interactions start by the option given to the user to define some extra features to the search, this features are used by the mobile application when the user selects a search to be performed. At that time some information is sent to the servlet, giving it the possibility to request the results to the CBR system. The results are then communicated backwards until they reach the mobile application again.

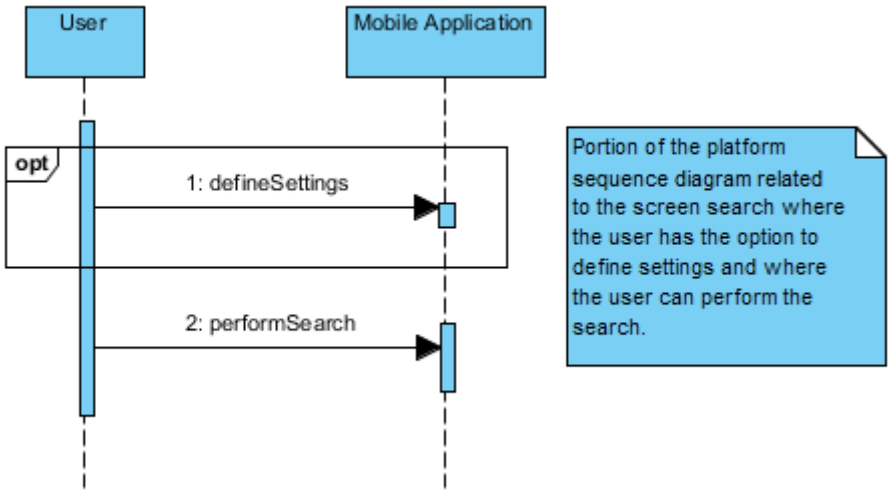


Figure 20 - Sequence Diagram of Search.

4.3.2 Settings

When other CBR systems were analyzed some flaws were detected, for instance besides all the adaptability of the systems, the user couldn't change some aspects on the fly, such as the weights of the cases attributes.

Together with the flaws, some new functionalities were thought, that lead to the creation of the screen.

On this screen, we try to give to the user the opportunity to customize the search process with the objective to get the information closer to his needs. Here the user has some options to change such as the age group, the location in terms of continent, the tags, the number of results and finally the weight of the case's attributes. Before explaining each one of them it is important to clarify that the tags, the age group and the location are not kept between searches. Those information are specific to each search and for that reason it would be bad for the system recommendations keep them between searches. In this project when the user performs the search the mobile application is responsible for cleaning this information. All the other features are saved until they are changed again. Those features are saved on shared preferences of Android. This Android feature is an abstraction of a database that saves and retrieves key-value pairs of primitive data types. This key-value pairs persist during the users sessions even if the application is killed, providing a reliable and fast way to stored. Another option it could be the creation of an internal database but there is no need to spend time on database transactions,

when the information is so short and it is not important to keep an history of the changes performed, since they are replaced.

Speaking about the features, the first option provided to the user, is the possibility to select the age group. This field is not to be filled specifically by the age of the user, it has to be filled with the age group that the user want to use to filter the results. For instance if we want to implement the parental control, an option would be block this field to "minor". The other options to this field is "grown up", "middle age" and "old man". In addition to the age group there's the location, in this project we decided to divide the location only in the five continents, which are Europe, Asia, America, Africa and Oceania. The purpose of this location is to try to filter the results by location, but instead of using an unknown algorithm from Bing it is used our new approach to the CBR system.

Tags are another feature present on settings screen, this field could be used by the user to add some information to the search. For instance, when searching "football", the word "sport" could be used as a tag. This tag is saved on the database and it is used by the CBR system to provide a better match with the cases on the repository, providing a context to the searches performed on the system.

The number of results is a simple filter that was added, and implemented on the mobile application after concluding the implementation of the Bing API. The Bing API is an interface that allows us to use search results using Bing system and one of the features detected was the possibility to select the number of results the system returns, and so we decided to add this feature also to the CBR system giving the user the possibility to select only the top ten for instance.

Finally on this screen it is possible to define the weights of the attributes of the case. This feature was added because as it was seen on chapter 2, the existent CBR system tried to be adaptable, but they don't allow the user to define the importance of each one of the attributes. In our particular case, we thought it would be important to explore this idea, because we wanted to provide an innovative solution, that fulfilled some gaps detected since we began this project. Furthermore when searching for results more adapted to us, we can consider getting recommendations based on the results we have, and don't give too much attention to the query itself. Other example could be related with the fact that if we are performing a search, we may want to give more importance to the information of the tag than the text searched.

Defined the features, we wanted to tried, to create a solution, that the user didn't need to learn how to use, a solution that the user reaches his goal in an intuitive way and the result was the screen on Figure 21:

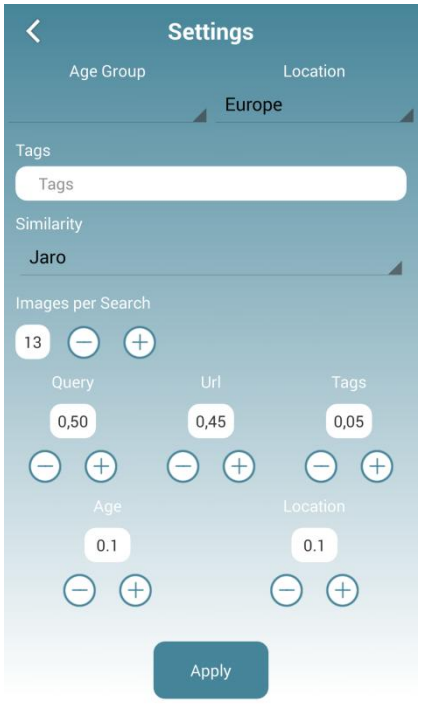


Figure 21 - Settings Screen.

To maintain some coherence the fields where the user had some options to choose (Age Group, Location, Similarity metric) it was used Android Spinners, this feature provides a drop down list with the options to select, something that was very common on web pages even before mobile applications exist. The tags, needed to be an input box, because only user know, the context to give before the search to the other fields, like the number of results and the weights, it was created a simple interface with a plus and minus button that changes the value or a fixed scale. At the bottom of the screen it was added an apply button which saves all this information, some of them permanently like it was said before, to be used on the following search. In the following diagram of Figure 22 it is possible to represent the definition of these features, in terms of actions that the user can perform.

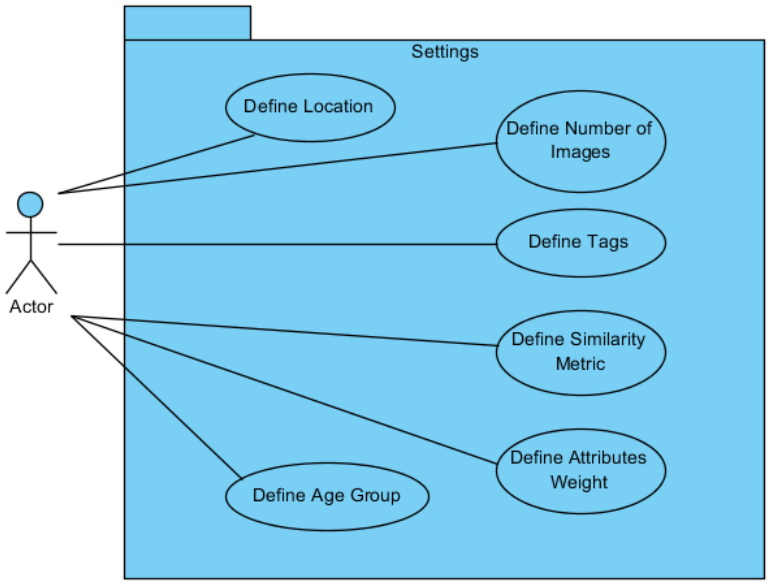


Figure 22 - Use Case Diagram of Settings.

As it could be seen on Figure 22, each feature was divided in a action that could be performed by the user. This happens because every feature presented on this screen have a default value, in some of them like tags, location or age group that value is empty giving to the user the possibility to only change the amount of features he wants. For example the user could enter this screen only to change the weights of the attributes letting all the other fields like they were defined last time. This idea explained here it is not supposed to be represented in the previous diagram and for that reason the following sequence diagram was created.

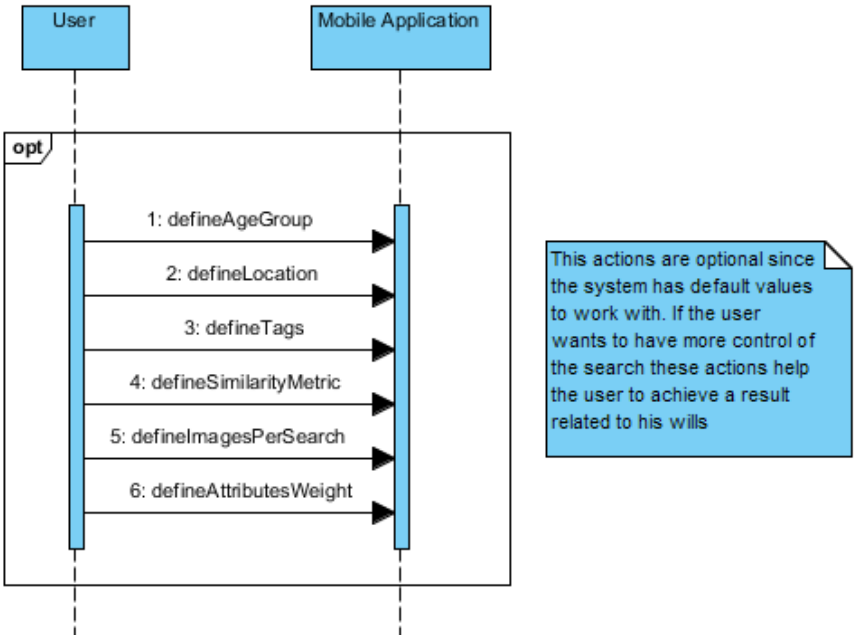


Figure 23 - Sequence Diagram of Settings.

This diagram of Figure 23 looks similar to the sequence diagram of the previous chapter, however in this case it is explored the optional block that already appeared, to explain that the user could, if he wants, set some additional features to the search.

4.3.3 Results

With the search selected and all the features defined, the next logic step is to perform the search to collect the results. This third and last screen of the mobile application represents that. This screen is responsible for providing to the user two types of results and navigation between them. Until we reach the final layout once more we look to the other system to find the best way to present the results to the user.

What we detected when analyzed other systems, was that they present the results in form of a grid, to give an overall perspective of the results and then only when we click the image, we see it on its real scale and we can access more options and more information. This idea was brought to our project and like in other systems we have a grid with the results. Related to the options, what we implemented was a double tap on the image to save to gallery on the smartphone.

However we still have other issues because we defined that our project will not block the users to find new knowledge. One of the main criticisms made to personalization is that the users cannot find new knowledge because the results are always connected to something he already search or likes. To solve this issue we decided that we will present two kinds of results the first one would be the result provided by Bing which is the result that the user gets when use this system and the second one would be the result provided by our implementation. This fact raised another concern, because with two different types of results there are some doubts about how to present the results to the user. Should the result be presented on a unique grid with a separator, should be presented on the same screen or should be another screen for the personalized results?

The result was the one presented on Figure 24.

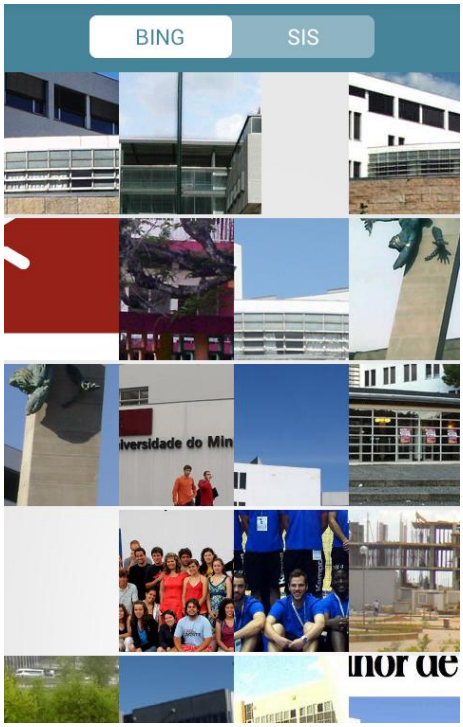


Figure 24 - Results Screen.

What was perceived by other search engines like Google, Yahoo! or Bing, but especially Bing was the usage of top separators. If we perform a search on Bing, in the top of screen there are some separators that let the user choose if we wants for instance look for images, videos and so on. This approach was adapted to our particular case and what was created was a top separator that provides to the user the possibility to switch from regular results to personalized results, only one click distance.

Solved the problem of how to present the results not all the problems were solved, because since we are dealing with images there's always a delay related to the request of the image from the URL, the loading of the image and the presentation of the image.

Without forget that this delay could be multiplied by ten, twenty, thirty or even more depending on the number of results chosen by the user. And this delay would cause the blockage of the system, letting the user on a state where we could not execute anything on the system. The solution was to use an image loader already tested and used on other frameworks that deal with that situation. In this case what our system tells the loader is that we want the image on that spot and then asynchronously the loader gets the images from the URLs and places them on the

right spot. This loader also handles other issues, for instance if image 2 is loaded first than the image 1, the loader presents the image 2 first, don't letting, the user wait excessive time for the images and not blocking other results, if any problem is detected on a particular result.

Finally when the image is presented the view saves as a tag the information that the result is available, avoiding the possibility of the user try to save the image before it was loaded. At this stage the image can only be saved to gallery, however as a future work, it will be created a viewer, to the images where other options like the sharing to social networks will be available.

5. Case Study

Considering this project as a case study we will consider the scenarios of the searches performed by user on the system. When a new search is presented to the system, it is created a new case with their description and some information could be similar to other searches, which means that the solution can be present on previous searches stored on the case repository. In this case study the age group will be represented by minor, grown up, middle aged and old man. When converted to a numeric scale the intervals will be [14,17], [18,39], [40,59] and [60,80] respectively. Related to the case study the location is represented by the continents America, Africa, Asia, Europe and Oceania that will be converted to 1,2,3,4 and 5 respectively. In this case the other three attributes of the case are strings and the analysis will be made by another system, so for simplification of the notation the fields query, URLs and tags will be considered one attribute called description.

The following image represents a new search that will be presented to the system. This search was defined with the Age Group, Middle Age, the Location was unknown and it was described by "Description search".

With this is mind some notations will be explained, for instance there is some unknown information which will be represented by ⊥ symbol. This unknown information is present on the location of the search1 and on the age group of search2. On search1 the age group is grown up and on search2 the location is Asia. Both searches have their own description.

Considering that the description will be analyzed by the string similarity Algorithm, the Description will be placed outside the case. In regards to this fact the representation of the cases in the case repository will be the set {<raw-case, normalized-case, description-case>}, where the case represents the case itself, the Normalized-case represent the case after being changed by the normalization process and the Description specifies the problem.

In term of a predicate the information shown on the previous image will be represented by:

$$Search: AG_{roup}, L_{ocation} \rightarrow \{0,1\} \quad (18)$$

In this notation 0 and 1 represent the truth-values, false and true, respectively. The extensions of the predicates will be represented by:

$$\begin{cases} \neg Search(AG_{roup}, L_{ocation}) \leftarrow not\ Search\ (AG_{roup}, L_{ocation}) \\ \\ Search_1\left(\underbrace{[18{,}39], \bot}_{attribute's\ values}\right) :: 1 \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains\ for\ AG_{roup}, L_{ocation}} \\ \\ Search_2\left(\underbrace{\bot, 3}_{attribute's\ values}\right) :: 1 \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains\ for\ AG_{roup}, L_{ocation}} \end{cases} \tag{19}$$

The following step is to convert all the values into continuous intervals. This step will produce the following result:

$$\begin{cases} Search_1\left(\underbrace{[18{,}39], [1, 5]}_{attribute's\ values}\right) :: 1 \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains\ for\ AG_{roup}, L_{ocation}} \\ \\ Search_2\left(\underbrace{[14, 80], [3, 3]}_{attribute's\ values}\right) :: 1 \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains\ for\ AG_{roup}, L_{ocation}} \end{cases} \tag{20}$$

With all the attributes in continuous scales, it is now possible to proceed with the normalization $\frac{Y - Y_{min}}{Y_{max} - Y_{min}}$

$$\begin{gathered} Search_1([0.061, 0.379], [0,1]) \\ Search_2([0, 1], [0.5,0.5]) \end{gathered} \tag{21}$$

This normalization has to be made before the introduction of the concept of DoC which is calculated by the equation $DoC = \sqrt{1 - \Delta l^2}$

$$\begin{cases} Search_{1_{DoC}}\left(\underbrace{0.948, 0}_{attribute's\ confidence\ values}\right) :: 1 \\ \underbrace{[18, 39], [1, 5]}_{attribute's\ values} \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains} \\ Search_{2_{DoC}}\left(\underbrace{0, 1}_{attribute's\ confidence\ values}\right) :: 1 \\ \underbrace{[14, 80], [3, 3]}_{attribute's\ values} \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains} \end{cases} \quad (22)$$

With the case repository normalized, into this approach it becomes clear that system "speaks" the same language. So in order to present the new case to be handle by the system, this new case will have to be normalized by the previous processes.

$$\begin{cases} \neg Search(AG_{roup}, L_{ocation}) \leftarrow not\ Search\ (AG_{roup}, L_{ocation}) \\ Search_{new}\left(\underbrace{[40{,}59], \perp}_{attribute's\ values}\right) :: 1 \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains} \end{cases} \quad (23)$$

Next step will be the conversion to continuous intervals providing the following result:

$$\begin{cases} Search_{new}\left(\underbrace{[40{,}59], [1, 5]}_{attribute's\ values}\right) :: 1 \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains} \end{cases} \quad (24)$$

The following step is the normalization and the new case is converted to:

$$Search_{new}([0.394, 0.682], [0, 1]) \tag{25}$$

Finally the calculation of the degree of confidence will place the new search in the same language of the system.

$$\begin{cases} Search_{new_{DoC}}\left(\underbrace{0.958, 0}_{attribute's\ confidence\ values}\right) :: 1 \\ \underbrace{[40{,}539], [1, 5]}_{attribute's\ values} \\ \underbrace{[14, 80], [1, 5]}_{attribute's domains} \end{cases} \tag{26}$$

Having all the cases working on the same language, the system becomes now more able to retrieve cases with higher similarity value. This similarity will be achieved by calculating the average between the attributes. In this case it will be assumed that all the attributes have the same weight, if the weights were different the average, must be weighted. When compared the search case three with the other two the results were the following:

$$\begin{cases} Search_{1_{DoC}}\left(\underbrace{0.948, 0}_{attribute's\ confidence\ values}\right) :: 1 \\ Search_{2_{DoC}}\left(\underbrace{0, 1}_{attribute's\ confidence\ values}\right) :: 1 \\ Search_{new_{DoC}}\left(\underbrace{0.958, 0}_{attribute's\ confidence\ values}\right) :: 1 \\ Search_{new_{DoC \to 1}} = \dfrac{|0.958 - 0.948| + |0 - 0|}{2} = 0.005 \\ Search_{new_{DoC \to 2}} = \dfrac{|0.958 - 0| + |0 - 1|}{2} = 0.979 \end{cases} \tag{27}$$

In this notation the "|" represents the modulus which is the absolute value of a given number.

Using Jaro-Winkler as the string similarity Algorithm the similarity value between the new search and the other two were the following:

$$\begin{cases} Search_{sim_{new \to 1}} = 0.425 \\ Search_{sim_{new \to 2}} = 0.908 \end{cases} \quad (28)$$

With all these data it is now possible to get the final similarity measure between the new case and the cases on the repository.

$$\begin{cases} Search_{new \to 1} = \dfrac{0.425 + 0.005}{2} = 0.215 \\ Search_{new \to 2} = \dfrac{0.908 + 0.979}{2} = 0.944 \end{cases} \quad (29)$$

Looking into these results it is possible to conclude that using our new approach the best solution for the search presented to the system is the search two because it is more similar to the input search.

6. Results

Until the development of this work the search engines already in use behave on a very similar way. They try to use as a first filter the location of the user, matching the results in regards of the country. The other filter applied is the history of previous searches, trying with this to establish a context to every search performed. Finally other way used to customize the results is the language.

These factors helped to solve a problem that was present in the beginning of all search engines that was presenting the same results to every user. This was bad because with the growth of the search engines and the increase of the number of users from all over the globe some companies started advertising their product or stores and if I am a Portuguese user and I am looking for the nearest restaurant of some sort of food I am not interested in restaurants in America.

That happened all the time because the results were ordered by the number of views. Nowadays it is different as it was explained, however the personalization of the results was not related with the user itself, the only factor that was directly related to the user were the previous searches. If we imagine two users that perform the first search near each other, they will be faced with the same results because all other factors analysis only goes to the country level.

Figure 25 - Result Sport Screen.

The Figure 25 shows the results of BING according to the "sport" query before we perform our personalization approach. Those results are not taking into account the tags added by the user and the usage of the CBR system that uses techniques of handling default data and string comparison.

What we try to perform with our solution was to place the results closer to the user's will, giving him a much more intimate relation with the searches that he will perform. At the beginning of the usage of this system it is a little bit difficult to get personalized results because the Bing API provide us the results for a specific query however does not provide us a database of images to work with. This would be great because it would be possible to process the images before they were introduced into the system. The fact of not providing an image database will make our system grow with the searches performed by the users until our database of images match the images database of Bing. When this happens we will take full advantage of our personalization approach.

In addition, we add a tag approach to our searches where each user can tag their search helping the system to retrieve better results not only to the user that performs the search but also to the other users since all the tags are stored on database connected to respective query and result.

Finally our approach to the handling of default data together with the string comparison metric provides the following results.

Figure 26 - Personalized Results Screen.

Those results presented on the Figure 26 were provide by the CBR system that after analyzing the parameters of the new case, detected three best matches with the similarity value between 0.50 and 0.62, this values of similarity could be seen as a little bit low, however there are two factors that have to be taken into account. The first one is related to the strings similarity metric, that have some difficulties to analyze the strings because detecting similarities between text is not a simple task. The second factor that makes this values a little bit low, is related to the few searches that were presented to the system.

With the control of the first factor, what was tried to do was the change of the string similarity measure. Using the Dice's Coefficient metric the results in terms of images were very similar however the similarity values were bigger being between 0.70 and 0.75. Using the Levenshtein string similarity metric, the results were completely opposite with the similarity value lowering below the 0.34. The change of the string similarity metrics were something that were very tested during the tests of the CBR system because there are a few metrics that try to handle the issue of the string similarity but some of them are more effective for a group of strings and others are more effective for small string. However it was detected that the Jaro-Winkler metric seemed to be the most balanced for the variety of queries that were performed and for that reason it was chosen to be the default metric to be used together with the handling default data technique.

A solution that was tried to be applied was the calculation of the best metric, the one with greater similarity value and the retrieval of the results from the best match, however with only a few cases on the repository it was concluded that this solution wouldn't be reliable in terms of time spent on the calculation of the best matches and so it was abandoned.

As a future work it would be great to have a system that release this concern from the user, a system that could decide for each situation which similarity metric must be used. This system could use also the CBR technique and work in parallel with the already existing CBR.

The second factor, which is related to the few searches already performed, evolved with the tests that were performed since the system became available. However the biggest improvement will happen when a significant number of users start to use the application, because the number of matches and the quality of them will increase.

From the tests that were performed a few more conclusions were drawn. For instance it was perceived that the system returned results very similar to the query sport ([0.50,0.62]) when working with searches until two or three words, excluding definite or indefinite articles and other similar terms. The system behaves better, when working with direct searches like the normal search engines do. This happens because what the similarity algorithms will do is compare words or parts of it on the case's repository and those terms like "a", "an" or "the" will reduce the similarity value causing some good results to be ignored.

Finally it was concluded that searches with more than four words are difficult to the system to determine some results that should be retrieved. In some searches performed with more than four words, some similarity metrics like Jaro did not find any matching case. In this case maybe the creation of an algorithm that use the same principles of several similarity metrics could be a solution to find the best results for the similarity of huge strings.

The following image represent the results of the search "How to play football" which lead to a similarity between 0.15 and 0.23. These results were only accepted to be presented on this chapter helping to explain the bad results when using long strings.

Figure 27 - Personalized Results Screen with long query.

These results would never be presented since it was defined that the threshold would be 0.40 for now.

The threshold is the minimum value of similarity that will be accepted as a result by the system. However this value was set to 0.1 in order to understand how the system reacts with longer strings.

With the increase of the number of the users on the system it is expected that the system evolves, and the threshold increases into values that allow us to provide results with a minimum connection to the user's will.

As it was seen from the previous two images it is possible to understand that the results vary from each other. In the first picture the images provided by the system are without any personalization according to our CBR system or tags added by user the are only taking into account the query that was performed and some abstract context that was assumed by Bing using location, language and some searches.

On the second image it is possible to detect some different images that appeared. This is because on those results are present some factors such as the CBR System results according to the analysis of the similar results using the handling default data and string comparison techniques, another factor are the tags added by the user. Those factors connected to the

previous factors such as the location of the user when the search was made, the language of the query and finally the previous searches.

This search shown here and many others that were performed helped to understand that this system can be useful because after a few searches the system start to retrieve some similar results which make us consider that when used on a large scale this system can be more useful and more accurate on its results providing the best experience to users that look for the appropriate content to his current will.

In sum the results were good since it was possible to find relevant matches for almost every searches. There were some cases where no results were provided or the similarity is not enough to be considered, but cases represented edge cases where not even the user was clear about what he was looking for.

Besides that, these results were assumed as a first step into the achievement of a solution able to handle the needs of the users when searching over the internet, so some new paths were already identified as the future work to be executed taking this project into the next level. It is important to refer again that the images context given in this project were just a way to present the results since, no work were performed with images.

7. Conclusion and Future Work

In this chapter it will be provided a general overview over the work developed during the elaboration of the dissertation.

Furthermore it will be explained some relevant work, related to the problem discussed on this dissertation that were made as a paper publication.

Finally it will be described some future work and some new paths to be explored.

7.1. Synthesis of the work done

This work started with the choice of the topic to be discussed on this dissertation. Then were set the objectives to be achieved and the motivations for the development of the work, those considerations are present on chapter 1.

After setting some directions for this work it was made a study about what was already developed regarding the topic in study and the concepts that will be handled, such as Case-Base Reasoning, Personalization and Personalization on Images Search (chapter 2).

With the objectives set, with the evolution and with the evaluation of the similar solutions analyzed, some developments were made in regards to the architecture and the technologies that would be necessary to fulfill the objectives that were proposed before.

One of the objectives was the development of a system of knowledge extraction able to retrieve the results effectively and able to handle default data. In regards to this system some decisions were taken related to the techniques of Intelligent Systems to use such as CBR, GEA or ANN. The result was a Java Servlet that is capable to receive a http request, analyze the cases and finally retrieve the result shown on the results presented in chapter 5. The search engine seen on results, was based on BING API, that was chosen to fulfill the objective of using a search engine able to be used along with the CBR system. Still related to the CBR system he is able to communicate with multiple frameworks as long as they can create an http request as it was said before and that was another achievement proposed on the beginning of this work.

Finalized the CBR system implementation, were developed some layout to be used on the mobile application and it was defined how the software will operate. The mobile application was another objective, achieved as it can be seen on results. The biggest concern at this stage was how to present the personalization issue into the user and how to make it work naturally, just like a normal system would do.

To create the solution that were explained and to fulfill the objectives, the platform was developed in two pieces. The server side was developed using java and implanted a CBR system that were responsible for receiving the new case, handle it and convert it into the new structure, then present him into the knowledge base and finally select the one who better fits a solution to the "case problem" presented.

The client side was developed also in Java as a mobile application for Android using as image search the BING API. With the adoption of Java it was possible to develop each piece in modules taking advantage of its modularity feature. This allowed the system to add or change some features without breaking all system. The communication between the two pieces was not forgotten and to that was used http communication using servlets.

It was also taken into account some gaps that were discovered during the development of this solution and the analysis of other softwares. For instance it is possible to select the

similarity metric to be used within CBR and it is possible to set the weights of the attributes manually. The usage of free text input was also another discovered gap however it was a prerequisite of the scope of the problem. These possibilities added, allow the user to narrow the search in so many other ways that it can access different information only manipulating the system according to its will.

After all this work the platform was tested with positive results and some improvements were made since then, however that doesn't mean that is everything at the best. It is always possible to improve, finding different solutions able to cover some paths that were never experienced before and cover some gaps that were not detected until now. But now it is stable and it is a reliable solution.

7.2. Relevant work

During the development of this project some challenges have emerged and one in particular seemed to be possible to explore and learn how it could be improved. The CBR systems created already seems to be much complex not so adaptable and does not fulfill all the needs of this project. So it was taken the decision to create a CBR system from scratch with the features that we thought that should be present on this kind of systems. This system is fully adaptable because the case itself is a module where it can be defined case's attributes, making each case related to the problem where will be applied.

As it was said before it is possible to use free input fields, set the weight of each attribute or even the similarity metric to use on string comparison and also the possibility to handle default data.

And it was this last feature that was explored on a scientific contribution that was submitted to ICAART 2015 and was accepted for presentation in Lisbon next year.

This paper was developed by a team composed by two members of the Department of Informatics of University of Minho a member from Department of Chemistry of the University of Évora and a colleague from WeDo Technologies. The members from University of Minho were Cesar Analide and José Maia Neves, the member from University of Évora was Henrique Vicente and the colleague from WeDo was Bruno Fernandes. I thank them again for all the help provided to this achievement of presenting this article on an international conference.

The title of this article is "Handling Default Data Under a Case-Based Reasoning Approach". This article talks about past experiences and its importance to solve problems using a CBR approach. However it is considered that the existence CBR systems are neither adaptable nor complete, and the cost of adapting it to a particular problem is too high. The solution proposed in this article is an adaptation to the CBR system that is intended to be easily adaptable and capable of handling default data, this is handling unknown or uncertain information.

ICAART (International Conference on Agents and Artificial Intelligence) is an international conference about Agents and Artificial Intelligence held by several countries of Europe since 2009 with purpose of bring together many people that study, develop or are interested in those two areas. In regards to Artificial Intelligence, they split that area in many subtopics such as knowledge representation or Evolutionary Computing for instance. Our contribution is intended to be presented as a new approach to knowledge representation.

Since it was submitted the work didn't stop and at this point the paper is being extended with some new ideas with the objective of enhance the contribution to this topic and create a journal article, The approach proposed by this article is intended to open new paths to knowledge extraction and representation and this represents another motivation to be added to the objectives explained above.

7.3. Future work

Even with the completion of the objectives that were proposed, there are always possibilities to improve this solution in so many ways such as:

- Finding better strings similarities metrics that help to filter wisely the context, providing better results to the system;
- Create another CBR system independent from the one created already, fully dedicated to the choice of the better string similarity to use;
- Understand from the usage of the framework new ways to personalize it, using other elements such as searches made on other frameworks;
- Integrate this solution with social networks, whose usage is growing and it would be important to share this images on them;

- Use social networks to collect more information about the user and use it to get better results adapted to each one;
- Give the user a possibility to vote the results as important or useless using this information within CBR to not use those results or use according to the votes;
- Create a favorite tab that provides information to the CBR, placing those images on top of results;
- Improve the server side to not compromise the system with the growing of the number of users;
- Extend the mobile platform for other users that use other OS different than Android, creating a solution cross-platform using iOS or Windows Phone.

Finally it will be finished the development of the journal article about handling default data using a CBR System.

8. Bibliography

Aamodt, A.; Plaza, E., "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", Artificial Intelligence Communications 7, 39-52, 1994.

Althoff K-D; Auriol E.; Barletta R.; Manago M., "A Review of Industrial Case-Based Reasoning Tools", AI Intelligence, 1995

Analide, C.; Novais, P., “Agentes:Internet e Comércio Eléctrónico”, 2012.

Analide, C.; Novais, P.; Neves, J., “Algoritmos Genéticos e Evolucionários”, 2012.

Analide, C.; Novais, P.; Neves, J., "Raciocínio Baseado em Casos", 2012.

Analide, C.; Novais, P.; Neves, J., “Redes Neuronais Artificiais”, 2012.

Auephanwiriyakul S. ; Attrapadung, S.; Thovutikull S. and Theera-Umpon N., "Breast Abnormality Detection in Mammograms Using Fuzzy Inference System", FUZZ-IEEE 2005: 155-160, 2005

Back, T., "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms", Oxford University Press, 1996

Balke T.; Novais P.; Andrade F.; Eymann T., “From Real-World Regulations to Concrete Norms for Software Agents – A Case-Based Reasoning Approach”, ICAIL, 2009.

Barysevich, Aleh, “Dealing with Personalized Search [Infographic][Online]”. atual. 19 Julho 2012. Available on WWW:<http://www.searchenginejournal.com/dealing-with-personalized-search-infographic>.(search in November,10th 2013).

Bauchspiess, A., "Introdução aos Sistemas Inteligentes - Aplicações em Engenharia de Redes Neurais Artificiais, Lógica Fuzzy e Sistemas Neuro-Fuzzy", 2004

Benedict G.C.; Haubl G. " Searching in Choice Mode: Consumer Decision Processes in Product Search with Recommendations ",Journal of Marketing Research, 2012

Bing blogs, "Adapting Search to You", Available on WWW:<http://www.bing.com/blogs/site_blogs/b/search/archive/2011/09/14/adapting-search-to-you.aspx>(search in January, 29th 2014).

Bing blogs, "Making search yours", Available on WWW:<http://www.bing.com/blogs/site_blogs/b/search/archive/2011/02/10/making-search-yours.aspx>(search in January, 29th 2014).

Bissol, S., Mulhem, P., Chiaramella, Y.; "Towards personalized image retrieval", CLIPS-IMAGE Laboratory, 2004.

Borkar, R., Tamboli, M., Walunj, P.; "Learn to Personalized Image Search from Photo Sharing Websites".

Brüninghaus S.; Ashley K., "Combining Case-Based and Model-Based Reasoning for Predicting the Outcome of Legal Cases", Proceedings of the Fifth International Conference on Case-Based Reasoning (ICCBR-03), 65-79, 2003.

Cardoso L.; Martins F.; Magalhães R.; Martins N.; Oliveira T.; Abelha A; Machado J.; Neves J., "Schizophrenia Detection through an Artificial Neural Network based System", 2013

Carneiro D.; Novais P.; Andrade F.; Zeleznikow J.; Neves J., "The Legal Precedent in Online Dispute Resolution", Jurix, 2009

Carneiro D.; Novais P.; Andrade F.; Zeleznikow J.; Neves J., "Using Case Based Reasoning to Support Alternative Dispute Resolution", Series Advances in Intelligent and Soft Computing, 2010

Clark D. E.; Westhead D.R., "Evolutionary algorithms in computer-aided molecular design", Journal of computer-aided molecular design 10(4):337-58, 1996

Cortez, Paulo; Neves,J., "Redes Neuronais Artificiais", 2000.

Costa, A.; Andrade, F.; Novais, P., "Privacy and data protection towards elderly healthcare"; IGI Global; 2013

Diehl K., "Personalization and Decision Support Tools: Effects on Search and Consumer Research", Advances In Consumer Research, 2003

Doman, James, “What is the definition of personalization ?”, Quora, 2012.

Gelfond M.; Lifschitz V., "The stable model semantics for logic programming", in Logic Programming - Proceedings of the Fifth International Conference and Symposium, 1070-1080, 1988

Goetz B., Peierls T., “Java concurrency in practice”, Addison-Wesley, 2006

Gosling J., McGilton H., “The Java Language Environment”, 1996

Halpern J., "Reasoning about uncertainty", Massachusetts: MIT Press, 2005.

Jelovic D., “Why Java Will Always Be Slower than C++”, 2012

Kakas A.; Kowalki R.; Toni F., "The role of abduction in logic programming", in Handbook of Logic in Artificial Intelligence and Logic Programming, 235-324, 1998

Khan A.; Hoffmann A., "Building a case-based diet recommendation system", Artificial Intelligence in Medicine 27, 155-179, 2002

Kolodner J., "An Introduction to Case-Based Reasoning", Artificial Intelligence Review 6, 3-34, 1992

Kolodsen, Janet L., "An Introduction to Case-Based Reasoning", Artificial Intelligence Review 6, 3-34, 1992

Kovalerchuck B.; Resconi G., "Agent-based uncertainty logic network", in Proceedings of the IEEE International Conference on Fuzzy Systems – FUZZ-IEEE, 596-603, 2010

Kruger, Andy Atkins, "Yandex Launches Personalized Search Results For Eastern Europe”. Search Engine Land [Online]. atual. December 12th 2012. Available on

WWW:<http://searchengineland.com/yandex-launches-personalizedsearch-results-for-eastern-europe-142186>. (search in November, 10th 2013).

Lerman, K., Plangprasopchok, A.; "Leveraging USer-specified Metadata to Personalize Image Search", 2010

Linden, R., "Algoritmos Genéticos - Uma importante ferramenta da Inteligência Computacional", Editora Brasport, 2006

Lucas P., "Quality checking of medical guidelines through logical abduction", in Proceedings of AI-2003, London: Springer, 309-321, 2003

Machado J.; Abelha A.; Novais P.; Neves J., "Quality of service in healthcare units", International Journal of Computer Aided Engineering and Technology, Vol. 2, 436-449, 2010

NetMarketShare, "Mobile/Tablet Operating System Market Share", Available on WWW:< http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qptimeframe=M>(search in August, 27th 2014)

Neves J., "A logic interpreter to handle time and negation in logic data bases", in Proceedings of the 1984 annual conference of the ACM on the fifth generation challenge, 50-54, 1984

Neves J.; Machado J.; Analide C.; Abelha A.; Brito L., "The halt condition in genetic programming", in Progress in Artificial Intelligence - Lecture Notes in Computer Science, Vol. 4874, 160-169, 2007

Neves J.; Ribeiro J.; Pereira P.; Alves V.; Machado J.; Abelha A.; Novais P.; Analide C.; Santos M.; Fernández-Delgado M., "Evolutionary Intelligence in asphalt pavement modeling and quality-of-information", Progress in Artificial Intelligence, 119-135, 2012

Pariser, E., "The Filter Bubble", 2011

Parrill, A.L., "Evolutionary and genetic methods in drug design", Drug Discovery Today, 1(12):514-521, 1996

Pedersen, J.T.; Moult, J., "Current Opinion in Structural Biology", Elsevier 6:227-231, 1996

Pereira L.; Anh H., "Evolution prospection", in New Advances in Intelligent Decision Technologies – Results of the First KES International Symposium IDT, 51-64, 2009

Pommerleau, D. A., "Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving ", Robot Learning, J.Connel and S. Mahadevan, ed., 1993

Rao L., “Microsoft Bing Search Queries Overtake Yahoo For The First Time In December”, TechCrounch, 2012

Riesbeck C.; Schank R., "From Training to Teaching: Techniques for Case-Based ITS", Intelligent Tutoring Systems: Evolution in Design, Lawrence Erlbaum Associates, Hillsdale, 55-78, 1991

Rissland E.; Ashley K., "HYPO: A Precedent-Based Legal Reasoner", Recent Advances in Computer Science and Law, 1989

Rissland E.; Skalak, D., "Case-Based Reasoning in Rule-Governed Domain", In Proceedings of the Fifth IEEE Conference on Artificial Intelligence Applications, 1989

Roman M. Balabin; Ekaterina I. Lomakina, "Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies",The journal of chemical physics, 2009

Search Engine Land, "The Periodic Table of SEO Success Factors", 2013

Seker .H.; Odetao M.;Petroric D.; Naguib R.N.G.(1994), "A fuzzy logic based method for prognostic decision making in breast and prostrate cancers", Biomedicine (IEEE transactions) 2003.73. S. Haykin. Neural Networks: A Comprehensive Foundation. Macmillan, New York, 1994.

Simpson T. W., "Evaluating Google as an Epistemic Tool", Metaphilosophy, 2012

Skalak D.; Rissland E., "Arguments and cases: An inevitable intertwining", Artificial Intelligence and Law, 1992

Sugiyama, K.; Hatano K.; Yoshikawa M., "Adaptive Web Search Based on User Profile Constructed without Any Effort from Users",ACM, 2004

Sullivan, Danny, “Bing Results Get Localized & Personalized [Online]”. Atual 2011. Available on WWW:<http://searchengineland.com/bing-results-get-localized-personalized-64284>.(search in January, 30th 2013).

Sullivan, Danny, “Chapter 8: Personalization & Search Engine Rankings [Online]”. Atual 2011. Available on WWW:<http://searchengineland.com/guide/seo/personalization-search-engine-rankings>.(search in November, 10th 2013).

Tsinakos A., "Asynchronous distance education: Teaching using Case Based Reasoning", Turkish Online Journal of Distance Education – TOJDE, 2003

Watson I. D., "Case-Based Reasoning Tools: An Overview", 1996

Willet, P., "Genetic algorithms in molecular recognition and design", Trends in Biotechnology 13:516-521, 1995

Yandex, About page, Available on WWW:<http://api.yandex.com/personalized-search/s>.(search in January, 27th 2014).