Modelo de un Meta Buscador que Realiza Agrupación de Documentos Web, Enriquecido con una Taxonomía, Ontologías e Información del Usuario

Carlos Alberto Cobos Lozada

Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas e Industrial

Bogotá D.C., Colombia, 2013

Modelo de un Meta Buscador que Realiza Agrupación de Documentos Web, Enriquecido con una Taxonomía, Ontologías e Información del Usuario

CARLOS ALBERTO COBOS LOZADA

Código: 299810

Tesis de investigación presentada como requisito parcial para optar al título de:

Doctor en Ingeniería de Sistemas y Computación

Dirigido por:

PH.D. ELIZABETH LEÓN GUZMÁN

Línea de Investigación:

Minería de Datos y Sistemas Inteligentes

Grupo de Investigación:

MIDAS – Minería de datos

Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas e Industrial

Bogotá D.C., Colombia, 2013

(Dedicated to)

My wife Martha Eliana and my daughter Laura Sofia for their love, understanding and unconditional support.

Acknowledgements

I would like to address my sincerest thanks to Professor Elizabeth Leon for giving me the opportunity to carry out this work, for her advice, time, support and guidance in all work related to this thesis.

I am very grateful to teaching staff at the Universidad Nacional de Colombia - Fabio González, Jonatan Gomez, Jenny Sanchez and Luis-Fernando Niño - for their contributions to my training as a researcher and university lecturer, and for all their help and collaboration.

Thanks also to Rafael Rengifo Prado, Dean of the Faculty of Electronic and Telecommunication Engineering, Eduardo Rojas Pineda, Vice-chancellor for Research and Danilo Reinaldo Vivas Ramos, Rector of the Universidad del Cauca for their collaboration and support.

To the Universidad Nacional de Colombia for giving me the appropriate place for my training in the Ph.D. program in Engineering-Systems and Computers.

To the Universidad del Cauca for financial support through the study commission and general support in many other activities related to the doctorate program.

Abstract

In pursuing the central theme of this Ph.D. thesis, which is effective web search, the author seeks through synergistic combination, to make the most of the different potentials of thematic indices, traditional web search engines, and meta web search engines, bypassing the weaknesses inherent in each, when they are operating in isolation. A general taxonomy of knowledge, ontologies, and user information (user profile and user feedback) are synergistically combined, together with the clustering of web results in a meta search model that brings up for the user only those results (documents) of greatest relevance, thereby reducing the time spent by users on searches.

The proposed model includes five main components. The first component is responsible for supporting the query expansion of the user based on the semantic relationship (extracted from ontologies that are organized in a taxonomic hierarchy) of the terms that each user has stored in their profile. The second component is responsible for search result acquisition from traditional web search engines (Google, Yahoo! and Bing). The third component is responsible for pre-processing documents and generating two representations of them, one based on the vector space model and another based on frequent phrases. The fourth component is responsible for cluster construction and labeling, for which there are three heuristic algorithms that perform clustering based on the vector space representation of the results, and labeling based on frequent phrase representation. The fifth component is responsible for visualization of the resulting clusters, which involves the presentation of search results organized into thematic groups (folders) and updating of the user profile based on the user feedback (relevant or not relevant).

The cluster construction and labeling component is supported by three new heuristic algorithms based on the following global search strategies: global-best harmony search, cuckoo search and a genetic algorithm. The K-means algorithm is employed as a local search improvement strategy in each of the algorithms. A new fitness function, called

Balanced Bayesian Information Criterion guides the evolution process of these algorithms and is proposed from the genetic programming approach. A hyper-heuristic framework is also presented and used to evaluate a wide set of heuristics that can be used to solve the problem of web result clustering.

The evaluation process of the model and the algorithms is based on synthetic data sets (from traditional repositories) and answers provided by a real population of users. The evaluation is supported by traditional validation metrics from the information retrieval field (precision, recall, F-measure, accuracy, and fall-out) and from user satisfaction (utility of each cluster, precision of allocation of documents in each cluster and their order, quality of labels for each cluster, and the Subtopic Search Length under k document sufficiency - SSL_k- measure used for assessing the ease with which the users can use the clustering results). The results obtained are compared against results delivered by other state of the art algorithms, among them Bisecting K-means, STC and Lingo.

Keywords: clustering search results, web clustering engine, taxonomies, ontologies, memetic algorithm, global-best harmony search, balanced Bayesian information criterion, cuckoo search, hyper-heuristic approach, user modeling, meta-search engine, personalized information retrieval, semantic search engine.

Resumen

Esta tesis doctoral tiene como tema central la Búsqueda Web. En ésta se aprovecha las potencialidades de los índices temáticos, los buscadores Web tradicionales y los meta buscadores, en un modelo que evita las debilidades que cada uno de ellos tiene por separado, y permite con ello disminuir el tiempo invertido por los usuarios en las búsquedas web. Para lograr esto, se combina sinérgicamente una taxonomía general de conocimiento, ontologías de dominio específico, información del usuario y agrupación de resultados (documentos) web en un modelo de un meta buscador que presenta

resultados más relevantes a las necesidades de información de los usuarios y de una forma mejor organizada.

El modelo propuesto contempla cinco componentes principales. El primer componente es el encargado de soportar la expansión de la consulta del usuario, basado en la relación semántica (extraída de las ontologías que se organizan en una jerarquía taxonómica) de los términos que cada usuario ha almacenado en su perfil. El segundo componente se encarga de la adquisición de los resultados desde los buscadores web tradicionales (Google, Yahoo! y Bing). El tercer componente es responsable del pre-procesamiento de documentos y genera dos representaciones de los mismos, una basada en el modelo espacio vectorial y otra en frases frecuentes. El cuarto componente se encarga de la construcción de agrupaciones y etiquetado, para lo cual se cuenta con tres algoritmos heurísticos que realizan el agrupamiento basado en la representación espacio vectorial de los resultados y el etiquetado basado en una representación de frases frecuentes. El quinto componente se encarga de la visualización de resultados, lo que implica la presentación de los resultados de la búsqueda organizados en grupos temáticos (carpetas) y la actualización del perfil del usuario basado en la re-alimentación que éste registre sobre los resultados (relevantes o no relevantes).

El componente de construcción de agrupaciones y etiquetado se soporta en tres nuevos algoritmos heurísticos basados en las siguientes estrategias de búsqueda global: la mejor búsqueda armónica global, la búsqueda cucú y un algoritmo genético. El algoritmo K-means se usa para optimizar localmente las soluciones en cada uno de los algoritmos. Una nueva función de aptitud denominada Criterio de Información Bayesiano Balanceado orienta el proceso evolutivo de estos algoritmos y fue propuesta desde un enfoque de programación genética. También se presenta el modelo de un entorno híper-heurístico que sirve para evaluar un conjunto mucho más amplio de heurísticas que pueden ser usadas para resolver el problema de agrupación de resultados web.

El proceso de evaluación del modelo y de los algoritmos se basa en conjuntos de datos sintéticos (de repositorios tradicionales) y en respuestas entregadas por una población real de usuarios. La evaluación se soporta en medidas tradicionales del área de recuperación de información (precisión, recuerdo, medida F, exactitud y fall-out) y de satisfacción de los usuarios (utilidad de cada grupo, organización de los resultados en los

grupos, calidad de las etiquetas de los grupos y la medida de longitud de búsqueda de sub tópicos mínima para encontrar k documentos relevantes -SSL_k-, usada para evaluar la facilidad con la que los usuarios usan los resultados del agrupamiento). Los resultados obtenidos se comparan con los resultados entregados por otros algoritmos del estado del arte, entre ellos: Bisecting K-means, STC y Lingo.

Palabras clave: agrupación de resultados web, motor que agrupa documentos web, taxonomías, ontologías, algoritmos meméticos, mejor búsqueda armónica global, criterio bayesiano de información balanceado, búsqueda cucú, enfoque híper heurístico, modelamiento de usuario, meta buscador, recuperación de información personalizada, motor de búsqueda semántica.

Contents

1 Introduction 1
1.1 Problem definition 2
1.2 Justification and importance 4
1.3 Objectives 6
1.3.1 General objective 6
1.3.2 Specific objectives 6
1.4 Methodology 7
1.5 Summary of contributions 12
1.6 Organization of the rest of the document 15

2 Background 17
2.1 Information retrieval 17
2.2 Thematic indices, web search engines and meta web search engines 19
2.3 Web clustering engines 21
2.4 Clustering and labeling algorithms 26
2.4.1 Data-centric algorithms 26
2.4.2 Description-aware algorithms 30
2.4.3 Description-centric algorithms 31
2.5 Query expansion process 35
2.6 User profile 37
2.7 Taxonomies and ontologies 39

3 The Proposed Model 43
3.1 Query expansion 43
3.1.1 Pre-processing and semantic relation 45
3.1.2 Concepts related to the profile 46
3.1.3 External service 47
3.2 Search result acquisition 53
3.3 Pre-processing 54
3.4 Cluster construction and labeling 56
3.4.1 IGBHSK algorithm 57
3.4.2 WDC-MA algorithm 61
3.4.3 WDC-CSK algorithm 63
3.4.4 Labeling 65
3.4.4.1 Statistically most representative concepts 65
3.4.4.2 Frequent phrases 66
3.5 Visualization 67

4 Hyper-Heuristic Framework and Web Application 69
4.1 Hyper-Heuristic Framework 69
4.1.1 The K-means algorithm 70

4.1.2 The fitness function 72
4.1.3 WDC-HH from an algorithm point of view 76
4.1.4 High-level heuristics 79
4.1.4.1 Performance-based rank selection 79
4.1.4.2 Tabu selection 80
4.1.4.3 Random selection 80
4.1.4.4 Performance-based roulette wheel selection 80
4.1.5 Low-level heuristics 81
4.1.5.1 Harmony search (HS) 81
4.1.5.2 Improved harmony search algorithm (IH) 82
4.1.5.3 A novel global harmony search algorithm (NH) 82
4.1.5.4 Global-best harmony search algorithm (BH) 82
4.1.5.5 Particle swarm optimization (PS) 83
4.1.5.6 Differential evolution (ED) 84
4.1.5.7 Artificial bee colony (CA) 84
4.1.5.8 Heuristics based on genetic algorithms 85
4.1.6 Replacement heuristics 87
4.1.6.1 Replace worst (WR) 87
4.1.6.2 Restricted competition replacement (RC) 88
4.1.6.3 Stochastic replacement (SR) 88
4.1.6.4 Rank replacement (RR) 88
4.2 Framework implementation 89
4.2.1 General architecture 89
4.2.1.1 Overview of classes 90
4.2.2 Minerva: The web application 92

5 Experimental Results 97
5.1 Proposed query expansion process 97
5.1.1 Data sets for assessment 97
5.1.2 Metrics for assessment 98
5.1.3 Compared systems 98
5.1.4 Scenarios 98
5.1.5 Results and discussion 99
5.1.5.1 CACM IR test collection - with no session memory 99
5.1.5.2 CACM IR test collection - with session memory 100
5.1.5.3 CACM IR test collection - with long-term memory 100
5.1.5.4 LISA IR test collection - with no session memory 103
5.1.5.5 LISA IR test collection - with session memory 103
5.1.5.6 LISA IR test collection - with long-term memory 104
5.2 Proposed web document clustering algorithms 107
5.2.1 Data sets for assessment 107
5.2.2 Metrics for assessment 108
5.2.3 Compared systems 109
5.2.4 Results and discussion 110
5.3 Experiments with Users 119

6 Conclusions, Recommendations and Future Work 123
6.1 Conclusions 123
6.2 Recommendations and Future work 127

Appendix A: A harmony search algorithm for clustering with feature selection

Appendix B: Expansion Algorithms based on a New Discrete Function of Relevance

Appendix C: Web Meta-Search Model Based on a General Taxonomy of Knowledge, a General Domain Ontology, Specific Ontologies and User Profile

Appendix D: Fitness Function obtained from a Genetic Programming Approach for Web Document Clustering using Evolutionary Algorithms

Appendix E: TopicSearch - Personalized Web Clustering Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents

Appendix F: Clustering of Web Search Results based on the Cuckoo Search Algorithm and Balanced Bayesian Information Criterion

Appendix G: Algorithm for clustering of web search results from a hyper-heuristic approach

Appendix H: Web document clustering based on Global-Best Harmony Search, K-means, Frequent Term Sets and Bayesian Information Criterion

Appendix I: Web Document Clustering based on a New Niching Memetic Algorithm, Term-Document Matrix and Bayesian Information Criterion

Appendix J: A hyper-heuristic approach to design and tuning heuristic methods for web document clustering

Appendix K: CMIN – A Case Tool Based on CRISP-DM to Support Data Mining Projects

Appendix L: A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes

Appendix M: Clustering of Web Search Results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion

Appendix N: Extractive single-document summarization based on genetic operators and guided local search

List of Figures

Figure 1-1: Methodology and general chronogram 8
Figure 2-1: IRS Components (Adapted from [9]) 18
Figure 2-2: WCE components (adapted from [25]) 22
Figure 3-1: General components of the proposed model 44
Figure 3-2: Components of the query expansion process 45
Figure 3-3: Query expanded by each term 46
Figure 3-4: General persistence structure of the GTK, the Ontologies and the Inverted Index of Concepts 49
Figure 3-5: User Profile Part 1, Concept-User Matrix (CUM) 50
Figure 3-6: User Profile Part II, Concept-Document-User Matrix (CDUM) 51
Figure 3-7: Algorithm used to construct the concept co-occurrence matrix (S) in each specific ontology related to a user 52
Figure 3-8: IDF function used to calculate the S matrix 52
Figure 3-9: Web result acquisition and construction of the Term-Document Matrix with the observed frequency of the terms 55
Figure 3-10: Concept-Document Matrix (Observed Frequency) Building Process 55
Figure 3-11: IGBHSK Algorithm 59
Figure 3-12: Best Memory Results 59
Figure 3-13: Steps in the GBHSK routine 60
Figure 3-14: MAK routine 62
Figure 3-15: WDC-CS algorithm 64
Figure 4-1: General diagram of WDC-HH framework 71
Figure 4-2: Classification of HH (adapted from [76]) 75
Figure 4-3: The K-means algorithm 76
Figure 4-4: The WDC-HH algorithm and the HHK routine 77

Figure 4-5: Best Memory Results 78
Figure 4-6: Tabu Selection 80
Figure 4-7: Improvisation steps of HS algorithm in HHK routine 81
Figure 4-8: Improvisation steps of NH algorithm in HHK routine 82
Figure 4-9: Improvisation steps of BH algorithm in HHK routine 83
Figure 4-10: PS algorithm in HHR routine 84
Figure 4-11: Heuristics based on genetic algorithms 86
Figure 4-12: General architecture of the framework 89
Figure 4-13: Some classes of the business logic layer 91
Figure 4-14: Minerva use case diagram 92
Figure 4-15: Minerva auto-complete option 93
Figure 4-16: Results display in Minerva 94
Figure 4-17: Minerva configuration options page 95
Figure 5-1: Precision-recall curves for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with long-term memory in four expansions 102
Figure 5-2: Precision-recall curves for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with long-term memory in four expansions 106
Figure 5-3: Precision, Recall and F-Measure for WDC-HH-BHRK (IGBHSK) through the various iterations on the AMBIENT data set 116
Figure 5-4: Effectiveness of new solution vectors generated at different number of iterations on AMBIENT data set for WDC-HH-BHRK (IGBHSK) algorithm 116
Figure 5-5: Precision, Recall and F-Measure for WDC-CSK through different iterations on AMBIENT data set 117
Figure 5-6: Effectiveness of new nest generated at different number of iterations on AMBIENT data set for WDC-CSK algorithm 117
Figure 5-7: Comparative chart of the results for each question (char of values in Table 5-12) 121
Figure 5-8: Overall comparison of the survey results by responses to each algorithm 121
Figure 5-9: Comparative grouping of survey results by response for each algorithm 121

List of Tables

Table 1-1:	New knowledge production and/or technological developments	12
Table 1-2:	Strengthening of the scientific community	13
Table 1-3:	Social appropriation of knowledge	14
Table 5-1:	Summary of IR test collections for query expansion process assessment	98
Table 5-1:	Precision-recall values for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with no user profile memory (best results are in bold)	100
Table 5-2:	Precision-recall values for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with session memory (best results are in bold)	101
Table 5-3:	Precision-recall values for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with long-term memory (best results are in bold)	102
Table 5-4:	Precision-recall values for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with no session memory (best results are in bold)	104
Table 5-5:	Precision-recall values for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with session memory (best results are in bold)	105
Table 5-6:	Precision-recall values for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with long-term memory (best results are in bold)	106
Table 5-7:	Consolidated results of best heuristics (best results are in bold)	112
Table 5-8:	Ground-Truth Validation Results (best results are in bold)	113
Table 5-9:	Ground-Truth Friedman Test Rankings for all algorithms (best results are in bold)	114
Table 5-10:	User Behavior Evaluation (best results are in bold)	118
Table 5-11:	Survey form for testing with users	120
Table 5-12:	Average results of the survey (best results are in bold)	120

List of Acronyms

Acronyms	Meaning
AMBIENT	AMBIguous ENTries data set
ARF	Automatic Relevance Feedback
BBIC	Balanced Bayesian Information Criterion
BH	Global-best harmony search
BIC	Bayesian Information Criterion
BMR	Best Memory Results
BMRS	Best Memory Results Size
C2	Social scaling parameter in particle swarm optimization heuristic
CA	Artificial bee colony heuristic
CACM	Communications of the Association for Computing Machinery information retrieval test collection
CDM	Concept-Document Matrix
CE-IDF	Query expansion model based on keywords
CM	Multi-point crossover
CORE	Computing Research and Education Association of Australasia
CR	Recombination Probability in differential evolution heuristic
CS	Cuckoo Search heuristic
CU	Uniform crossover
DMOZ	Open Directory Project
DMOZ-50	Data set with 50 queries derived from Open Directory Project
ED	Differential evolution heuristic
EPR	Exploitation Probability Random for the Artificial bee colony algorithm
FED	Mutation Factor in differential evolution heuristic
FS	Feature Selection

Acronyms	Meaning
FTDM	Frequent Term-Document Matrix
FTDM	Frequent Concept-Document Matrix
GBHS	Global-best Harmony Search heuristic
GBHSK	Global best Harmony Search with K-means routine
GP	Genetic Programming
GTK	General Taxonomy of Knowledge
HCMR	Harmony Memory Considering Rate
HH	Hyper Heuristic
HM	Harmony Memory
HMS	Harmony Memory Size
HS	Harmony Search heuristic
IDF	Inverse Document Frequency
IGBHSK	Iterative Global Best Harmony Search K-means algorithm
IH	Improved Harmony search heuristic
IR	Information Retrieval
IRS	Information Retrieval System
KeySRC	Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering. Web clustering algorithm available at http://keysrc.fub.it
Lingo	Web document clustering algorithm used in http://www.carrot2.org
LISA	Library & Information Science Abstracts information retrieval test collection
LSI	Latent Semantic Indexing or Latent Semantic Analysis
MA	Memetic Approach
MAK	Memetic Approach with K-means routine
MaxB	Maximum Bandwidth for mutation operation
MET	Maximum Execution Time
MinB	Minimum Bandwidth for mutation operation
Minerva	Web application develop for testing the proposed model
MM	Multi-bit uniform mutation
MNI	Maximum Number of Islands
MNN	Maximum Number of Nests in Cuckoo Search algorithm
MO	One-bit uniform mutation
MORESQUE	MORE Sense-tagged QUEry results data set

Acronyms	Meaning
MR	Mutation Rate
NH	Novel global Harmony search heuristic
NI	Number of Iterations
NR	No replacement strategy
NSGTK	Nodes selected from the General Taxonomy of Knowledge
ODP	Open Directory Project
ODP-239	Data set with 239 queries derived from Open Directory Project
OF	Objective Function or Fitness Function
OPTIMSRC	OPTImal Meta Search Results Clustering algorithm
PA	Probability of Abandonment in Cuckoo Search algorithm
PAR	Pitch Adjusting Rate
PCA	Principal Component Analysis
PEB	Probability Employed Bee for the Artificial bee colony algorithm
PS	Particle swarm optimization heuristic
PS	Population Size
RC	Restricted competition replacement
RG	Randomly Generated
RGS	Replacement Group Size for restricted competition replacement
RK	Rank selection
RM	Restrictive mating selection
RR	Rank replacement
RW	Roulette wheel selection
SGS	Selection Group Size for restricted mating
SR	Stochastic Replacement
SSE	Sum of Squared Error
SSLk	Subtopic Search Length under k document sufficiency measure
STC	Suffix Tree Clustering algorithm
SVD	Singular Value Decomposition
TDM	Term-Document Matrix
TF-IDF	Term Frequency – Inverse Document Frequency
UP	One-point crossover
URF	User Relevance Feedback
VP-IDF	Query expansion model based on weighted vectors

Acronyms	Meaning
VSM	Vector Space Model
WCE	Web Clustering Engine
WDC	Web Document Clustering
WDC-CSK	Web Document Clustering algorithm based on Cuckoo Search and K-means
WDC-HH	Hyper Heuristic framework for Web Document Clustering
WDC-HH-BHRK	Web Document Clustering algorithm based on Global Best Harmony Search and K-means. It is the name for the IGBHSK algorithm inside the hyper heuristic framework.
WDC-MA	Web Document Clustering algorithm based on Memetic Approach
Wmax	Particle Inertia Minimum in particle swarm optimization heuristic
Wmin	Particle Inertia Maximum in particle swarm optimization heuristic
WR	Replace worst

1 Introduction

Today, web search engines form the starting web page for the vast majority of Internet users [136, 189]. Despite this, the results shown by these search engines are not always those most relevant to the needs of the user. First, because entering only a few key words clearly leaves semantic gaps for the search engine, reducing its capacity to provide more exact results. Secondly, the fact that the World Wide Web is growing on such a huge scale means that search engines are unable to index all information in real time (this problem becomes even greater for a thematic index like DMOZ (Open Directory Project)). Thirdly, search engines do not record or make appropriate use of user information (user profile, user feedback). Many further problems could also be mentioned.

To see how this affects Internet users, a study by Dogpile.com carried out by a collaboration researchers from Queensland University of Technology and Pennsylvania State University in 2007 was analyzed. The study showed that a staggering 88.3% of the time, the query results from the four major search engines - Google, Yahoo!, MSN Search, and Ask - are different (unique). In fact, they generally have only a small percentage of their results in common.

This Ph.D. thesis research takes advantage of the potential combined strengths of thematic indices, traditional web search engines and meta web search engines, thereby avoiding their individual weaknesses. In order to achieve this, a new meta search model (web search clustering model) is put forward that brings up for the user, only documents of much greater relevance, and that are furthermore organized in thematic clusters, thus reducing the time spent by users on search tasks. The model comprises three main components: i) a general taxonomy of knowledge and ontologies linked to each taxonomy node, ii) a profile and feedback for each user, and iii) algorithms for web document clustering.

The Dewey decimal classification and the ACM Computing Classification System were the starting point in managing the general taxonomy of knowledge and ontologies suitable for use in the evaluation experiments. The proposed model is based on the vector space model, which is traditional in terms of information retrieval theory, but incorporates the management of concepts, by documents. Regarding the visualization of results, specifically in the area of web document clustering, three global search strategies were used in our experiments - global-best harmony search (GBHS), cuckoo search and a genetic algorithm - together with a local search algorithm such as K-means. Finally, a frequent phrase-based approach was used for labeling clusters.

The evaluation process of the model was initially conducted with synthetic and traditional repositories (DMOZ, AMBIENT, MORESQUE, and ODP-239) or data sets, then with a real user population (in this case, students in the Systems Engineering programs of the Universidad del Cauca in Popayán), using the traditional measurements from the information retrieval field and comparing the obtained results with other state of the art web document clustering algorithms, namely: Bisecting K-means, STC, Lingo, Lingo3G, KeySRC, OPTIMSRC, and Yahoo! results.

1.1 Problem definition

Web search engines today are the initial page for most internet users [136, 138, 177]. Although very useful, they present some difficulties: each one has a separate user interface, each interprets queries in its own way, they all support different kinds of advanced search functionalities, use different kinds of search algorithms and show different sets of results for the same search conditions (keywords). In April 2007, Dogpile.com, in collaboration with Queensland University of Technology and Pennsylvania State University, compared results from web searches on Google, Yahoo!, Windows Live™(formerly MSN Search) and Ask™(formerly Ask Jeeves). They discovered that [48]:

- 88.3% of web search engine results were unique to that search engine; 8.9% were shared by two search engines; 2.2% shared by three search engines; and just 0.6% of results were shared by all four web search engines.

- On average, 69.6% of first page results on Google [49] were unique to Google; with Yahoo! that figure rose to 79.4% and with MSN Search, 80.1%.
- In relation to non-sponsored web searches, first results concurred in only 3.6% of cases; the four web search engines never concurred on their first three results, and more than 38.6% of the time their results were totally different.

To obtain better search results, users could make a query manually in each of the four most popular web search engines, take the first page of results of each and analyze them in detail, but they would have to deal with different interfaces (adding confusion and cognitive overload [125]) and they would likely give up since the task is long and time-consuming. To provide support for this web search strategy, meta web searchers appeared, among them WebFerret [58] and DogPile [89]. These searchers make use of a unique interface from which to search and retrieve information from several web search engines - allowing the user to save search results in a history file and filter them - and try to simplify the language for communication with the user.

But whether using a traditional or a meta web search engine, it is common that queries return inconsistent results referring to irrelevant documents (documents that comply with the search criterion but are not relevant for the user) [114, 131]. This happens in most cases because search engines make no use of a user profile in order to identify specific needs. Neither do they take advantage of user feedback to improve future results for users having a similar profile. Finally, it is also common that keywords used by users are too general - too vague - and may have several different meanings (polysemy).

The model of document representation employed for several web search engines is based on vector space model. In this model, documents are seen merely as bags of words, ignoring any relationship among or between those words (synonyms, hyper-nyms, hypo-nyms [9, 10]). Several solutions are proposed, include latent semantic indexing (LSI) [9, 145], re-ranking and filtering with ontologies [182] in a specific field of knowledge or general like WordNet [56], among others.

Finally, not only are results generally different across search engines, but they are displayed as an ordered list. Users check documents in sequence, wasting time reviewing documents on irrelevant subjects, making such a model for result visualization not very

practical, since normally only the first of the documents featured on the first results page are checked [95]. As a result, clustering models for result visualization are gaining popularity on sites known as web clustering engines (WCE) like Carrot (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy[1] (http://yippy.com/), iBoogie (www.iboogie.com), KeySRC (http://keysrc.fub.it), and WebClust (http://www.webclust.com); and the number of scientific publications related to techniques for web document clustering [2, 24, 26, 59, 63, 84, 107, 109, 110, 116, 117, 122, 131, 133, 146, 167, 173, 179, 186, 205] is growing.

Despite progress, results in different web clustering engines and independent algorithms show there is still much to do. In recent studies, precision, recall, and F-measure reported values between only 0.6 and 0.8 (their values depend on the various data sets), when the target value is 1.0. Specifically, there is a call for research with a more holistic approach in terms of components involved in the search process.

In this thesis and from a holistic perspective, a new meta web search model is presented that improves current levels of relevance in document results shown to users who make keyword-based queries. This meta search model reduce the vagueness of the queries and make appropriate use of user information (profile and feedback) [196]. The model takes results delivered by the three most-used web search engines in the world (Google, Yahoo! and Bing). It is based on the integration of one general taxonomy of knowledge, ontologies, user information (profile and feedback), and web document clustering (based on a concept-document matrix) to improve user satisfaction (measured by precision, recall, and F-measure of the documents presented in each cluster) when searching information on the web.

1.2 Justification and importance

One of the questions that arose in the development of this project was: "Which pages does a user really want to retrieve when typing keywords into a web search engine?" While search engines are very popular [96, 136, 138] and extremely useful when wanting to retrieve information on the web, their internal functioning still presents flaws in filtering, sorting and handling the information semantics, thus presenting results that often have

[1] Originally named as Vivisimo and then as Clusty

nothing to do with the query performed. It is here that this project provides a better solution through the integration of taxonomies, ontologies, clustering (used today in different areas [88, 175, 176]), and user information (profile and feedback). The aim of integrating these concepts was to build a meta web search engine that carries out better filtering, sorting and visualization of the results delivered by the three most-used web search engines on the Internet today - Google, Yahoo! and Bing.

Each component mentioned has a specific objective and works synergistically with the other components. The general taxonomy of knowledge and the ontologies add semantics to the query (user request) and in this sense reduce the vagueness of the queries made using only simple keywords. This means two things: 1) users must be aware that they are looking implicit in specific branches of knowledge (nodes of the taxonomy) and 2) there should be a manual quality certification process of the ontologies associated with each branch of the taxonomy. When the user does not select a taxonomy node, the proposed model uses an automatic and approximate (based on cosine similarity, ontologies and user profile) way to define that node.

The clustering technique along with feedback from the user (a page is relevant, not relevant or simply ignored by the user) allow emphasis of the personalization of future searches (query expansion process).

From an academic and scientific point of view, it can be said that most research into the web search has been made to improve only specific aspects of search engines, but in recent years a more holistic approach to improvement has begun. This project has that same vision, and integrates components for which up until now no reports of similar research are known. From this perspective, new knowledge for the international scientific community was produced; this knowledge may directly be applied in the most-used web search engines or in direct marketing of the experimental prototype developed in this project. In addition, three new and alternative algorithms for web document clustering based on advanced meta-heuristics were proposed.

From a practical point of view, the proposed model reduces the time internet users spend in information retrieval processes and avoids them reading and reviewing resources unrelated to the queries formulated.

1.3 Objectives

Below are the objectives achieved in the course of this research Ph.D. thesis.

1.3.1 General objective

To model, develop and evaluate a meta web searcher that performs the clustering of documents resulting from traditional web searchers, enriched with a general taxonomy of knowledge, ontologies and user context information (profile and feedback), thereby seeking to provide greater relevance in search results and reduce the time spent by users on these searches.

1.3.2 Specific objectives

- To define a meta web searcher model that will:
  - use the results of traditional web search engines (taking advantage of the fact that these web search engines continually index the web)
  - expand the web query supported in a general taxonomy of knowledge and concepts of the ontologies associated with that taxonomy
  - consider key aspects of user information (profile and feedback) about previous queries to customize the search process more effectively
  - perform a web document clustering process based on the snippets returned by traditional web search engines, the selected ontology, the user information (profile and feedback), and a concept-document matrix with frequent concepts
- To model and implement three algorithms based on hybridization of global-best harmony search, cuckoo search, and one genetic algorithm with K-means algorithms, to solve the web document clustering problem.
- To define a new fitness function in order to guide the optimization process in web document heuristic algorithms based on k-means from a genetic programming approach.
- To model and implement a hyper-heuristic framework for web document clustering that will include: four high-level selection strategies, a wide set of low-level heuristics (some of them based on micro-heuristics), four replacement strategies, the K-means algorithm, and the Balanced Bayesian Information Criterion.

- To develop a web application based on the proposed meta web search model, with a multi-tier architecture and XML web services that support the logic and use of web search engine APIs (Google, Yahoo! and MSN Search).
- To evaluate the model (through user satisfaction[2], average response time and relevance using precision, recall, F-measure, accuracy, fall-out and SSL_k), comparing the results of the meta web search algorithms with the result provided by other state of the art algorithms, among these Bisecting K-means, STC and Lingo.

1.4 Methodology

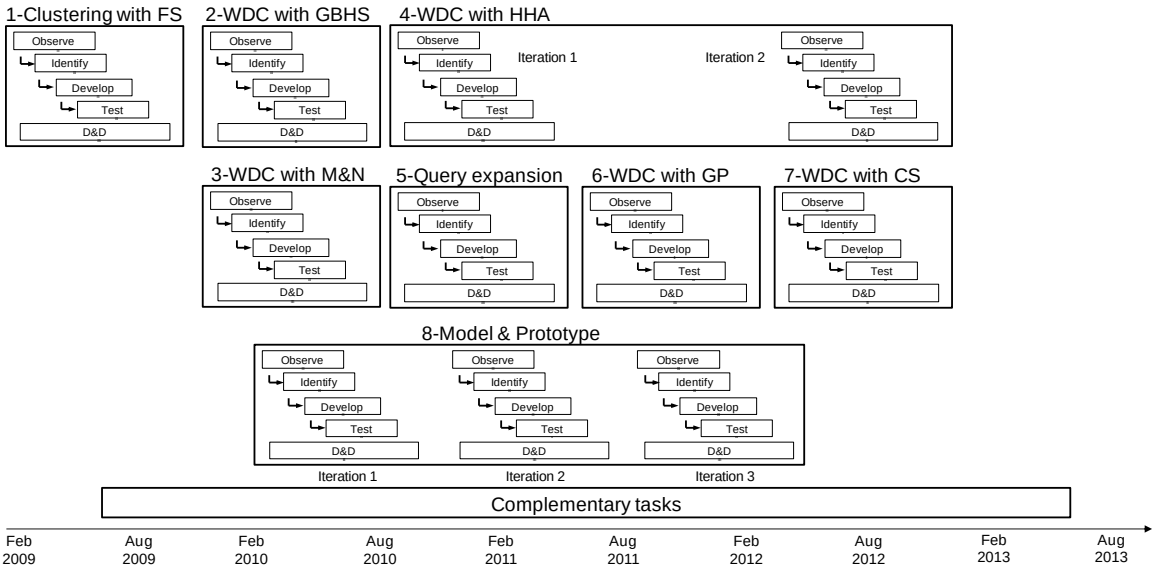
This research was oriented by eight (8) instances of the research process proposed in the Iterative Research Pattern [153]. The process originally had four (4) steps: Observe (the problem), Identify (the problem), Develop (the solution), and Test (the solution). In this research, an additional phase was added, relating to Complementary Tasks. This phase includes the writing-up of the papers, a continuous bibliographic gathering and analysis (to keep the state of the art up to date), the systematization of the project (in terms of its development process and the products obtained products), publication of the results in international journals and events, among other activities.

Each instance was aimed at developing a specific product (see **Figure 1-1**). Products were: 1) a clustering algorithm with feature selection (Clustering with FS), 2) a web document clustering algorithm based on global-best harmony search (WDC with GBHS), 3) a web document clustering algorithm based on memetic algorithms (WDC with MA), 4) a web document clustering algorithm based on the hyper-heuristic approach (WDC with HHA), 5) a new query expansion process based on feedback and a new IDF function, 6) a new fitness function for web document clustering evolutionary algorithms obtained from a genetic programming approach (WDC with GP), 7) a web document clustering algorithm based on cuckoo search (WDC with CS), and 8) the entire proposed model and the application web (Model & Prototype). Most of the instances were executed in a single iteration, but instances 4 and 8 required further iterations.

[2] Indicators previously used by Lingo: utility of each cluster, precision of allocation of documents in each cluster and their order, and quality of labels for each cluster

The first instance (1 - **Clustering with FS**) allowed defining a new data clustering algorithm with feature selection. This algorithm was published in an ISI national journal rated category A1 by PUBLINDEX-COLCIENCIAS (See Appendix A) [37]. With the development of this product the following were defined: 1) the feature selection process is computationally expensive and this process is not feasible in an online scenario. Text (or snippet) collections have a lot of dimensions (features) but feature selection is not viable for inclusion in the clustering of web documents. 2) Those indexes that are computationally expensive to evaluate the quality of clustering solutions should be avoided in the clustering of web results (unless an approach such as STC is used).

Figure 1-1: Methodology and general chronogram



The second instance (2 - **WDC with GBHS**) allowed defining a new web document clustering algorithm based on Global-Best Harmony Search. This algorithm was published in an international event (rated category A by the Computing Research and Education Association of Australasia - CORE) (See Appendix H) [35]. With the development of this product the following were defined: 1) bearing in mind the short execution time of the algorithm, the local improvement strategy (K-means algorithm) should be executed in all harmony vectors (solutions). 2) Reuters-21578 data set can be used for testing, however it differs substantially from snippets in web document clustering; therefore some data sets based on DMOZ were built. 3) Using frequent term sets in the Reuters collection could be effective as a document representation model but they are less effective in DMOZ data

sets. 4) Bayesian Information Criterion (BIC) allows the effective evaluation of solutions in web document clustering in both DMOZ and Reuters data sets. A greater execution time corresponds to more accuracy in results (the evolution process is fairly well oriented by BIC). 5) Users expressed a very favorable evaluation of the algorithm, therefore, the research work is moving along the right lines.

The third instance (3 - **WDC with MA**) allowed the defining of a new web document clustering algorithm based on Memetic algorithms. This algorithm was published in an international event (rated category A by CORE) (See Appendix I) [40]. With the development of this product the following were defined: 1) Memetic algorithms offer results (based on precision and F-measure) similar to those obtained by GBHS on DMOZ data sets using term by document matrix but more experiments were required. 2) The quality of clusters and labels is better when the algorithm uses term sets instead of frequent terms sets. 3) Results of the algorithm are more promising than results of Carrot (lingo algorithm) in the test data set. 4) BIC allows evaluating solutions effectively in web document clustering. 5) Web document clustering algorithms should avoid general labels such as “others” because such labels markedly decrease cluster quality. 6) GBHS algorithms and specific evolutionary methods are suitable for the web document clustering problem, but it is necessary to define which strategy is best (this affirmation is also supported by the Non-free Lunch Theorem [18]). 7) Users also expressed a very favorable evaluation of the algorithm. 8) Bearing in mind the short period of execution time of the algorithm, the evolution process should be oriented by individual solutions, rather than by entire populations. 9) The frequent phrases approach to labeling (a variation of that proposed in Lingo) reports better results than that of statistically representative terms. Labels generated with this approach are clearer and easier for users to read.

The fourth instance (4 - **WDC with HH**) was executed in two iterations and allowed defining a new web document clustering algorithm based on memetic algorithms from a hyper-heuristic approach. Preliminary results of this work were published in an international event (rated category A by CORE) (See Appendix J) [38] and a final paper is currently in the evaluation process with an international ISI journal (category A1 by PUBLINDEX-COLCIENCIAS) (See Appendix G). With the development of this product the following were defined: 1) Memetic algorithms based on global best harmony search

report better results (F-measure and SSL_k) than ninety-six other heuristics. 2) The replace strategy was less significant but should be computationally economic. The rank replacement and replace worst acceptance strategies were almost equally appropriate. 3) Bearing in mind the short execution time, an algorithm is more effective if its evolution process generates better solutions than the initial population quickly. 4) Combining several heuristics and micro-heuristics reported competitive results (not necessary the best) but the algorithm is quite complex.

The fifth instance (5 - **Query expansion**) allowed defining a new query expansion process based on a new function (derived from the concept known as Inverse Document Frequency - IDF) over the vector space model. This new function is continuous, based on user feedback, and takes into account the relative importance of each term in the user profile. This process was published in a national journal (rated category B by PUBLINDEX-COLCIENCIAS) (See Appendix B) [36]. With the development of this product the following were defined: 1) The method proposed obtained better results than Rocchio [9, 119, 201] over the CACM (Communications of the ACM) and LISA (Library & Information Science Abstracts) IR test collections. 2) The IDF function and the proposed user profile are easily adaptable to web clustering engines. 3) The obtained results are better than Rocchio algorithm in three scenarios: without memory (user profile persists only from one query to the next), session memory (user profile persists in only a set of related queries), and long-term memory (user profile persists over all time in the system). Rocchio reports a strong decrease in the precision-recall curve over the long-term memory, while the process proposed is less sensitive to that situation. This characteristic is very important in web search because users change their search topics and the user profile must adapt quickly to the new requirements.

The sixth instance (6 - **WDC with GP**) allowed defining a new Fitness Function for Web Document Clustering Evolutionary Algorithms obtained from a Genetic Programming Approach. This process was published in an international event with Lecture Notes in Computer Science memories (rated category C by PUBLINDEX-COLCIENCIAS) (See Appendix D). With the development of this product the following were defined: 1) a new fitness function called Balanced Bayesian Information Criterion was proposed. 2) Preliminary results of BBIC in web document clustering were better than BIC results using

the same evolutionary algorithms. 3) New evolutionary algorithms should use the BIC and BBIC fitness functions and compare the results in depth.

The seventh instance (7 - **WDC with CS**) allowed defining a new web document clustering algorithm based on Cuckoo Search and Balanced Bayesian Information Criterion. This work is currently in the evaluation process of an ISI international journal (rated category A1 by PUBLINDEX-COLCIENCIAS) (See Appendix F). With the development of this product the following were defined: 1) A memetic algorithm with cuckoo search as global search strategy reported excellent results (measured by F-measure and SSL_k) against other state of the art algorithms. 2) Lévy Flights in original cuckoo search was successfully replaced with split and merge operations on the nests in the current population. 3) Several test sets, 447 in total based on DMOZ-50, AMBIENT, MORESQUE and ODP-239, show the real behavior of the algorithm in different situations, i.e. query terms are key to a better definition of the cluster labels based on frequent phrases (a modified version of the Lingo strategy).

The first iteration of the eighth instance (8 - **Model & Prototype**) allowed defining a draft of the entire model. This draft included WordNet as semantic tool for improving the quality of user queries, web clustering algorithms and user profile. This draft was presented in a local symposium (Universidad National de Colombia) and feedback received allowed improving the model. In the second iteration, the model included agents and the prototype was used as an evaluation tool. This second version of the model and prototype was extended using both online and off-line scenarios and organized in a web clustering engine called TopicSearch. This work is in the process of publication in an international Scielo Journal (rated category A1 by PUBLINDEX-COLCIENCIAS) (See Appendix E). In the final iteration, the model detailed all components originally proposed and the prototype (called Minerva) includes the full functionality of the model. A reduced version of the model that shows its abilities to work with a general taxonomy of knowledge, a general domain ontology, specific ontologies and user profile was published in a national journal (rated category B by PUBLINDEX-COLCIENCIAS) (See Appendix C) [143]. The prototype development allowed tuning some detailed aspects of the model. Prototype was developed using a multi-tier architecture and XML web services that support the logic and use of web search engine APIs (Google, Yahoo! and Bing). A detailed presentation of these components in the final version of the model is made in the following chapters.

Complementary Tasks included: 1) Advisor for six degree projects in systems engineering; 2) Advisor for six Master theses in computer science; 3) Lecturer of several courses related to information retrieval, data mining and meta heuristics in undergraduate and graduate programs; 4) Invited researcher in two short term stays (two and a half months in Idaho Falls, USA and two and a half months in Granada, Spain); 5) Assessment of a total of nineteen documents for international journals, events and contests in the research area; 6) Researcher of a new case tool for data mining based on CRIPS-DM published in a national ISI journal (rated category A1 by PUBLINDEX-COLCIENCIAS) (See Appendix K) [43]; 7) Researcher in a new way to use recommender system concepts based on singular value decomposition applied to pedagogical patterns, published in an international ISI journal (rated category A2 by PUBLINDEX-COLCIENCIAS) (See Appendix L) [42]; 8) Researcher in a new way to use Fuzzy C-means and Bayesian Information Criterion for web document clustering with promissory results published in an international event (rated category B by CORE) (See Appendix M) [39]; and 9) Researcher in a new memetic algorithm for multi-document summarization with promissory results published in an international ISI journal (rated category A1 by PUBLINDEX-COLCIENCIAS) (See Appendix N).

1.5 Summary of contributions

Table 1-1 presents the achieved outcomes of this research in relation to the generation of new knowledge and/or technological developments, with their respectively indicators and beneficiaries. Similarly, **Table 1-2** shows the achieved outcomes of the project in relation to strengthening the national scientific community, and **Table 1-3** shows the achieved outcomes in relation to processes of social appropriation of knowledge involved in research development.

Table 1-1: New knowledge production and/or technological developments

Outcomes	Indicator
A new meta web search model released at an international level	Ph.D. thesis dissertation and papers in national/international events or journals.
Web application that uses the proposed model (prototype)	Web application source code. Taxonomy and ontologies used in the experiments.

Table 1-2: Strengthening of the scientific community

Outcomes	Indicator
Formation of human resources at a professional level	Degree projects developed by students of systems engineering. Eleven (11) undergraduate students of systems engineering program (5 degree projects). One (1) degree project in progress by one student.
Formation of human resources at a postgraduate level	Completed Ph.D. thesis. One (1) Ph.D. student in Computer and Systems Engineering. Completed Master thesis. Two (2) Master of Science graduated in Computer Science. One (1) Master of Science graduated in Mathematics Education. Three (3) Master in Computer Science students developing thesis related to information retrieval and intelligent systems.
Undergraduate and postgraduate students training in Information Retrieval, Data Mining and Meta Heuristics	Optional courses in the Master program in Computers Science at the Universidad del Cauca taught in several semesters (lecturer twice in information retrieval, lecturer three times in data mining, and once in meta heuristics). Also, lecturer for two semesters in data mining course in system engineering undergraduate program. Workshop on Recommender Systems in CAVA 2010 (II Congreso Internacional de Ambientes Virtuales de Aprendizaje Adaptativos y Accesibles, Septiembre 1-3, Cartagena de Indias, Colombia).
Assessment of research proposals and international scientific papers	Eight (8) international journal papers related to Personalized Document Recommendation by Latent Dirichlet Allocation (Information Sciences), particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique (Applied Mathematics & Computation), feature reduction using a RBF network for the classification of learning styles in engineering students (Neural Computing & Applications), Integration of Particle Swarm Optimization and Immune Genetic Algorithm-based Dynamic Clustering for Customer Clustering (International Journal of Information Technology & Decision Making), Personalized Subject Learning based on Topic Detection and Canonical Correlation Analysis (Journal of Intelligent and Fuzzy Systems), Generating Interactive Narrative from Narrative Text (Journal of Intelligent and Fuzzy Systems), Fisherman Search Procedure (Progress In Artificial Intelligence), and Analysis of the influence of Evaluation Functions in the performance of a Simulated Annealing approach for the solution of the University Timetabling Problem (Progress In Artificial Intelligence). Seven (7) international event papers: four of them in IEEE WCCI 2012 related to memetic algorithms and the impact of local searchers, particle swarm optimization with local search for multimodal optimization, comparison of different optimization techniques in the design of electromagnetic devices, and the dangers of using intention as a surrogate for retention in brand positioning decision support systems. One paper in IEEE Symposium Series on Computational Intelligence 2013 related to a novel diversity maintenance scheme for evolutionary multiobjective optimization. One paper in 2013 IFSA-NAFIPS Joint Congress related to decision aids systems using fuzzy prototypes and data quality criteria. And one paper in 5th World Congress on Nature and Biologically Inspired Computing (2013) relate to forecasting FTSE Bursa Malaysia KLCI trend with hybrid particle swarm optimization and support vector machine technique. One (1) national research proposal in COLCIENCIAS related to incremental clustering (2010) and Three (3) research proposals from CYTED (2010 and 2011). Finally, several papers and projects in different Colombian universities.

Table 1-3: Social appropriation of knowledge

Outcomes	Indicator
Seven papers in national/international indexed journals directly related with thesis	Cobos, C., León, E., and M. Mendoza, "A harmony search algorithm for clustering with feature selection," Rev. Fac. Ing. Univ. Antioquia, 2010. 55: p. 153-164. JCR (ISI) National Journal. Category A1 by PUBLINDEX-COLCIENCIAS. IF (2010): 0.089. See Appendix A. Cobos, C., Estévez, E., Mendoza, M., Gómez, L., and E. León. Query Expansion Algorithms based on a New Discrete Function of Relevance. Algoritmos de Expansión de Consulta basados en una Nueva Función Discreta de Relevancia. Revista Ingenierías, 2011. 10 (1): p. 9-22. Facultad de Ingenierías Físico Mecánicas. Universidad Industrial de Santander. ISSN: 1657-4583. EBSCO Journal. Category B by PUBLINDEX-COLCIENCIAS. See Appendix B. Ordoñez, H., Cobos, C., and E. León. Semantic Web Meta-Search Model Based on a General Taxonomy of Knowledge, a General Domain Ontology, Specific Ontologies and User Profile. Modelo de un Meta-Buscador Web Semántico Basado en una Taxonomía General de Conocimiento, una Ontología de Dominio General, Ontologías Específicas y Perfil de Usuario. Revista Ingenierías, 2011. 10 (1): p. 23-38. Facultad de Ingenierías Físico Mecánicas. Universidad Industrial de Santander. ISSN: 1657-4583. EBSCO Journal. Category B by PUBLINDEX-COLCIENCIAS. See Appendix C. Cobos, C., Muñoz, L., Mendoza, M., León, E., and Herrera-Viedma E. Fitness Function obtained from a Genetic Programming Approach for Web Document Clustering using Evolutionary Algorithms. Lecture Notes in Computer Science. IBERAMIA 2012 - Ibero-American Conference on Artificial Intelligence. LNCS journal. Category C by PUBLINDEX-COLCIENCIAS. See Appendix D. Cobos, C., Mendoza, M., León, E., and M. Manic. TopicSearch - Personalized Web Clustering Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents. Polibits Journal. In evaluation process. Scielo International Journal. Category A1 by PUBLINDEX-COLCIENCIAS. See Appendix E. Cobos, C., Muñoz-Collazos, H., Urbano-Muñoz, R. Mendoza, M., León, E. and Herrera-Viedma, E. Clustering of Web Search Results based on the Cuckoo Search Algorithm and Balanced Bayesian Information Criterion. Information Sciences. ISSN: 0020-0255. In evaluation process. JCR (ISI) International Journal. Category A1 by PUBLINDEX-COLCIENCIAS. IF (2013): 3.643. See Appendix F. Cobos, C., Duque, A., Bolaños, J., Mendoza, M., and León, E. Algorithm for clustering of web search results from a hyper-heuristic approach. Applied Soft Computing. ISSN: 1568-4946. In evaluation process. JCR (ISI) International Journal. Category A1 by PUBLINDEX-COLCIENCIAS. IF (2013): 2.140. See Appendix G.
Three presentations at international conferences	Cobos, C., Andrade, J., Constain, W., Mendoza, M., and E. León. Web document clustering based on Global-Best Harmony Search, K-means, Frequent Term Sets and Bayesian Information Criterion in 2010 IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, 2010, pp. 4637-4644. ISBN: 978-1-4244-6910-9. Event category A by CORE (Computing Research and Education Association of Australasia). See Appendix H. Cobos, C., Montealegre, C., Mejía, M.-F., Mendoza, M. and E. León. Web Document Clustering based on a New Niching Memetic Algorithm, Term-

	Document Matrix and Bayesian Information Criterion in 2010 IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, 2010, pp. 4629-4636. ISBN: 978-1-4244-6910-9. Event category A by CORE. See Appendix I. Cobos, C., Mendoza, M., and E. León. A hyper-heuristic approach to design and tuning heuristic methods for web document clustering in 2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, USA., 2011, pp. 1350-1358. ISBN: 978-1-4244-7833-0. Event category A by CORE. See Appendix J.
Four papers in national/international journals/events about complementary work to this thesis	Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., León, E. (2014). Extractive single-document summarization based on genetic operators and guided local search. Expert Systems with Applications, In Press, Accepted Manuscript. ISSN: 0957-4174. Cobos, C., Mendoza, M., León, E., Manic, M., and Herrera-Viedma Clustering of Web Search Results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion in 2013 IFSA-NAFIPS Joint Congress, Edmonton, Canada, 2013. Focus Session on Soft approaches to Web Information Retrieval. Event category B by CORE. See Appendix M. Cobos, C., Rodriguez, O., Rivera, J., Betancourt J., Mendoza, M., León, E., and E. Herrera-Viedma. A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes. Information Processing & Management, 49(3), 607-625. ISSN: 1657-4583. JCR (ISI) International Journal. Category A2 by PUBLINDEX-COLCIENCIAS. IF (2013): 0.817. See Appendix L. Cobos, C., Zuñiga, J., Guarin, J., León, E., and M. Mendoza. CMIN – A Case Tool Based on CRISP-DM to Support Data Mining Projects. Revista Ingeniería e Investigación - Universidad Nacional de Colombia. Volumen 30 Número 3. December 2010. pp. 45-56. ISSN: 0120-5609. JCR (ISI) National Journal. Category A1 by PUBLINDEX-COLCIENCIAS. IF (2010): 0.049. See Appendix K.

1.6 Organization of the rest of the document

Chapter 2: Background. This chapter shows initially some basic concepts of information retrieval and types of web search engines and then presents the state of the art in web clustering engines, the query expansion process, user profile, taxonomies and ontologies.

Chapter 3: The Proposed Model. This chapter shows the proposed way of integrating a general taxonomy of knowledge, ontologies, web document clustering, and user profile into a new meta web search model.

Chapter 4: Hyper-Heuristic Framework and Web Application. This chapter describes in detail all the components of the hyper-heuristic framework developed in this research. It also provides a general description of the prototype (web application).

Chapter 5: Experimental Results. This chapter is divided into three main sections. The first section shows algorithm results on traditional data sets (DMOZ-50, AMBIENT, MORESQUE, and ODP-239) with a total of 447 queries, and compares results with Bisecting K-means, STC, Lingo, and other state of the art algorithms. Next, a detailed assessment of the query expansion process proposed is shown. Finally, in the third section the results of the entire model with users are shown.

Chapter 6: Conclusions, Recommendations and Future Work. This chapter provides a brief summary of this dissertation and its contributions. Limitations, recommendations, and future research directions are also discussed.

The bibliography contains all the references used in this dissertation.

2 Background

This chapter shows initially some basic concepts of information retrieval and types of web search engine and then presents the state of the art in web clustering engines, the query expansion process, user profile, taxonomies and ontologies.

2.1 Information retrieval

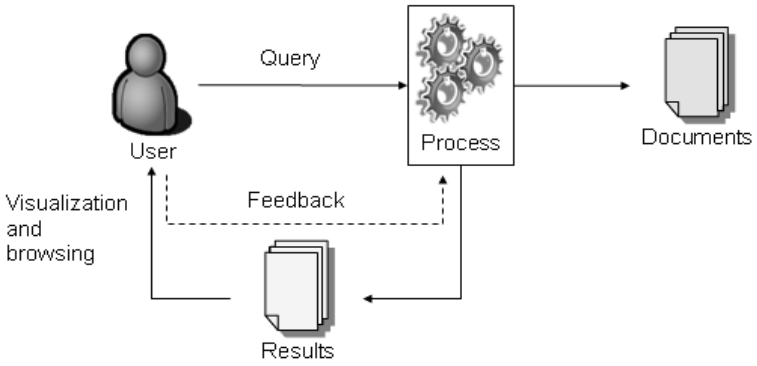
Information retrieval is an interdisciplinary field of study that looks for the best ways automatically to represent, store, organize and access items of information [9]. To understand this definition, it is necessary to consider such items of information as documents (usually unstructured) that are associated with search requests from a user [119].

Information retrieval offers the user the ability to perform searches on a large number of documents, taking into account partial matches or the best matches regarding an information request, an inference mechanism based on induction, a probabilistic search model, the possibility of classifying documents in multiple topics, the use of a query language similar to the natural language which implies incomplete query criteria, and a display of documents ordered by relevance and with a high probability of mistaking the order of display of those documents [9, 158].

Information retrieval has acquired great importance since 1940 and the increasing use of computers has created the possibility of automatically managing large volumes of information. In this context, a general structure for an information retrieval system (IRS) has been defined (see **Figure 2-1**), which mainly comprises: documents (stored in databases or directories), users, queries (requests), results/answers (related documents sorted by relevance), feedback (from the user to the system) and process (software and hardware that perform the information retrieval process) [9, 119, 158].

The central topics of research in information retrieval began with the definition of efficient storage mechanisms (indices, weighted indices, inverted indices, probabilistic indices, automatic classification of keywords, discrimination and representation), automatic classification, file structures, search strategies (Boolean model, vector space model, correlation functions, serial search, representative cluster, feedback, re-queries, probabilistic model), and evaluation (performance and user satisfaction) of the system in a collection of "controlled" documents [9, 64, 119, 158]. Over time, and specifically through the change that the Internet has imposed on people's lives, web information retrieval or web search (one of the most essential services in this environment [136, 138, 177]) had to take methodological and conceptual contributions from a great number of areas of knowledge. As such, statistics and probability, artificial intelligence, pattern recognition, parallel processing and other areas have incorporated many other "non-traditional" techniques of information retrieval: among them Bayesian networks, fuzzy logic, genetic algorithms, natural language processing, concurrent algorithms and distributed storage, while the study of multimedia data, handling of multiple languages, browsing and visualization of data has gained greater importance [9, 10, 29].

Figure 2-1: IRS Components (Adapted from [9])



There are currently several models of information retrieval (IR). The best known [9, 158] are the Boolean model, the vector space model and the probabilistic model. In addition there are some variations to these first three models, namely: fuzzy set model, the extended Boolean model, the generalized vector space model, the latent semantic indexing (LSI) model, the neural network model, the model of Bayesian networks, the network inference model, and the belief network model, among others.

Just as with any other software systems, the information retrieval systems should be evaluated before starting the operation in the real production environment. This evaluation includes aspects such as analysis of functionality, unity, integrity, fault tolerance and performance (response time to the user, additional storage space required for search index, speed of communication channels, etc). Also, the precision of the answer set should be evaluated in information retrieval systems, referred to as retrieval performance evaluation. The most popular measures to perform this evaluation are precision (fraction of the retrieved documents which are relevant), recall (fraction of the relevant documents which have been retrieved), F-measure (harmonic mean of precision and recall), Fall-out (fraction of non-relevant documents that are retrieved out of all non-relevant documents available) and Accuracy or Rank Index (fraction of relevant documents that are retrieved plus the non-relevant documents that are non-retrieved) [9].

2.2 Thematic indices, web search engines and meta web search engines

Web search can be viewed as a broader application field for the concepts involved in the original information retrieval systems. The components of a web search system are similar to those of an IRS and in this proposal the web engine plays an important role, which is taking charge of the automatic process of representation, organization and retrieval of documents dispersed on the Internet. These web engines present to the user an interface where the requests (queries on a topic, usually through a set of keywords) are entered, the system performs the search and returns the links for the user to analyze, access and decide whether they are adequate or not. There are three main types of web engines: thematic web indices or web directories, web search engines and meta web search engines [101].

Thematic indices or web directories are lists of resources organized into hierarchies from the most general to the most specific. Normally, the classification process is done manually. Web directories have the following advantages: they are easy to use for inexperienced users; the search is done by choosing the category that is closest to the query and going down into the hierarchy until it finds links to the desired resources, and there is less noise in the resources. But these directories also have some disadvantages: they only cover a fraction of the web resources and there are no uniform criteria for the

classification and selection of these resources. Some examples of these are Yahoo! (www.yahoo.com), Terra (www.terra.es), Galaxy (www.galaxy.com) and DMOZ (http://www.dmoz.org).

Web search engines crawl the network collecting and indexing as much information as possible based on automated programs known as robots (spiders or crawlers). The principal advantages of web search engines are: the processes of collecting and indexing are automatic and as a result a lot of information is collected and in addition the web search engines may have methods to automatically update that information. Among the main disadvantages are: the robots are restricted from browsing deep into the web [22] because their contents are generated dynamically through queries that have to be authenticated and authorized, among other things. For this reason, they just go over the surface of the web. These engines are also more complex to use for inexperienced users, since the users must know the syntax for the web search engine and they should be extremely careful when performing a query in order to get optimal results (the process of search refining); finally, there is not a "controlled" process of quality and reliability of the resources. Examples of these include Google (www.google.com) and AltaVista (www.altavista.com).

Meta web search engines are search systems that do not have their own databases; therefore they look at other search engines (usually web search engines). They collect the user's request and send it to the web search engines. These return the results and the meta web search engines rank them before presenting them to the user (which involves among other things, a re-ranking and a filtering process [48]). Among the most important advantages that can be mentioned are that the search is more extensive, the users access a single site to make the query and this query is typed only once. One disadvantage is that when formulating the query, the syntax may not be the best for each of the web search engines used in the background and that the search process is rather on the slow side [21]. Some examples of meta web search engines are Ixquick (http://www.ixquick.com), DogPile (http://www.dogpile.com), Webferret (http://www.Webferret.com), Copernic (http://www.copernic.com), metacrawler (http://www.metacrawler.com), Monster Crawler (http://monstercrawler.com) and mamma (http://www.mamma.com).

2.3 Web clustering engines

Although traditionally the presentation of results is carried out with an ordered list of documents according to a real value that represents the document relevance for the user, in recent years it has been considered appropriate to present the results in thematic groups (clusters). This alternative presentation of results is based on a hypothesis known as the 'cluster' hypothesis [158], according to which the 'clustering' of documents may be beneficial to users of information retrieval systems, since results relevant to the user are likely to be close to each other in the document space and will tend to fall into a relatively reduced number of clusters [122] and thereby reduce search times.

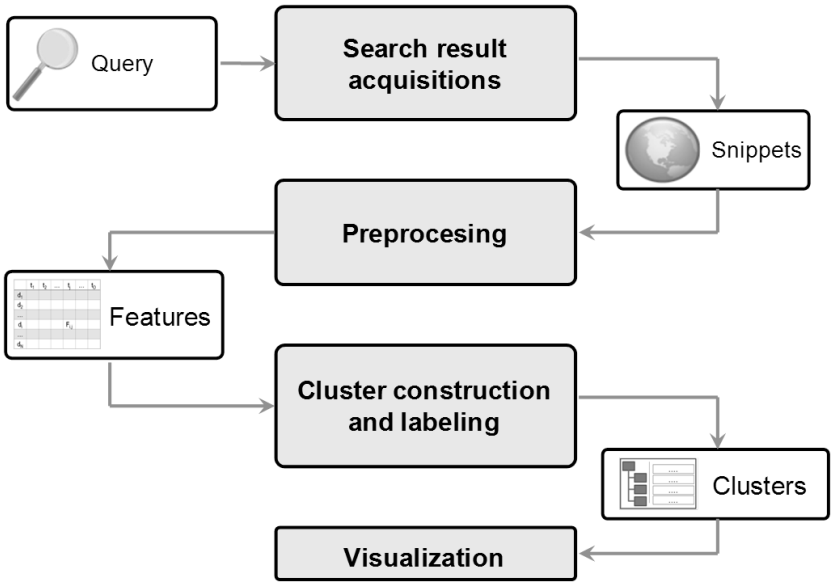
Web clustering engines (WCE) seek to increase the coverage (amount) of documents presented for the user to review, while reducing the time spent in reviewing documents [9]. Among the most prominent ones are Carrot^2 (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, originally named as Vivisimo and then as Clusty), iBoogie (www.iboogie.com), and KeySRC (http://keysrc.fub.it)[24]. But to make this kind of visualization more effective, Web clustering engines places more emphasis on the following goals [25]:

- **Fast subtopic retrieval**: if the documents are properly grouped and if the user is able to choose the right path from the cluster label, such documents can be accessed in logarithmic rather than linear time.
- **Topic exploration**: Based on the topic list of a query, the user can reformulate the same query. This is useful when queries are performed on unknown or dynamic domains.
- **Alleviating information overlook**: Users view only the first page of traditional web searcher results [95], therefore this page is the most important or dominant. With a topic list, users can quickly review a larger number of documents that may be related to their information needs, without the need to be passing from one page to another.

Web clustering engines usually consist of four main components: search result acquisition, preprocessing of input, cluster construction and labeling, and visualization of resulting clusters [25] (see **Figure 2-2**).

The **search result acquisition** component begins with a ***query*** defined by the user. With this query, a document search is conducted in diverse data sources, in this case in traditional web search engines such as Google, Yahoo! and Bing. In general, web clustering engines work as meta search engines and collect between 50 to 200 results from traditional search engines. These results contain as a minimum a URL, a snippet and a title [25]. The **preprocessing** of search results comes next. This component converts each of the search results (as snippets) into a sequence of words, phrases, strings, general attributes, characteristics or ***features***, which are then used by the clustering algorithm. A series of tasks are performed on these results including: the removal of special characters and accents, the conversion of the string to lowercase letters, removing stop words (which reduces the dimensionality by more than 40%), stemming of the words (which reduces words to their canonical stem or root form) [9] and the control of terms or concepts allowed by a vocabulary [25]. Once the preprocessing is finished, **cluster construction and labeling** is begun. This stage can be carried out using three different types of algorithm [25]: data-centric algorithms, description-aware algorithms and description-centric algorithms. Each of these algorithms builds ***clusters*** of documents and assigns a label to each group.

Figure 2-2: WCE components (adapted from [25])



Data-centric algorithms are the algorithms traditionally used for data clustering (partitional, hierarchical, density-based, etc.) [15, 25, 78, 91, 109, 116, 144, 178]. They

look for a solution in data clustering, but are lacking in their capabilities of presentation of the labels and in providing explanations of the groups obtained. These algorithms treat the clustering of web results problem like any other data clustering problem.

Description-aware algorithms give greater emphasis to one specific feature of the clustering process. For example, they might prioritize on the quality of the labeling of groups and as such achieve results that are more easily interpreted by the user. The quality of these algorithms however deteriorates during the cluster creation process. A good example of this type of algorithm is Suffix Tree Clustering (STC) [144], which incrementally creates labels easily understood by users, based on frequent phrases that appear in the documents.

Description-centric algorithms [13, 25, 67, 109, 122, 150, 203] are designed specifically for clustering of web results (or web document clustering), seeking a balance between the quality of clusters and the description (labeling) of clusters. An example of such algorithms is Lingo [150] (implemented by www.carrot2.org), which makes use of Singular Value Decomposition (SVD) to find the best relationships between terms, but groups the documents based on the most frequent phrases in the document collection.

Finally, in the **visualization** component, the system displays the results to the user in hierarchically organized folders. Each folder seeks to have a label or title that represents well the documents it contains and that is easily understandable for the user. As such, the user simply scans the folders that are actually related to their specific needs. The presentation folder tree has been adopted by various systems such as Carrot2, Yippy, SnakeT, and KeySRC, since this metaphor is already familiar to computer users. Other systems such as Grokker and Kart004 use a different display scheme based on graphs [25].

The two predominant problems with existing web clustering engines are inconsistencies in cluster content and inconsistencies in cluster description [25]. The first problem refers to the content of a cluster that does not always correspond to the label. Also, the navigation through the cluster hierarchies does not necessarily lead to more specific results. The second problem refers to the need for more expressive descriptions of the clusters (cluster labels are confusing).

In order to obtain satisfactory results in web clustering engines, the cluster and labeling component must meet the following specific requirements [78, 144]:

- automatically define the number of clusters that are going to be created
- generate relevant clusters for the user and assign documents to the appropriate clusters (clusters in the document collection have extremely skewed distributions of cluster sizes)
- define labels or names for the clusters that are easily understood by users
- handle overlapping clusters (documents can belong to more than one cluster)
- reduce the high dimensionality of document collections
- handle sparse data that are very common in documents collections
- handle the processing time i.e. the algorithm must be able to work with snippets and process time should be less than or equal to 2.0 seconds
- handle the noise frequently found in documents
- optionally, have the ability to process the documents in an incremental way as soon as the system is receiving or recovering them

Another important aspect of web clustering engines is the document representation model. The most widely used models are [94]:

- Vector space model [9, 78]: In this model, documents are designated as bags of words and the document collection is represented by a matrix of M-terms by N-documents. Each document is represented as a row vector d in the terms space where d= $\{w_1, w_2, \ldots, w_M\}$, and w_i is equal to the normalized frequency term (tf_i) by the collection multiply by the document inverse frequency for that term, in what is known as TF-IDF value, which is summarized by formula (3.2) or a variation of the same. Also in this model the cosine similarity is used for measuring the degree of similarity between two documents or between a document and the user's query, calculated by formula (3.7).

- Latent Semantic Indexing (LSI) [9, 145]: This is a variation of the vector space model that uses matrices decomposition theory such as singular value decomposition (SVD) [50]. With the decomposition, hidden relationships between the terms of the collection are sought and therefore find concepts that best represent the documents [50]. Furthermore eigenvalues can be used to find the number of clusters that should be generated.

- Ontology-based model [109, 173]: This is a variation of the vector space model in which ontologies are used as WordNet to find the relationships among the terms of the collection, such as synonymy, hypernymy, hyponymy, meronymy, among others. In this way the matrix is built with M-Concepts by N-Documents.

- N-gram [144]: In this model, the document is represented as a sequence of characters. Using a sliding window of size n, the document is scanned to extract all n-character sequences, called n-grams. This model tolerates minor spelling mistakes and reaches minor language independence levels when it is used with a stemming algorithm. The similarity is based on the number of shared n-grams between documents.

- Phrases-based model [144]: In this model, documents are scanned in order to find the common phrase suffixes and a suffix tree is built in which each node represents a part of a phrase and it is associated with documents containing that suffix. Another approach is syntactical, where linguistic information is used to form the phrases. For example, it places an adjective and a noun together to form a phrase [126].

- Frequent word (term) sets model [13, 67, 109, 112, 197]: A document is represented as a transaction of terms that are frequent in a database, similar to the problem of finding association rules in data mining [31, 80, 97, 105, 106, 175]. Using algorithms such as Apriori or FP-growth the frequent terms are found, and each document has a similarity greater or smaller than this list of frequent terms that later becomes the names or labels of the clusters.

- Rich Document Representation: “In this model, the document is represented by a set of logical terms and statements. These logical terms and statements describe the relationships that have been found in the text with a logical notation close to the multivalued logic of Michalski. For example, a proposition such as “for” in the sentence fragments such as “... operating systems for personal computers...” suggests a relationship between two noun phrases “operating systems” and “personal computers”. Then, these relations are represented with a format similar to that of multivalued logic as used in the theory of human plausible reasoning; that is, operating system (personal computers)” [126].

2.4 Clustering and labeling algorithms

As mentioned above, there are three types of web document clustering algorithms: data-centric, description-aware and description-centric.

2.4.1 Data-centric algorithms

In general, data clustering algorithms can be classified into [91, 92]: hierarchical, partitional, spectral, density-based, grid-based, and model-based, among others, the hierarchical and partitional ones having been the algorithms most commonly chosen for web document clustering [78].

Hierarchical algorithms generate a dendogram or tree of groups. This tree starts from a similarity measure, among which are: single link, complete link and average link. In relation to web document clustering, the hierarchical algorithm that brings the best results in accuracy is called UPGMA (Unweighted Pair-Group Method using Arithmetic averages) [91]. UPGMA was devised in 1990 [109, 178] and is based on the vector space model, using an average link based on the clusters cosine distance divided by the size of the two clusters that are being evaluated. UPGMA has the disadvantage of having a time complexity of O(n3) and being static in the process of assigning documents to clusters.

In partitional clustering, the algorithms perform an initial division of the data in the clusters and then move the objects from one cluster to another based on the optimization of a predefined criterion or objective function [92]. The most representative algorithms using this technique are: K-means, K-medoids, and Expectation Maximization. The K-means algorithm is the most popular because it is easy to implement and its time complexity is O(n), where n is the number of patterns or records, but it has serious disadvantages: it is sensitive to outliers; it is sensitive to the selection of the initial centroids; it requires prior definition of the number of clusters; and the obtained clusters are only hyper spherical in shape (based on Euclidian distance and cosine similarity) [144]. In the K-medoids algorithm [92] each cluster is represented by one of the objects that comprise it, which is called medoid or “the true centroid”. In this way, the clusters are subsets of objects that surround the medoid object. Later, a distance function to measure the similarity between a document and a medoid is defined. The K-medoids algorithm has two advantages: it does not present limitations in the data types and it is less sensitive to outliers [14]. The

Expectation-Maximization (EM) algorithm is a popular iterative refining algorithm that assigns each object to a cluster according to a weight that represents the probability of admission in the cluster.

In 2000, Bisecting K-means [78, 109, 178] was devised. This algorithm combines the strengths of the hierarchical and partitional methods, reporting better results concerning precision and efficiency of the UPGMA and K-means algorithms. In this algorithm, the data set is initially managed as a whole cluster. Then, based on a rule, a selected cluster is divided into two using K-means algorithm. This process is repeated until the desired number of clusters is obtained. Some disadvantages of the Bisection K-means are: it does not assign adequate names to clusters, it does not manage adequately high dimensionality of document collections, and it requires that the number of clusters is defined in advance. This last disadvantage can be overcome by processing the algorithm many times and selecting the best choice, but this is extremely time-consuming.

In partitional clustering, from an evolutionary approach, in 2007 three hybridization methods between Harmony Search (HS) [69] and K-means algorithms were compared. These were: the Sequential Hybridization method, which first runs HS (in this research called HSCLUST) and then refines the best result in Harmony Memory (HM) with K-means; the Interleaved Hybridization method, which executes the sequential hybridization algorithm several times until the algorithm exceeds a threshold or a maximum number of iterations (he HM is updated if the vectors optimized by K-means are better than those in the memory); and the hybridization of K-means as an HSCLUST step that executes a step of K-means in each improvisation step of HS. This research shows that all the hybrid methods outperform (best clusters in least time) K-means and HS algorithms run independently. Generally, the last method was the best choice of the three.

Later in 2008, the HClust, HKClust, and IHKClust [116, 117] algorithms were presented in detail. The HClust algorithm is an adaptation of HS to web document clustering. HKClust is a sequential hybridization between HS and K-means. IHKClust runs HKClust for a predefined number of times, always working on the same HM. These algorithms show good results, and based on Markov Chains theory the researchers demonstrate that IHKClust converges to the global optimum. The disadvantages of this proposal are: the need to determine the number of groups (K value) in advance, sensitivity to noise and

outliers resulting from using the K-means algorithm, and the lack of a report with worldwide known data sets, for example Reuters-21578 or TREC, this last report in order to compare the algorithm with others in the research field.

Next in 2009, a Self-Organized Genetic [173] algorithm was proposed for text clustering based on the WordNet ontology. In this algorithm, a modified LSI model is also presented, which appropriately gathers the associated semantic similarities. This algorithm outperforms the standard genetic algorithm [174] and the K-means algorithm for web document clustering in similar environments. One of the disadvantages of this algorithm is that the WordNet ontology is not accurate enough when evaluating semantic similarities in some specialized areas, such as in the Reuters-21578 data set. Experiments were executed in a text clustering scenario but not in a real web document clustering scenario. Finally, not enough attention is paid to the labeling process.

In 2009, a link-based algorithm was proposed [30].This algorithm uses the web hyperlink structure to find dense units and also improve the clustering process for creating hierarchical clusters of web documents. This proposal has the advantages of creating clusters in various shapes (with high precision) and removing noisy data. For the clustering process, it uses a specific measure that provides the possibility of dynamically determining the cluster boundaries. Experimental results show higher clustering quality over other density-based clustering algorithms, but test data sets and compared algorithms are not the traditional (state of the art) in this research area. Also, the authors do not pay attention to the cluster labeling process.

A new learning algorithm based on K-means and neural networks was also proposed in 2009 [85]. This proposal uses Principal Component Analysis (PCA) to reduce the dimensionality of the document matrix (feature selection), SVD to find the similarity measure and the multilayer neural network for reducing the time of the document clustering process. The algorithm was tested with different kinds of web pages and the results were attractive. The performance of the algorithm was proved to be satisfactory and the system can be used to cluster and classify downloaded web pages and other electronic text documents, but test data sets and compared algorithms are not the traditional ones in the web document clustering research area. Also, this proposal does

not pay attention to the cluster labeling process and requires a training process which it is not in reality feasible in the clustering of web results.

Also in 2009 a new algorithm called ArteCM was proposed [27]. This algorithm uses an incremental approach for clustering documents, offers the ability to grow the number of clusters adaptively, and to employ domain-tailored similarity measures. In this proposal, an explicit centroid definition is avoided and substituted by a similarity-based concept of centroid. ArteCM was compared with two variants of K-means and SOM with satisfactory results on speeds and clustering quality. The proposed solution includes the requirement of a specific domain, specialized similarity measure and two parameters that can be a limitation since effective and efficient definition of these two components can be tricky.

The RElational Document clustering (RED-clustering) algorithm proposed in 2010 [59] takes into account both contents information and hyperlink structure of web page collection. The algorithm finds embedded patterns of web document collection, converges to a solution that includes different kinds of information: semantic visual coherence, content features and several relations with different degrees of importance between documents. The experimental results show that RED-clustering outperforms both K-means and Expectation Maximization in terms of effectiveness, purity and agreement between classes and partitions, but they have not been used on traditional benchmark data sets of the research area. Neither were they compared with other state of the art algorithms.

In 2011, an algorithm that performs spectral bisecting and merge operations over web documents, called METIS, was put forward [107]. Bisecting and merge operations are optimized to work with skewed distributions of cluster sizes. Results show an improvement in performance of approximately 56%, 49% and 36% compared with spectral bisection and K-means respectively in terms of F-measure, but in this proposal the number of clusters should be previously defined, data sets used for testing are not those traditionally used in the research area, the proposal has a cubic complexity order, and does not present a specific algorithm for the labeling process. Also, in this year another method based on multiclass spectral clustering for grouping of documents, including web pages in English and Chinese, was proposed [84]. The algorithm starts from a traditional term by document matrix (TF-IDF) but uses different preprocessing

algorithms based on the language of the web page. To construct the similarity matrix it uses the cosine similarity measure. Finally, clusters are built using a multiclass spectral clustering algorithm (based on SVD). Researchers claim that the proposed algorithm runs more quickly when the number of documents is below 200. Improvements in precision were obtained but a substantially reduced number of data sets were used. Also, in this proposal there is a lack in the labeling construction process and the algorithm requires the number of clusters to be known previously.

In 2011 a comparison of K-means results was presented using two similarity measures - cosine similarity and geodesic distance [186]. This proposal is based on the fact that documents are not represented by VSM with a flat but curved space, and that the curvature provides additional information (distance, angle, volume, and curvature) that improves the quality of the clustering process. In practice, the geodesic measure makes a weighted combination of text-based similarity measures and rank measures based on links in a hybrid approach. Results show a slight improvement in precision when using the geodesic distance, but they are not consistent across all data sets. Experiments were performed on Wikipedia documents taking into account the text of the page, hyperlinks, and a predefined value of the number of clusters (k value). Unfortunately, this scenario does not correspond to the current scenario of web clustering engines. Moreover, geodesic distance measure is more expensive in computation time.

Finally, in relation to fuzzy clustering, FTCA [121] uses a fuzzy transduction-based clustering algorithm (2010). FTCA results are promising but they are not compared over recognized data sets, and neither do they use appropriate metrics, which are necessary to correctly compare the algorithm's results with other state of art algorithms.

2.4.2 Description-aware algorithms

These algorithms give greater weight to one specific feature of the clustering process than to the rest. For example, they make as their priority the quality of the labeling of groups and as such achieve results that are more easily interpreted by the user. Their quality drops, however, in the cluster creation process.

The main algorithm based on this approach is based on frequent phrases shared by documents in the collection. The algorithm is called Suffix Tree Clustering (STC) [144]

and it was put forward in 1998. STC is an incremental algorithm with a time complexity of O(n) and consists of three logical steps: document cleaning, base clusters definition through a suffix tree, and a combination of base clusters in the final clusters. A clear advantage of STC is that it uses phrases that provide concise and significant descriptions of the clusters. However, STC has the following disadvantages: thresholds play a very important role in the clustering process and they are quite difficult to tune; the heuristic pruning of phrases tends to eliminate high quality phrases, leaving just the less informative and the short ones; if a document does not have any of the phrases extracted from the collection, it will not be included in the results, even though it could be relevant; STC does not reduce the high dimensionality of the text documents, and it ignores the semantic and lexical relationships among terms [109].

2.4.3 Description-centric algorithms

In 2001, a SHOC (Semantic, Hierarchical, Online Clustering) algorithm was introduced [203]. SHOC improves STC and includes two important concepts: Complete phrases (STC only extracts incomplete phrases) and the definition of continuous clusters (unlike in STC, with SHOC the documents can belong to several clusters with different intensity). SHOC uses the semantic, hierarchical, online clustering approach, which is based on LSI and frequent phrases. SHOC uses a data structure called suffix array (instead of STC's suffix tree) to identify complete phrases and their frequencies in a time complexity of O(n), where n is the size of collection.

In 2003, the Lingo algorithm [145-150] was devised. This algorithm is used by the Carrot2 web searcher and is based on complete phrases and LSI with Singular Value Decomposition (SVD). Lingo is an improvement of SHOC and STC, and unlike most of the algorithms, tries first to discover descriptive names for the clusters and only then organizes the documents into appropriate clusters. Specifically, frequent phrases are extracted from documents, hoping that they are the most informative source of appropriate descriptions for the topics. After that, conducting a reduction of the original terms-document matrix with SVD, Lingo attempts to discover any relationship implicit in the collection and defines the number of clusters (K value) to be obtained. Finally, it relates the descriptions of groups with documents. One disadvantage with this algorithm is that the topic separation phase usually requires algebraic transformations - in this case, SVD - that demand a lot of computing time.

NNMF (also in 2003) is another example of these algorithms. It is based on the non-negative matrix factorization of the term-document matrix of the given document corpus [192]. This algorithm surpasses LSI and the spectral clustering methods in document clustering accuracy but does not care about cluster labels.

In 2004, the Tolerance Rough Set Clustering (TRSC) [86, 104] algorithm was proposed. TRSC is based on rough sets and the adaptation of K-means, being relatively fast and achieving good quality clusters. The use of the tolerance space and its specific approximation approach allows the algorithm to discover subtle similarities otherwise undetected. TRSC uses retrieved phrases from the documents within the clusters as candidates for the cluster descriptions. TRSC has the following disadvantages: it requires a previously defined number of clusters to be formed, and the K-means adaptation has difficulty managing the objects that are in the region boundaries. This means that certain documents cannot be classified within the clusters, since these documents are clustered as “Other” documents relatively large in size.

A different approach was offered by the Pairwise Constraints guided Non-negative Matrix Factorization (PCNMF) algorithm [207] in 2007. This algorithm transforms the document clustering problem from an un-supervised problem to a semi-supervised problem using must-link and cannot-link relations between documents. In 2007, the Dynamic SVD clustering (DSC) [122] algorithm was made available. This algorithm starts with the creation of the term-document matrix. Then in an iterative way from K=2 to a specific parameter, it calculates SVD to the matrix and finds the minimum spanning tree (MST) for the graph that represents the matrix. After that, using a quality measure, DSC selects the best MST and creates the clusters. Finally, it selects the most frequent terms in each cluster and assigns the names or labels to each cluster. This algorithm outperforms Lingo, since it is not necessary to calculate the whole SVD from the original matrix. DSC has been integrated in the Noodles (http://www.db.unibas.it/projects/noodles) web search. One of the advantages of DSC is that it uses a special strategy for selecting the K value and it does not assume a fixed value, or calculate it based on fixed thresholds.

In 2008, CFWS (Clustering based on Frequent Word Sequences) and CFWMS (Clustering based on Frequent Word Meaning Sequences) [109] were proposed. These algorithms represent text documents as frequent word sequences and frequent concept

sequences, respectively. They use as a similarity measure the amount of frequent terms or concepts shared by the documents. They also show that better results are obtained when using frequent concepts sequences, hence CFWMS presents better results than CFWS. For the pre-processing of the documents, CFWMS uses synonyms, hyponyms, and hypernyms provided by WordNet ontology, which makes the gathering of the topics in the documents more accurate. Both algorithms use General Suffix Tree (an improved version of Suffix Tree in the STC algorithm) to extract frequent phrases performing a previous analysis based on the frequent itemset concept from the association rules. These algorithms have the following disadvantages: they are sensitive to noise and outliers, the use of WordNet could generate a high dimensionality, they ignore the other terms or concepts that are not frequent in the document collection, and experiments were executed in text clustering but not in web document clustering.

Proposals using frequent word sets for document representation in the clustering of web results include FTC (Frequent Term-Based Text Clustering) and HFTC (Hierarchical Frequent Term-Based Text Clustering) algorithms (2002) [13]. These algorithms use combinations of frequent words (association rules approach) shared in the documents to measure their proximity in the text clustering process. Then, in 2003, the algorithm FIHC (Frequent Itemset-based Hierarchical Clustering) was introduced [67], which measures the cohesion of a cluster using frequent word sets so that the documents in the same cluster share more frequent word sets than those in other groups. One advantage of FTC, HFTC, and FIHC is that they assign labels that describe adequately the clusters based on frequent word sets shared by the documents. These algorithms provide accuracy similar to the one reported by Bisection K-means, with the advantage that they assign descriptive labels to associate clusters. One problem of these algorithms is that they are strongly dependent on frequent word sets, which are disorganized and in some cases can not represent the documents well. In 2009, a method based on granular computing (WDCGrc) was presented [205]. This algorithm transforms the term by document matrix (TF-IDF) to a document by binary granules matrix, then, using an association rules algorithm obtains frequent word sets between documents. These frequent word sets are pruned and finally used to create clusters. WDCGrc takes the number of identical words shared by documents as a similarity measure. Finally, the paper shows that WDCGrc is practical and feasible, with good quality of clustering, but it does not use standard benchmark data sets and nor is it compared against other state of the art algorithms.

Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering (KeySRC) [16] also was proposed in 2009, an algorithm based on key phrases. These are extracted from a generalized suffix tree built from the search results. Then documents are clustered based on a hierarchical agglomerative clustering algorithm. Also in this proposal, a novel measure for evaluating full-subtopic retrieval performance is presented. The measure is called Subtopic Search Length under k document sufficiency (SSL_k) and currently is one of the state of the art measures to evaluate the performance of web clustering engines. KeySRC outperforms STC and Lingo algorithms on AMBIENT data sets using the proposed measure.

A novel approach based on the automatic discovery of word senses from raw text, a work referred to as Word Sense Induction (WSI) [133] was presented in 2010. The authors show how web directories, semantic information retrieval (SIR) systems and search results clustering systems (the most popular approach) are used to solve the query ambiguity problem. They show how SIR systems perform indexing and searching of concepts rather than terms based on different strategies, for instance, ontologies or dictionaries like WordNet. SIR systems have reported high precision on uncommon terms but still have problems when searching names instead of concepts. The key idea of this proposal was to automatically induce senses for the target query using a graph-based algorithm focused on the notion of cycles (triangles and squares) in the co-occurrence graph of the query. Then, web search results are clustered based on their semantic similarity to the induced word senses. Experiments show better results than STC, Lingo and KeySRC algorithms on AMBIENT and MORESQUE data sets.

In 2010, a study of the search results clustering problems as a meta heuristic search was performed [26], showing that a stochastic discrete optimization algorithm could provide fast approximations to the optimal solution for the search result clustering (SRC) problem. The proposed algorithm was called OPTIMSRC (OPTImal Meta Search Results Clustering) and outperforms results shown by KeySRC, Lingo, Lingo3G and the original order of results reported by the Yahoo! search engine. The labeling process uses labels generated by other algorithms (i.e. STC o Lingo) and matches the generated clusters with the most appropriate labels.

In 2010 and 2011, three new algorithms based on heuristics, partitional clustering and different strategies for labeling were put forward. The first, IGBHSK [35] was based on global-best harmony search, K-means and frequent term sets. The second, WDC-NMA [40] was based on memetic algorithms with niching techniques and frequent phrases. Lastly, HHWDC [38] was designed from a hyper-heuristic approach and allows defining of the best algorithm for web document clustering based on several low-level heuristics and replacement strategies. The above three algorithms evaluate two different document representations models (term-document matrix and frequent term-document matrix), use Bayesian Information Criterion (BIC) for evaluating quality of solutions, and all three outperform STC and Lingo.

In 2012, a new algorithm called Topical was put forward [167], modeling the problem of clustering of web results as the problem of labeling clustering nodes of a new graph of topics. Topics are Wikipedia pages identified by a topic annotator and edges of the graph denote the relatedness of these topics. The new graph is based on annotation by Tagme that replaces the traditional bag of words paradigm. This constructs a good labeled clustering in terms of diversification and coverage of the snippet topics, coherence of cluster content, meaningfulness of cluster labels, and small number of balanced clusters. Finally, a large user study conducted on Amazon Mechanical Turk, which was aimed at ascertaining the quality of the cluster labels produced by this approach against Clusty and Lingo3G systems showed that the algorithm outperforms the other approaches, improving the SSLk measure by about 20% on average for different values of k.

2.5 Query expansion process

In the Vector Space Model (VSM) commonly used in information retrieval and web search processes, it has been demonstrated that the query expansion process improves the relevance (measured by precision) of the results delivered to the users [9, 119, 201]. Usually, query expansion on a web search system is conducted from one of four perspectives: user relevance feedback (URF), automatic relevance feedback (ARF) [9, 119, 201], morphological techniques that process query terms, and semantic techniques seeking similar terms typed by the user.

URF requires the user to mark documents as either relevant or irrelevant. The terms in these marked documents that the system has found to be relevant or not are added to or

removed from each of the user's new queries [9, 119, 201]. Rocchio proposes formula (2.1) for generating the expanded query, where q_i is the query originally typed by the user, R is a set of relevant documents, R' is a set of irrelevant documents, α, β and γ are tuning constants for the model and q_e is the expanded query [9, 119, 201].

$$\vec{q}_e = \alpha \times \vec{q}_i + \frac{\beta}{|R|}\sum_{d \in R}\vec{d} - \frac{\gamma}{|R'|}\sum_{d \in R'}\vec{d} \qquad (2.1)$$

In contrast to URF, ARF (also known as pseudo feedback) automatically expands queries based on two method types: global documents and partial documents [9, 119, 201]. In the global document-based methods, all documents in the collection are analyzed and relationships established between the terms (words). As such, these methods are typically thesaurus-based. Their disadvantage is that they need all the documents, while thesaurus updating can be expensive and complex [9, 119, 201]. Other strategies dependent on the domain (or collection) may be based on clusters or groups of terms [90] and similarity of terms [55]. Unfortunately, these approaches in specific applications such as web search, can promote advertising information [55]. In approaches independent of domain or collection use dictionaries or lexical databases such as WordNet.

In the methods based on partial documents, the query is originally sent to the search engine. From the results obtained, a subset of the documents is selected (the first results being the most relevant) and with these the query is reformulated (Rocchio's formula with γ=0) and sent back to the search engine. The results from the second (or expanded) search are those presented to the user [9, 119, 201]. Studies such as Robertson & Sparck Jones [68, 159] that modify the weight of query terms, or Dillon & Desper [55] that leave the user's terms, and use terms of initially retrieved documents, are examples of this strategy.

Both expansion models bring some problems. For example, the first assumes that the user is always going to mark documents as relevant or not, while the other assumes that the first results from the original query are "all" relevant [9, 119, 201].

New approaches include social tagging and the use of semantic knowledge represented in ontologies. Social tagging approaches [18, 120] take advantage of the growing

popularity of social networks and collaborative tagging systems, extending the family of well known co-occurrence matrices. Semantic knowledge approaches [51, 135] analyze the relations of concepts and terms, functions, instances, axioms, and methods that hybridize several techniques (such as the use of ontologies in collaborative filtering and artificial neural networks [83]).

2.6 User profile

Modeling and describing correctly user interests and preferences has become one of the most important research issues, and are considered the key to improving existing retrieval systems. Thus, in the late 70s the first research that attempted to model user information and show how the systems can adapt some of their functionality was reported [7, 163]. In the 90s with the emergence of large-scale networks and the offer of massive computer services to more and more unpredictable customers [6] , a significant amount of research reports and commercial products were produced [52, 183]. Some functions that have been the subject of adaptation or customization include: content filtering [152], sequencing [183], content representation [171], recommendation [170, 211], search [61, 154], user interfaces [45, 46, 103], sequencing tasks [127], or online help [162]. In addition, some typical application domains for user modeling and adaptive systems include the education sector [127 , 171, 183, 202, 208], electronic commerce [3, 12, 44, 65], news [170, 184, 195], digital libraries [34, 206], cultural heritage [93] and tourism [52], among others.

In information retrieval, user information, especially the profile, is used together with the queries made by users to perform a process of personalized information retrieval. This customization seeks to estimate in a better way the needs of users and select the set of documents relevant to these needs [17]. In this process, the query describes the user's current search, referred to as their local interest [11], while the user profile describes user preferences over a long period of time, known as their global interest. Depending on the way global interests affect the local ones, the query operations are classified into two operations: query expansion and re-weighting of terms [9]. A system can have a combination of the two techniques, changing the weightings of the terms (even taking into account user feedback on the results of previous queries) and adding new terms to the query (expansion).

Query expansion is often used in personalized meta search engines. The meta search engine adds to user queries the terms or user profile components and sends the extended query to each search engine [111]. The user feedback (a page is relevant or not) [7, 203] may also be used to expand the query and re-weight the terms, using well-known formulas such as that proposed by Rocchio [9].

Other customization models exist, for example those based on links, in which they are directly affected by the document ranking techniques. One advantage of this approach is that the system does not have to take into account the context of the document, only the hyperlinks inherent in any website. In general, the customization algorithms based on links are modifications of PageRank from Google [57, 154], authority HITS and the Hub algorithm [5, 70]. There are different ways of introducing customized searches into PageRank type algorithms, e.g. PageRank Sensitive to topics and relevant documents [5, 154]. The alterations of the customized PageRank algorithms are mostly easy to develop, but there is still an imbalance related to scalability, since calculating these values requires high computational resources, and currently it is impossible to calculate a full personal page rank value for each user. Some solutions put forward have been the calculation of only a small set of values for a small set of topics [57, 190], or more efficient algorithms where several partial page rank vectors are calculated, allowing the combination of these for a final customized vector [154].

In contrast, the interests of the user are represented in [173] in terms of relationships and values (e.g. romantic movies, movies by a particular director x). The results are classified in terms of properties and are sorted by those that are relevant for the user. Also in [209], a novel dual representation of a user's semantic profile to deal with the user interests that may change over time is presented. This representation uses: (1) a lower-level semantic representation, consisting of user activities and standard machine learning algorithms to detect user convergence, and (2) a higher-level semantic representation that detects shifts in the user activities once this shift is detected.

The number of search engines with customization capabilities has increased, from social search engines, where users can collaboratively suggest which are the best results for a given query [124], to vertical search engines [82, 111], where users can customize a

search engine of specific domain. Among these are found, for example, Google Personal [75], Google Co-op [82], iGoogle [2], Eurekster [111], K-bus [67], and MyYahoo [99].

Other very important user information is the context, which complements the profile and has a very broad meaning, covering for example the elements most recently selected or the latest purchases made [3, 169], documents that have been accessed recently [4], web pages visited [73], previous queries and data obtained through the logging of clicks [73, 79, 87, 180], texts related to a query [60, 204], and texts highlighted by a user [204], among others. A simple solution to the acquisition of context is the application of explicit feedback techniques, such as relevant feedback [160, 203]. Relevant feedback creates a representation of the context by means of an explicit interaction with the user. In addition, registration of clicks is one of the most used resources for the acquisition of context, in fact, studies suggest that it may be a good estimator of explicit feedback [1]. In [79] a short-term profile based on the statistical combination of documents accessed in previous queries within the user´s current session is constructed.

2.7 Taxonomies and ontologies

In general, taxonomies are defined as hierarchical organized structures that represent some kind of knowledge. In taxonomies, categories are created in order to organize the elements in simple maps [164, 166]. One of the advantages of taxonomies is that they provide a basic and systematic structure of a knowledge field at different levels of abstraction, and have been used to cover big, complex information bases. Examples of taxonomies include zoological and botanical organization, classification in libraries, etc [113].

In this project, a general knowledge taxonomy (GKT) was used, which is a taxonomy that represents different areas or disciplines of knowledge in a general way (not specific to a knowledge area). This taxonomy creates a hierarchical tree of concepts and for this an existing classification knowledge that is appropriate for the needs of the project will be used. Some examples are Dewey Decimal [139, 140], United States Congress Library [188], DMOZ [47], MERLOT [123], Yahoo! [199], and Google [74].

In thematic indices, automatic document classification in multiple topics has been a research subject for years, beginning with “traditional” algorithms such as K-nearest

neighbors (K-nn) [53], Naive Bayes [53], Support Vector Machines [32], and more recently some others like that proposed in [164] where the use of dynamic taxonomies to improve the information retrieval process is reported, or in [20] where the use of the consistent bipartite spectral graph co-partitioning technique for the categorization of texts in taxonomies is reported. The use of taxonomies as ontology reference for intelligent agents to perform web searches in documents with different formats is also reported, and in the specific case of Biological Resources [181].

There are several definitions as to what is ontology, but, for the purposes of this project the following definition will be used: an ontology is an explicit specification of objects, concepts and entities of an interest area, along with the relationships among these concepts expressed through axioms [187]. This specification also provides an unambiguous terminology that can be shared by a particular community, and should be represented in a formal, readable and usable way for computers [66]. In this way the ontology provides a reference frame to understand the reality and a classification of itself, from which the concepts can be extracted to allow the creation of an abstraction of such reality [98].

Ontologies are composed of concepts (they can be kinds of objects, methods, plans, strategies, reasoning processes, etc.), relationships (e.g. subclass-of, part-of, exhaustive-part-of, connected-to, etc.) functions, instances (they represent specific objects from a concept), and axioms (e.g. “C1: If A and B are from class C, then A is not a subclass-of B, for all A that meets the C1 condition, A is B”, etc.), that are useful to represent knowledge in a specific domain.

Ontologies are normally represented using an XML (eXtensible Markup Language) extension, for example RDF (Resource Description Framework), RDFS (RDS Schema) y OWL (Web Ontology Language) [108]. In addition, today there are tools for the editing of ontologies, for example Protégé [157]; different methodologies to build ontologies, for example, the one proposed by Noy and McGuinness [137]; and ontology repositories created in several parts of the world [28].

The application of ontologies is very diverse, and it extends to different areas of knowledge such as education, software engineering, medicine, business management, representation and organization of information, among others.

Several researchers have involved semantics in the web search, in [161] there is a proposal of a general framework based on matrices that contemplate the semantics of the terms along with the existing structural relationships in the web documents, highlighting the impact of semantics in the document ranking. In [33] the use of ontologies to generate more intelligent queries before moving them to the general web search engines is presented. And, in [210] an approach to visualize an ontology driven information retrieval system in the context of E-learning is presented.

The ALVIS Project aims to develop an open source code web search engine, with extended semantic search mechanisms. ALVIS intends to process the query in a more accurate way, while taking into account the topic and context of the search in order to refine the query and the document analysis. The development of ALVIS makes use of NLP (Natural Language Processing) architecture to enrich the documents with linguistic information. This platform is being designed to be generic in document processing [23].

The Aufare [8], Mustafa [132] and Song [172] proposals already show the use of ontologies in a feasible way to improve the traditional keyword-based web search engines, and they propose the so-called Semantic Web Search Engines or Semantic Information Retrieval Models, which work on unstructured document collections that have not been previously built over Semantic Web concepts.

3 The Proposed Model

The proposed model includes five main components, namely 1) Query expansion, 2) Search result acquisition, 3) Pre-processing, 4) Cluster construction and labeling, and 5) Visualization (see **Figure 3-1**). The first component is responsible for supporting the expansion of the user query based on the semantic relationship (extracted from ontologies that are organized in a taxonomic hierarchy) of the terms that each user has stored in their profile. The second component is responsible for search result acquisition from traditional web search engines (Google, Yahoo! and Bing). The third component is responsible for pre-processing documents and generating two representations of them, one based on vector space model and another based on frequent phrases. The fourth component is responsible for cluster construction and labeling, for which there are three heuristic algorithms that perform clustering based on vector space representation of the results, and labeling based on frequent phrases representation. The fifth component is responsible for visualization of the resulting clusters, which involves the presentation of search results organized into thematic groups (folders) and updating of the user profile based on feedback registered (relevant or not relevant).

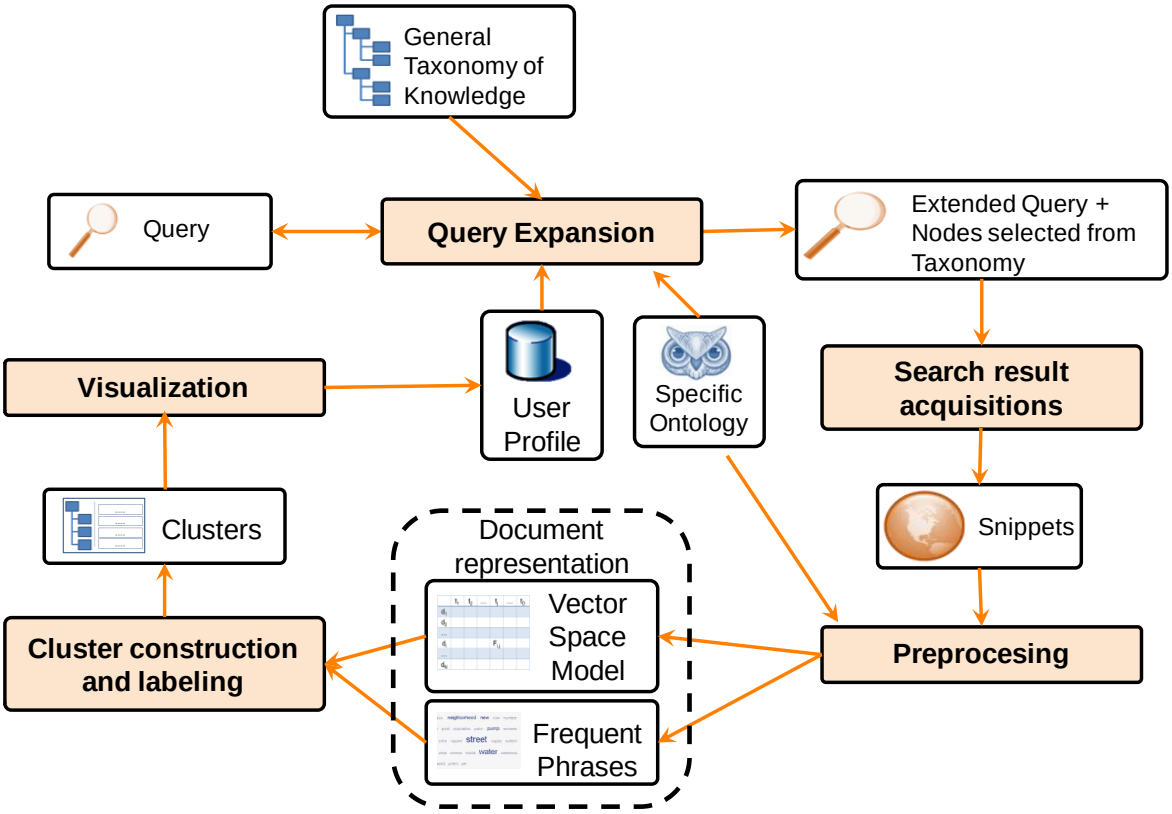
The organization of the model is based on the general architecture of a web clustering engine, but differs in the following aspects: 1) it includes a query expansion component and assigns to it the main responsibility for finding semantic relations between query terms and user profile information, 2) in the pre-processing component, two models are obtained for the representation of the results (documents) to improve the clustering process (based on the vector space model) and labeling (based on frequent phrases), and 3) visualization contemplates the possibility of including user-feedback and modifying the corresponding profile.

3.1 Query expansion

In the proposed model - a meta web search engine that clusters web documents -, a query expansion process is carried out, but here it is handled from a perspective that

gives greater importance to the semantic similarity between the terms (words), while there is also the possibility for users to give feedback (inform the model) about whether documents are relevant or not.

Figure 3-1: General components of the proposed model



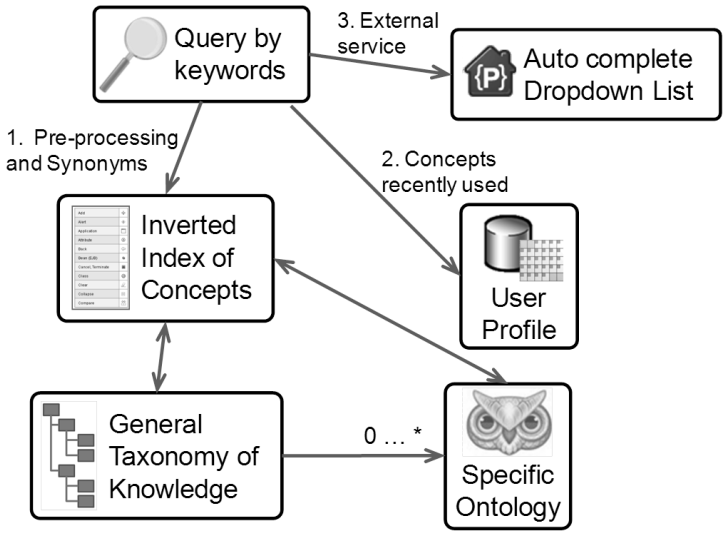
As in the reference architecture for web clustering engines (WCEs), the proposed model begins the search process with a user query (based on keywords). This query is expanded with explicit help from the user, via an auto complete option. This option is based on a dynamic drop-down list of concepts, similar to how Google operates.

The auto complete option is generated based on the list of concepts that have been relevant to the user in previous queries and that is stored in direct relationship with the specific domain ontologies and the nodes of a general taxonomy of knowledge (GTK) previously defined in the model. The three (3) steps defined for the query expansion process are shown in **Figure 3-2**. In the following, each of these steps is explained in further detail, and then the structures involved in this process are detailed.

3.1.1 Pre-processing and semantic relation

Initially the user query is taken and the special characters are removed, the words are converted to lower case, the language is detected and stop words are removed based on the respective language (English or Spanish). Language detection is conducted by way of statistical methods (e.g. creating and using an n-gram model for a set of training texts), based on a list of stop words in different languages, using an online language detection service (e.g. Google Translation API, in http://code.google.com/intl/es/apis/language/translate/overview.html or http://www.google.com/uds/samples/language/detect.html and TextCat Language Guesser in http://odur.let.rug.nl/~vannoord/TextCat/Demo/textcat.html), to name but a few).

Figure 3-2: Components of the query expansion process

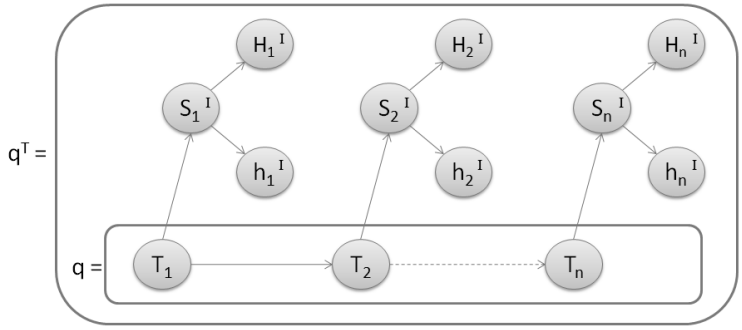


Then, based on the Inverted Index of Concepts (a structure that groups concepts and instances of the ontologies stored in the GTK) it searches for the most frequently used synonyms (S, terms in different languages used to represent the same concept), hypernyms (H, concepts in the immediately superior level in the ontology hierarchy, generalization of the concept) and hyponyms (h, concepts in the immediately inferior level in the ontology, specialization of the concept) of the terms entered by the user (see **Figure 3-3**). This is done based on sets, since there can be one or more terms that describe the concept on the same level of the ontology, and they can be expressed in multiple languages (in this case, English and Spanish). The terms are looked up in the

index based on partial concordance from the left (both English and Spanish are written from left to right, therefore the terms are given the options to auto complete the moment they are being typed) and terms in Spanish are searched with and without accents.

In short, each term is taken and they together form the vector of original terms that conform the query q = {T_1, T_2,…,T_n}, and the concepts are formed in a way that each concept C = (T, S, H, h, I). Each concept is equal to the term entered by the user and the semantically related terms that were recovered from the ontology. For each term the id (I) of the ontology - from where the values of S, H and h were obtained - is also registered, since a term can be in several specific domain ontologies.

Figure 3-3: Query expanded by each term



Synonyms (S), hypernyms (H) and hyponyms (h), each one with the id (I) from the ontology from where it was recovered.

3.1.2 Concepts related to the profile

In the previous step, a temporary extended query was obtained, but these terms must be presented to the user in an auto complete list. It is thus necessary to define the presentation order of the terms in such a way that they are better related to the user's needs. The objective of this step is defining the presentation order of the terms in relation to the user profile.

For this, the co-occurrence concept matrix is consulted for every specific ontology and each user (matrix S), and the correlation degree determined between each term and its associated terms (S, H and h) in the ontology (I) for the current user (U), arranging them in descending order of correlation (from the most related to the least).

The first element in the drop-down list to be shown to the user is obtained by linking together the original query without any preprocessing and the term (S, H or h) with the

highest correlation index. The second term is obtained in the same way - the original query and the term with second highest correlation degree - and so on until reaching the maximum number of terms to be presented on the interface (parameter from the model called AutoComplete List Size, ALS).

Given the case that a user does not have information in the S matrix, the drop down list is built giving priority to the last written terms (from left to right, due to the fact that the first terms received their auto complete list the moment they were written), and adding line by line the synonyms first, followed by the hyponyms (more specific terms) and finally the hypernyms (more general terms).

3.1.3 External service

If in the first step (pre-processing and semantic relation) the model does not find related information in the Inverted Index of Concepts (there are not enough ontologies in the model, or the user needs are in a domain that has not been modeled), it becomes necessary to turn to an external auto complete service, for example the service provided by Google (based on query registry analysis from its users, an approach centered on collaborative filtering). At this point, and as future work, the model can incorporate an approach based on automatic relevance feedback based on the Top-N recovered documents (automatic relevance feedback method based on partial documents).

The **General Taxonomy of Knowledge** (GTK) defined for the model is a hierarchically organized structure that represents human knowledge in multiple languages [193]. Other examples of these taxonomies are: Dewey Decimal[140], United States Congress Library [188], DMOZ [47], MERLOT [123], and those used by Yahoo! and Google. The components of a GTK are: a ***hierarchic relationship*** that links concepts from general to specific. The hierarchic relationship is also known as an “is a” relationship. It consists of several levels, the highest level is the most abstract one and the lower ones are the most concrete and specific. Every element in a ***level*** has to have the same approximate degree of abstraction; the root is the top of the structure, the ***node*** denotes a concept within the structure. Most of the nodes are parents (of a lower level) and sons (of a higher level); a ***leaf node*** is a node that has no child nodes; a ***brother*** is a node with the same parent node as another node and the ***path*** is the sequence of nodes that is needed to travel to reach a specific node from root.

A ***Specific Ontology*** or specific domain ontology defines concepts, relationships, functions, instances and axioms of some domain in a shared and agreed manner. In the proposed model, specific ontologies: are presented in a formal, legible and usable way by the PC via Ontology Web Language (OWL); must be designed to support multiple languages (each concept or instance can be represented in one or more languages. A collaborative editing process of these ontologies accompanies the proposed model); take into account only: Basic ideas or Concepts for which attempts have been made to formalize them; Relationships that represent the interaction and connection between the concepts in the domain and that usually form the specific domain ontology (e.g. subclass-of, part-of, exhaustive-part-of, connected-to, etc.); and Instances that represent specific objects of a concept.

Due to the fact that the query and access to the concepts, instances and relations on an ontology stored in OWL text generally constitute a time costly process, these components of the specific ontologies are represented in a structure denominated as the Inverted Index of Concepts.

Figure 3-4 shows the entities proposed to represent the general taxonomy of knowledge (GTK), the specific ontologies, the Inverted Index of Concepts and their relationships.

Both the GTK and Inverted Index of Concepts have two auto-reflective relations that allow managing the hierarchic relationships between the areas of human knowledge in the GTK or the hierarchic relationship between concepts (“is a” relationship) in the ontologies, and the relationship of synonymy in multiple languages (idioms) that may have the name of an area of knowledge on the GTK or a concept in the ontology. Finally, in the inverted index of concepts, the concepts are separated from the instances or specimens from the class through the attribute ***IICType*** (C for classes and I for instances).

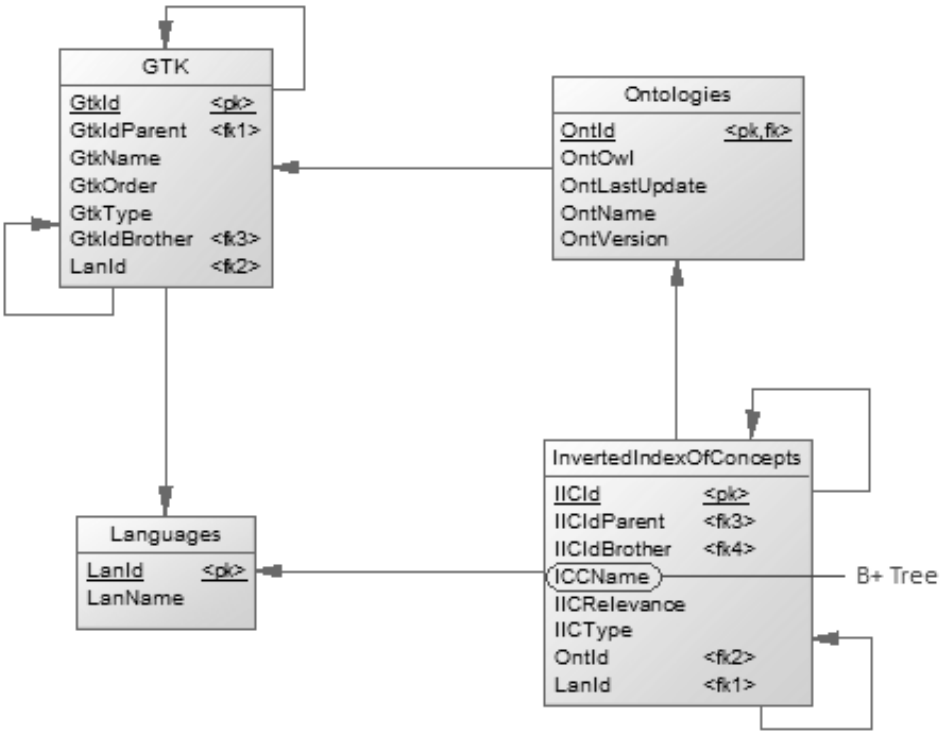
Model implementation observation: keeping in mind that the inverted index of concepts is a key structure for the performance of the model and that its implementation can be conducted for example as an ordered array, a balanced tree or a multiple path (tries) tree [64], for this model it is advised that a relational table is created and indexed using a B+ tree (balanced) using the ***IICName*** attribute and storing the tables registries (tuples) physically for that same field. Also, considering the most recent services provided by

Database Management Systems, this table can be partitioned into different hard drive locations to allow a faster access to the concepts and instances. With this it is possible to obtain a dynamic, flexible structure that provides the best possible performance.

The ***user profile*** is a fine grain structure that relates each user to the ontology concepts (and indirectly with all the nodes on the GTK) that they have inquired about (see **Figure 3-5**). For each relation of user to concept, the following are stored: the number of documents evaluated by the user (user feedback) as relevant or irrelevant (N), the number of documents that contain the concept i (n_i), the number of relevant documents (R), and the number of relevant documents that contain the concept i (r_i).

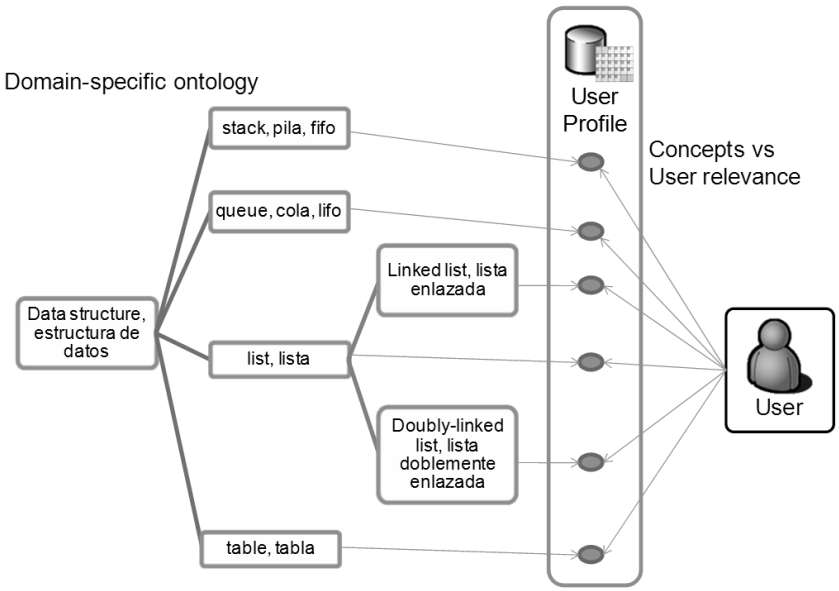
The user profile also registers the presence or absence ($cf_{i,j}$) of the concepts of each specific ontology in relation to the documents that have been evaluated by the user (it requires only the URL as identifier) (see **Figure 3-6**).

Figure 3-4: General persistence structure of the GTK, the Ontologies and the Inverted Index of Concepts



In conjunction with the two previous matrices (CUM and CDUM), it is possible to generate a co-occurrence matrix of concepts in every specific ontology and for each user. This co-occurrence matrix, called S, is calculated based on the algorithm presented in **Figure 3-7**, based on the values registered in the CUM and CDUM matrices.

Figure 3-5: User Profile Part 1, Concept-User Matrix (CUM)



O_1	c_1	c_2	...	c_j	...	c_F
U_1						
U_2						
...						
U_i				$N \mid n_i \mid R \mid r_i$		
...						
U_S						

Relation of the concepts (c_1, c_2… c_F) present in the ontologies (for example O_1) to each of the users (U_1, U_2,... U_s) from the model and matrix (CUM) resulting from the relationship.

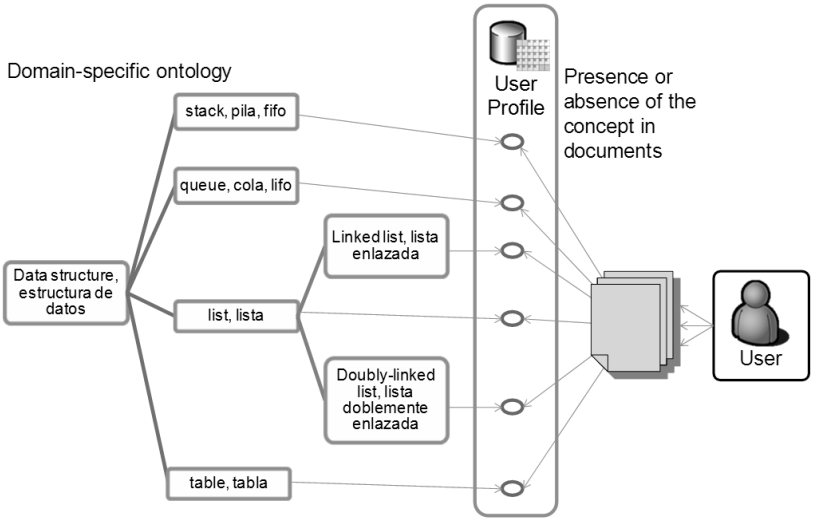
In information retrieval, the relative importance of a concept is known as the IDF value. A vast number of formulas exists for defining this value, the formula proposed by Robertson and Sparck-Jones (RSJ) [68] being one of the most referenced in the literature. For the purposes of our investigation, and through a study of limit values conducted in the RSJ

proposal, it was found not to be a viable solution for constructing the S matrix. For this reason, a new function is proposed based on the formula (3.1) [36].

$$idf_i = \left\{ \begin{array}{ll} \frac{r_i}{N} \cdots & Si \quad n_i \leq R \\ \frac{r_i * R}{n_i * N} \cdots & Si \quad n_i > R \end{array} \right\} \tag{3.1}$$

This IDF function (see **Figure 3-8**) defines the importance of a term in relation to the amount of documents evaluated by the user (N), the number of documents relevant to the user (R), the amount of documents in which the term i appears (n_i) and the number of relevant documents in which the term i appears (r_i).

Figure 3-6: User Profile Part II, Concept-Document-User Matrix (CDUM)



$U_1 \rightarrow O_1$	c_1	c_2	...	c_j	...	c_F
d_1				1		
d_2				0		
...						
d_i				$cf_{i,j}$		
...						
d_N				0		

Relationship of the concepts (c_1, c_2,…,c_F) present in the ontologies (e.g. O_1) for each of the documents (d_1, d_2,... d_N) previously evaluated by the users (e.g. U_1) of the model and matrix (CDUM) resulting from the relationship (1 for the presence of the document and 0 for its absence).

The proposed IDF function has a range of continuous values between zero and one [0, 1], zero when the term is not relevant at all, and one when it is completely relevant. The

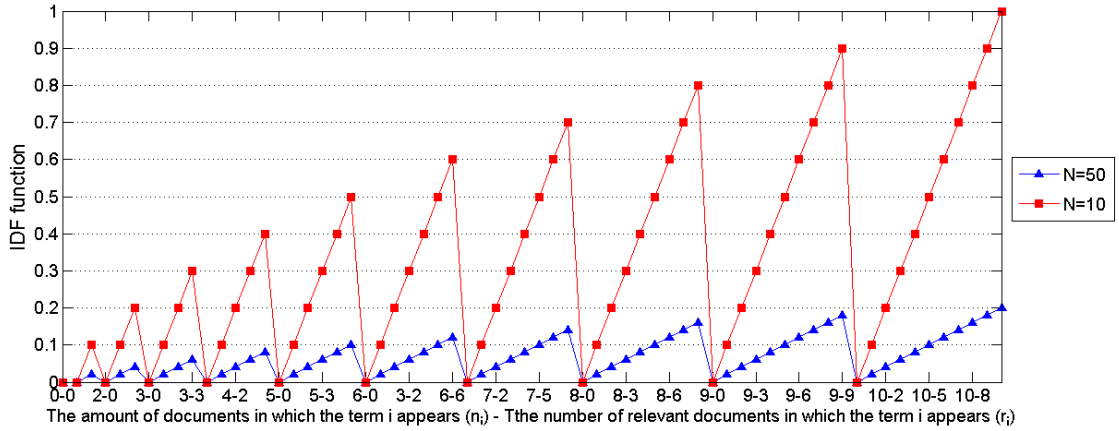
degree of relevance is given in relation to the radius of relevant documents, i.e. if there are many evaluated documents (e.g. in the graphic N=50) and among them the term appears in only a few (e.g. 6) and they are all relevant, the functions value is of 0.1. In contrast, with a smaller amount of documents (e.g. in the graphic where N=10), the obtained value would be 0.6.

Figure 3-7: Algorithm used to construct the concept co-occurrence matrix (S) in each specific ontology related to a user

```
01  For each document d Є D do
02      For each concept c_i Є d do
03          For each concept c_j Є d where j>i do
04              S_{i,j} = f_{i,j} / (f_i + f_j − f_{i,j}) * IDF_i * IDF_j
05              S_{j,i} = S_{i,j}
06          End For
07      End For
08  End For
```

Where f_i is the observed frequency of the c_i concept in the documents related to the user specific ontologies, f_j corresponds to the observed frequency of the concept c_j in the documents related to the ontology, and $f_{i,j}$ represents the observed frequency of the concepts c_i and c_j at the same time (concurrent), in the documents related to the ontology.

Figure 3-8: IDF function used to calculate the S matrix



In **Figure 3-8**, it is shown with a square marker the function with N = 10. The triangle shaped marker shows the function with N = 50. The X axis shows different values of and starting with (0-0), for example passing through (6, 3) and finishing with (10-6). On this graphic are shown values for n_i between 0 and 10, and values of r_i between 0 and 6. For

both functions, the maximum is obtained when $n_i = r_i$, in this case (6, 6), and the minimum when $r_i = 0$, not mattering the value of n_i.

This IDF function was evaluated as a tool for query expansion and was compared with Rocchio's standard formula in three query result user-feedback scenarios [36]. These scenarios involved the user profile (feedback) between queries, after conducting the query of a specific subject (several queries for the same subject) and finally without deleting the user profile [36]. The results are very promising both on closed IR test collections (CACM and LISA) and also with users from a meta-web search engine.
The users' matrix co-occurrence of concepts (S) on the specified ontology permits an ordered generation of the list of concepts that complement those employed by the users in the query expansion (auto complete) process, as explained above in section 3.1.2.

The proposed matrices allow that when a new positive (relevant) or negative evaluation of a document arrives, the update process can be done quickly, updating only the concepts from the document that are involved in the ontology and the S matrix.

When there is not enough user information or specific ontologies, and the model fails to find related concepts to expand the query, it becomes necessary to make use of an external auto complete service (e.g. Google) or include the use of pseudo-feedback. The model is customized to user needs as the system obtains more specific domain ontologies and user information, but it can also cater to requirements of general queries that are not originally included in the ontologies or in the user information (i.e. it does not have startup problems).

3.2 Search result acquisition

After completing the query expansion process, the next step is to start ***Search result acquisition***. At this point, the query is composed of:

- The ***key words*** entered by the user (those words typed by the user as well as those selected by the user from the auto complete list).
- The ***nodes selected from the GTK*** (NSGTK) that are related to the user's query, derived from the concepts selected from the drop-down list of concepts used to complete the query. They have the same ids as the ontologies.

The acquisition process conducts the result recollection from the different traditional search engines in parallel (different execution threads). Initially the model will make use of Google, Yahoo! and Bing.

3.3 Pre-processing

As the results are being returned by the traditional search engines, the ***input pre-processing*** step is conducted. This process includes stop word removal depending on the document language (English or Spanish) and thus involving language detection for the text; word stemming, according to the document language; and filtering of duplicated documents (results returned by more than one of the traditional search engines).

The observed term frequency is also calculated for each document and marked as processed. Every document is organized in a ***Term by Document Matrix*** that stores the Observed Frequency of each Term in a Document (TDM-OF). **Figure 3-9** shows a summary of this process.

In parallel (separate execution threads) and for every document previously marked as processed, a process is begun to convert from terms to concepts, as follows:

Based on the NSGTK and documents terms, a process of matching terms with the concepts they represent is carried out. The different expressions of a concept and its instances (specimens) in the different languages are taken into account. The matching process involves the accumulation of the observed frequency for the terms that have been brought together in the same concept.

The above process aims to construct a Concept by Document Matrix (CDM), recording the observed frequency of each concept in every document (CDM-OF). In this stage of the process, a thread synchronization process is performed and it only continues once all the documents have been processed. The CDM-OF matrix has to be complete to be able to continue the grouping construction and labeling process. **Figure 3-10** shows an abridgment of this process.

With the CDM-OF matrix finished, the definitive concept by document matrix (CDM) is built and the weighting of the concepts in each document is registered considering the

relative importance of each concept in the collection. This weighting is calculated based on the formula (3.2) proposed by Slaton [9, 165] - a formula commonly used for document representation in the vector space model for information retrieval - where $F_{i,j}$ is the observed frequency of the concept j in the document I, $Max(I_f)$ is the greatest observed frequency in the document i, N is the number of documents in the collection and n_j is the number of documents in which the j concept appears.

Figure 3-9: Web result acquisition and construction of the Term-Document Matrix with the observed frequency of the terms

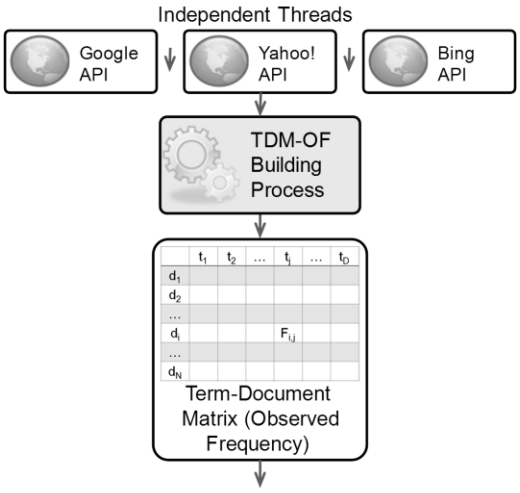
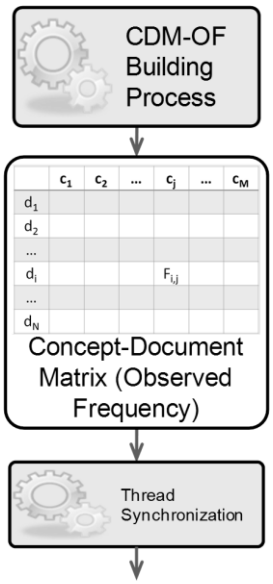


Figure 3-10: Concept-Document Matrix (Observed Frequency) Building Process



The model further contemplates the creation and use of a Frequent Concept-Document Matrix (FCDM) that is built based on the CDM-OF and the FP-Growth algorithm [13, 80, 112]. This matrix helps reduce the high dimensionality present in the document collections, and helps find a correlation of common concepts in texts of the same topic. If the model is used with this matrix, two parameters have to be set: the support and the confidence of the FP-Growth algorithm.

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{N}{n_j + 1}\right) \tag{3.2}$$

In order to facilitate the labeling process (the component that follows) based on documents represented by frequent phrases (steps 1-3 in Section 3.4.4.2), the following information is stored for each document: the position of the stop words in the original text (snippet), the plain text fully processed (final terms) and the plain text fully processed including stop words.

3.4 Cluster construction and labeling

Once the search query result acquisition is completed, the Cluster Construction and Labeling process begins. This process can be carried out by a variety of algorithms, among them Lingo [150], SHOC [203], FIHC [67], Dynamic SVD [122], FTC and HFTC [13], which perform a document grouping process centered on description (labeling) of the groups. But motivated by the poor precision reports, usefulness of the groupings, and label clarity (values of F-measure between 0.6 and 0.8 depending of data set when the goal is 1.0) for these algorithms, it was decided to make three new proposals based on population meta-heuristics.

The first of these, called **IGBHSK**, is a hybridization of the Global-best Harmony Search with the K-means algorithm (as a local optimizer or exploitation strategy). Global Best Harmony Search is an algorithm that combines harmony search with swarm techniques, specifically Particle Swarm Optimization (PSO), as a global search or exploration strategy. The second, called **WDC-MA**, is a memetic algorithm (global search or exploration technique based on roulette wheel selection, uniform crossover, multi-bit uniform mutation and rank replacement) using local optimization based on K-means. The

third and last, called **WDC-CSK**, is a memetic algorithm based on the cuckoo search heuristic, K-means, and two special operations of split and merge clusters.

These three algorithms automatically define the number of groups based on a fitness (aptitude) function that allows the comparison of different clustering solutions. This function was based on ***Balanced Bayesian Information Criterion (BBIC)*** - new, proposed in this thesis [41] - and Bayesian Information Criterion (BIC) expressed by formula (3.3) and formula (3.4) respectively.

$$BBIC = n \times Ln\left(\frac{SSE}{n * ADBC}\right) + k \times Ln(n) \tag{3.3}$$

$$BIC = n \times Ln\left(\frac{SSE}{n}\right) + k \times Ln(n) \tag{3.4}$$

$$SSE = \sum_{j=1}^{k}\sum_{i=1}^{n} P_{i,j}\left\|1 - Sim_{\cos}(x_i, z_j)\right\|^2 \tag{3.5}$$

$$ADBC = \frac{2}{k*(k-1)}\sum_{j=1}^{k-1}\sum_{i=j+1}^{k}\left\|1 - Sim_{\cos}(z_i, z_j)\right\| \tag{3.6}$$

Where n is the total number of documents, k is the number of clusters, and z_i is the center of a cluster. SSE is the sum of squared errors of the similarities of the different clusters. $P_{i,j}$ equals 1 when the document belongs to the z_j cluster, or 0 otherwise. Sim_{cos} is the cosine similarity calculated for the weighting of each cell ($W_{i,j}$), which is calculated with the formula (3.2).

$$Sim_{\cos}(d, q) = \frac{\sum_{i=1}^{D} W_{i,d} \times W_{i,q}}{\sqrt{\sum_{i=1}^{D} W^2{}_{i,d}}\sqrt{\sum_{i=1}^{D} W^2{}_{i,q}}} \tag{3.7}$$

The algorithms used may be based on term-document matrix (TDM), concept-document matrix (CDM) or frequent concept-document matrix (FCDM) as data source. Each of the three algorithms was evaluated independently before its integration into the model. The results were compared with Bisecting K-means, STC, Lingo, among others, and finally a user evaluation was conducted. In every evaluation, the results were superior to those reported in previous research.

3.4.1 IGBHSK algorithm

Figure 3-11 shows the main steps from the IGBHSK algorithm, described as follows:

01: **initialize the algorithm's parameters**: in this research, the optimization problem boils down to minimizing the BBIC (the best choice), or the BIC index that becomes the fitness function for the algorithm. IGBHSK needs some specific parameters - the Best Memory Result Size (BMRS), Maximum Execution Time (MET), and other parameters from the standard GBHS (Global Best Harmony Search): Harmony Memory Size (HMS, with typical values between 4 and 10), the Harmony Memory Consideration Rate (HMCR, with a common value of 0.95), the Pitch Adjustment Rate minimum and maximum (PAR; its values dynamically oscillate between a minimum of 0.01 and a maximum of 0.99) and the Number of Improvisations (NI) [69, 117, 118, 142].

03: **Initialize the BMR and execute the GBHSK routine**: the Best Memory Result (BMR) is a memory location that stores the best solution vectors (see **Figure 3-12**). Each row within the BMR stores the results of a call to the GBHSK (Global Best Harmony Search K-means, explained further on in this section) routine in a basic repetition loop. Each row vector inside the BMR has two parts: the centroids and the fitness value for the solution. This step is similar to evolution process carried out in different islands independent of each other and can be easily implemented in parallel.

05: **Return the Best Result**: this step is responsible for finding and returning the best result (the best of the best) from the BMR. The best result is the row or solution vector with the lowest fitness value, since the goal is to minimize f(x). Then the algorithm returns that row (solution vector) as the best document clustering solution (centroids and fitness value).

In the GBHSK routine, each generated solution vector has a different number of clusters and the fitness value (based on the results obtained from the first tests of the algorithms, Balanced BIC is used) depends on the location of the centroids in each solution vector and their amount (K value). The GBHSK routine conducts the steps presented in **Figure 3-13**, and explained in the following:

01: **Initialize the Harmony Memory**: the harmony memory (HM) is a memory location that stores all the solution vectors. Each vector is created with a random amount of centroids ($k < Kmax$), a set of centroids randomly selected from the documents (Forgy strategy [156], different from the one proposed in GBHS) and the fitness value is set to

null for the current solution. K-means (Steps 02-06 of **Figure 4-4**) is executed next (iteratively calculate document membership and recalculate centroids), and the fitness value (BBIC) is calculated for the current solution. The general structure of the HM is similar to that of the BMR. In short, HMS solution vectors are generated and fitness is calculated for each one.

Figure 3-11: IGBHSK Algorithm

```
01      Initialize the algorithm's parameters
02      For each i Є [1, BMRS] do
03              BMR[i] = GBHSK (TDM, FTDM, CDM, or FCDM) // execute the GBHSK routine
04      End For
05      Return the best result
```

Figure 3-12: Best Memory Results

$$BMR = \begin{bmatrix} Centroids_1 & Fitness_1 \\ Centroids_2 & Fitness_2 \\ \vdots & \vdots \\ Centroids_{BMRS-1} & Fitness_{BMRS-1} \\ Centroids_{BMRS} & Fitness_{BMRS} \end{bmatrix}$$

The initial number of clusters, i.e. the value of K, is randomly selected between 2 and a Kmax value, where K is a natural number and Kmax is the highest limit for the number of clusters, and it is calculated as $\lfloor\sqrt{N}+1\rfloor$ (where N is the number of documents in the matrix). If Kmax is lower than 8 (typical number of documents in a web page of Google results) and the number of documents is greater than 8, the Kmax value is set to 8, but if the number of documents is lower than 8, the Kmax value is equal to N/2. This is an adaptation of a heuristic used by many researchers in data clustering literature [168].

02: **Improvise a new harmony**: a new harmony, or solution vector, is generated. For this, a variation of the GBHS improvisation step has been carried out. Each new centroid (in this research the minimum unit of information, similar to gen, is the centroid) is created based on three rules: 1. Harmony memory consideration, 2. Pitch adjustment based on PSO, and 3. Random selection. Rule one randomly selects a solution vector (harmony) from the harmony memory and a centroid is selected. The new centroid takes its value from the selected centroid in harmony memory. In rule two, the centroid of the new harmony is modified with that of a centroid randomly selected from the best solution in the

harmony memory (in the original version of the algorithm, it is taken from a random dimension of the selected vector). In rule three, the centroid in the new improvisation is randomly selected from the input matrix of documents; this is known as the Forgy strategy, and it does not search the whole space as is done in the original GBHS. Then K-means (Steps 02-06 of **Figure 4-4**) is executed and the fitness value for the new harmony is calculated. NB: to define the k value, the same three rules were applied in a prior operation.

Figure 3-13: Steps in the GBHSK routine

```
01    Initialize the Harmony Memory
02    Improvise a new harmony: define k centroids for this solution, and for each one do:
      If U (0, 1) ≤ HMCR Then /*consider the memory*/
            K= HM[U (1… HMS)].k
            If U(0,1) ≤ PAR Then  /*Pitch adjustment based in PAR value*/
                  K= HM[best].k
            End If
      Else /*Random Selection*/
            K= U (2… Kmax)
      End If
      For i=1 to K (number of centroids) do
            If U (0, 1) ≤ HMCR Then /*consider the memory*/
                  j ~ U (1… HMS); c ~ U (1… HM[j].k)
                  N-Centroid [i] = HM [j].Centroide[c]
                  If U(0,1) ≤ PAR(iteration) Then  /*Pitch adjustment based in PAR value*/
                        c ~ U (1… HM[best].k)
                        N-Centroide [i] = HM[best].Centroide[c]
                  End If
            Else /*Random Selection: forgy */
                  Rand ~ U (1… N)
                  N-Centroide [i] = TDM[Rand], CDM[Rand], or FCDM[Rand]
            End If
      End For
      Execute K-means (Steps 02-06 of Figure 4-4) and calculate the fitness function (Balanced BIC) for
      the new harmony.
03    Update the harmony memory
04    Check the stop criteria: If NI is reached or the MET is exceeded, the iteration ends. Otherwise,
      repeat 02 and 03.
05    Return the best harmony
```

03: **Update the harmony memory**: the New Harmony replaces a selected harmony from the HM (based on the concept of Rank Selection) if its fitness value is better than the fitness of the second. Another option is to perform replace worst. Replace worst consists in replacing the worst harmony in HM if the fitness value is worse than the fitness value of the New Harmony.

Rank selection was initially proposed by Baker to eliminate high convergence presenting proportional selection methods. The selection strategy selects a harmony based on a

rank, and this rank is based on the fitness value of solutions (harmonies). Harmonies are organized based on fitness value in ascending order, and then the table of ranks is created. The table of ranks contains different probability values for each low-level heuristic. Probability values are calculated based on formula (3.8) [102, 129, 185].

$$\frac{0.25 - 1.5 * (i/(HMS - 1))}{HMS} \quad (3.8)$$

Where HMS is the harmony memory size (similar to population size in genetic algorithms) and i (between 0 and HMS-1) is the order number of each specific harmony

04: **Check stop criteria**: If the maximum number of iterations (NI) is reached or the maximum execution time (MET) is passed, the iteration ends. Otherwise, steps 02 and 03 are repeated.

05: **Return the best harmony**: find the best harmony in the HM and return (centroids and fitness value) to IGBHSK.

3.4.2 WDC-MA algorithm

WDC-MA works with agents that are used to represent the solution vectors. The algorithm uses roulette wheel to select the parents in the reproduction process. The progeny is generated by uniform crossover and multi-bit uniform mutation of the dimensions is done based on a small adjustment of the current value. Then, the progeny centroids are locally optimized using K-means, and replacement is performed based on a ranking process. The algorithm evolution process is based on only one new solution vector per iteration (generation), so it is compact. The algorithm ends when the value of the objective function or the fitness of the best solution in the population has not changed in several iterations, or the execution time reaches the defined threshold. The output of the algorithm is the best solution found during the evolution. The general structure of WDC-MA is based on islands equal to **Figure 3-11** in IGBHSK but step 03 changes to call the MAK routine and the harmony memory receives another name, population. **Figure 3-14** shows the main steps of the MAK routine.

As in IGBHSK, the problem lies in minimizing BBIC. In step 01, WDC-MA needs **initialization of the following parameters**: Population Size (PS), Mutation Rate, Minimum Bandwidth (MinB) and Maximum Bandwith (MaxB) for the mutation operations,

Number of Iterations (NI), and the Maximum Execution Time (MET) in milliseconds to stop the execution of the algorithm.

Representation and initialization: Similar to IGBHSK, in WDC-MA each agent has a different number of clusters, a centroid list and the target function value (BBIC) that depends on the location of each agent and the centroid number. Centroids of each cluster of an agent consist of D x K real numbers, where K is the number of clusters and D is the total number of concepts (dimensions). For example, in a tridimensional data point, the agent < [0.3|0.2|0.7], [0.4|0.5|0.1], [0.4|0.1|0.9], [0.0|0.8|0.7], 0.789> coded four (value of K) clusters with fitness value of 0.789. Initially, each centroid corresponds to a different document randomly selected from the TDM, CDM or FCDM matrix (Forgy strategy).

Roulette Wheel Selection: Is the same process as explained further in section 4.1.4.4 but in this case it is used to select two parents from a population of agents. Agents are selected based on fitness value (BBIC).

Figure 3-14: MAK routine

01	Randomly initialize PS agents. Each agent encodes a different number of cluster centers.
02	Execute K-means (Steps 02-06 of **Figure 4-4**) for each agent in the initial population.
03	Calculate the fitness function of each agent in the initial population using BBIC.
04	**Repeat**
05	Select a pair of parents based on roulette wheel ***selection***.
06	Generate one offspring applying **uniform *crossover*** and ***multi-bit uniform mutating*** of the parents.
07	Execute K-means (Steps 02-06 of **Figure 4-4**) for the generated offspring.
08	Calculate the fitness function of the offspring based on BBIC.
09	**Rank replacement**: the offspring competes with a rank selected agent from population
10	**Until** the stop conditions are reached. In this case, a maximum execution time (MET) or Number of Iterations (NI).
11	**Return** the best result (agent).

Uniform crossover: The size of the new offspring is calculated, generating a random value between the size of the minor parent (smaller number of centroids) and the size of the major parent (greater number of centroids). Subsequently for building each new centroid the framework generates a random number between 0 and 1. When the number is 0 the centroid is taken from parent 1 and if it is 1 the centroid is taken from parent 2. Checking at all times that the centroids are not repeated [102].

Multi-bit uniform mutation: For each of the centroids of the new individual, if a random generated number is lesser than the mutation rate parameter, a modification of the attributes takes place by adding or subtracting a value resulting from the formula (3.9).

$$(BWmax - BWmin) * Random + BWmin \quad (3.9)$$

Where BW_{max} = 0.005 y BW_{min} = 0.0005.

Rank replacement: the offspring replaces a selected agent from the population (based on the concept of Rank Selection) if its fitness value is better than the fitness of the second.

3.4.3 WDC-CSK algorithm

The Web Document Clustering based on the Cuckoo Search Algorithm (WDC-CSK) is a description-centric algorithm [25] for clustering web results, inspired by the new meta-heuristic algorithm Cuckoo Search (CS) [198]. CS is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies [200]. The algorithm combines a global/local strategy (from this point of view it is a memetic algorithm [134]) of search in the whole solution space. The K-means algorithm was used as a local strategy for improving CS global solutions. Lévy flights are replaced by two operations or methods, split and merge, which are used to promote diversity in the population and prevent the population converging too quickly to local optimal solutions. Finally, Balanced Bayesian Information Criterion or BIC can be used as a fitness function (BBIC represents the best choice) and helps the algorithm to automatically find the number of clusters. **Figure 3-15** shows the main steps executed by WDC-CSK. Detailed explanations follow of the most important steps. Steps 01, 02, 13, and 14 serve the same purpose of managing independent islands as in **Figure 3-11**.

01: **Initialize algorithm parameters**. In this research, the optimization problem lies in minimizing the BBIC or BIC criteria (index), called fitness function. WDC-CSK needs the following parameters: Maximum Number of Islands (MNI - an integer number between 1 and 5); Population Size (PS - an integer number between 5 and 10); Objective Function (OF - an enumeration value between BBIC and BIC; Probability of Abandonment (PA - a real value between 0.1 and 0.2); and finally - as an algorithm stopping criterion - Maximum Number of Nests (MNN) or Maximum Execution Time (MET, in milliseconds).

Figure 3-15: WDC-CS algorithm

```
01  Initialize algorithm parameters
02  Execute in parallel a specific number (MNI) of Islands
03      Initialize population of nests; create randomly a set of nests (population of nests) from the
        current island
04      Execute K-means (local optimizer, steps 02-06 of Figure 4-4) for each nest in population from the
        current island
05      Calculate fitness values (BBIC or BIC) according to (3.3), (3.4) for all nests in population from the
        current island
06      Repeat
07          Create a new nest using abandon, split or merge operations (methods) based on a randomly
            selected nest (current nest) from the current island
08          Execute K-means (local optimizer, steps 02-06 of Figure 4-4) for the new generated nest
09          Calculate fitness value (BBIC or BIC) according to (3.3), (3.4) for the new generated nest
10          Store best solution, if the new generated nest is better than another randomly selected nest,
            this last nest is replaced in the population for the new generated nest
11      Until stopping conditions are satisfied (MNN or MET parameter are reached)
12      Select the best nest in the population of nest from the current island
13  End on parallel execution
14  Select the best nest from all islands
```

03: **Initialize population of nest**. WDC-CSK algorithm works with nests, which are used to represent solutions. Each nest has a different number of clusters, a list of centroids, and the objective function value, based on BBIC or BIC, which depends on the centroids' location in each nest and the number of centroids. The cluster centers in the nest consist of D x k_i real numbers, where k_i is the number of clusters and D is the total number of terms (words in vocabulary). For example, in three-dimensional data, the nest < [0.5|0.1|0.8], [0.2|0.5|0.3], [0.4|0.2|0.8], [0.1|0.7|0.7], 0.819 > encodes centers of four (K value) clusters with fitness value of 0.819. Initially, each centroid corresponds to a different document randomly selected in the TDM/FTDM/CDM/FCDM matrix (Forgy strategy in the K-means algorithm). The initial number of clusters k_i, K value, is randomly calculated from 2 to Kmax similar to IGBHSK algorithm.

07: **Create a new nest**. To create a new nest (solution), the algorithm performs an operation of abandon, merge or split. Given a specific probability of PA, the algorithm creates a new nest with randomly selected centroids from the TDM/FTDM/CDM/FCDM matrix. This operation corresponds to an abandon, inspired by the situation where a cuckoo egg is discovered by the host bird. In this case a totally new nest is created to complete the population of cuckoo nests in the current island. This operation provides diversity and prevents the population nests converging too quickly.

Whereas given a specific probability (1-PA)/2, the split or merge operation is executed. These operations replace Lévy Flights in the original cuckoo search algorithm. For both operations, initially a nest is randomly selected from the current population. This nest is copied in a new nest and is called the **base nest**. In the merge operation, the two most similar centroids (measured by cosine similarity) from the base nest are selected and joined. In the split operation, the most disperse cluster is selected and divided into two clusters. The most disperse cluster is selected based on the sum of squared error value reported for each cluster, associated to each centroid in the base nest. To divide the cluster, the most different document in the selected cluster is selected and a new cluster is created with this document as a centroid.

12: **Select the best nest.** In this step the algorithm finds and selects the best solution in the population of a nest from the current island. The best nest is the nest with the lowest fitness value (minimize BBIC or BIC). This solution is then returned as the best clustering solution (centroids and fitness) from the current island.

3.4.4 Labeling

Once the best clustering solution is obtained, delivered by IGBHSK, WDC-MA or WDC-CSK, the next step is cluster labeling. In the model, two labeling methods were defined: Statistically Most Representative Concepts and (the best choice) Frequent Phrases of each cluster. Then, given that web documents can address or be related to more than one topic, a Cluster Overlapping final step was contemplated to complete the clustering and labeling process. In this, each cluster includes documents that can also belong to other clusters, if those documents are at a distance lower than or equal to the average distance from each cluster centroid to the original documents.

3.4.4.1 Statistically most representative concepts

A group is represented by a set of statistically most representative concepts (greatest observed frequency of the concepts in the group). This algorithm functions as follows:

01: **Parameter Initialization**: The maximum concept threshold and minimum concept frequency threshold are defined. The maximum concept threshold represents the maximum amount of concepts that the label of each group can include. The minimum concept frequency threshold represents a percentage of the total sum of the observed frequency of the concepts that are considered as the label for each group.

02: **Candidate label introduction**: For every group, a term-document matrix is created registering only the observed frequency of the concepts in each document of the cluster. After that, the total frequency for the concepts in the group is calculated and ordered from most frequent to least frequent.

03: **Concept pairing with its shortest terms**: Each concept is replaced with the shortest term that represents it.

04: **Label creation**: The concepts with the highest frequencies are selected, provided that they do not exceed the maximum concept threshold or the minimum concept frequency threshold. Concept representation (shortest terms) are concatenated, separated by a blank space, while repeated terms are deleted should they appear.

3.4.4.2 Frequent phrases

This step corresponds to step 2 “Frequent Phrase Extraction” in Lingo [147], but in this research the method is used for each cluster generated in the previous steps. As such, some changes were made to the original algorithm, so that it functions as follows:

01: Conversion of the representation: All documents in the current cluster are selected, one by one, and converted from character-based to word-based representation.

02: Document concatenation: In the current cluster the documents are concatenated and a new document is created with the inverted version of the concatenation.

03: Complete phrase discovery: In the current cluster the right-complete phrases and left-complete phrases are discovered and alphabetically sorted by the method and combined into a set of complete phrases.

04: Final selection: Terms and phrases located in the current cluster that exceed the Term Frequency Threshold (TFT) are selected. User query terms are removed from selected terms or phrases in order to improve the quality of the labeling process.

05: Building of the "Others” label and cluster: The algorithm uses the TFT on the documents and if some of them do not reach it, they are sent to other clusters.

06: Cluster label induction: A term-document matrix is built for the current clusters, and then cosine similarity is used to find the most similar candidate terms or phrases for the cluster (which optimizes SSE).

The labeling method based on frequent phrases reported the best results during the evaluation process. Therefore, the entire model combines two document representation models, the vector space model in most of the steps and finally, frequent phrases in the labeling step.

3.5 Visualization

Finally, the clusters and labels generated for each are presented to the user, commencing the process of Visualization. In this case, the model also features an evaluation (rating) process for each of the documents presented in each cluster. The documents evaluated as relevant or irrelevant enrich the user profile and the relationship of the concepts used in the ontologies and the NSGTK. This way, future queries can have a more relevant query expansion process, a more accurate creation of matrices (TDM-OF, CDM-OF, CDM, and FCDM), as well as better cluster building and labeling.

4 Hyper-Heuristic Framework and Web Application

This chapter describes in detail all the components of the hyper-heuristic framework developed in this research. It also provides an overview of the prototype used in the experimentation with users (web application). The hyper-heuristic framework itself can be used as another algorithm for cluster construction and labeling in the proposed model, but its main utility is as a tool for evaluating a wide set of heuristics in the web result clustering scenario.

4.1 Hyper-Heuristic Framework

WDC-HH is a hyper heuristic framework with online learning that combines pre-existing heuristics in an iterative way to search for better solutions in the search space. WDC-HH uses a high-level search strategy to intelligently control the use of a set of low-level sub-algorithms over a single optimization run of a real problem, in this case, web document clustering.

In WDC-HH, the algorithm executes several islands, each of which can evolve separately, and selects the best solution for them. In each island, a population or set of solutions is first Randomly Generated (RG), then, each of the solutions is optimized by the K-means algorithm. Next, WDC-HH tries to select based on high-level heuristics the appropriate low-level heuristic in each of the iterations to generate a new solution vector (chromosome, harmony solution, vector, particle, food source or agent), optimizes the current solution using the K-means algorithm, decides if the current solution should replace one solution vector in the population based on the fitness value of the solutions, and finally records the success or failure of the specific low-level and replacement heuristics.

Figure 4-1 shows different components of the algorithm and explains the flow of information between these components. Four (4) high-level selection heuristics: Performance-based Rank (Rank), Tabu, Random, and Performance-based Roulette Wheel (Roulette). Twenty five (25) low-level heuristics are used in WDC-HH, namely: Harmony search (HS), Improved harmony search algorithm (IH), Novel global harmony search (NH), Global-best harmony search (BH), Particle swarm optimization (PS), Differential evolution (ED), Artificial bee colony (CA), and eighteen (18) genetic algorithms generated from various selection, crossover and mutation micro heuristics. The selection micro heuristics are: Restrictive mating (RM), Roulette wheel selection (RW) and Rank (RK). The crossover micro heuristics are: One-point crossover (UP), Uniform crossover (CU) and Multi-point crossover (CM). And the micro level heuristics for mutation are: One-bit uniform mutation (MO) and Multi-bit uniform mutation (MM). At the same time each heuristic with the exception of CA is combined with four replacement methods: Rank (RR), Replace worst (WR), Restricted competition replacement (RC), and Stochastic replacement (SR), resulting in a total of 97 combined low-level and replacement heuristics (24 * 4 + 1).

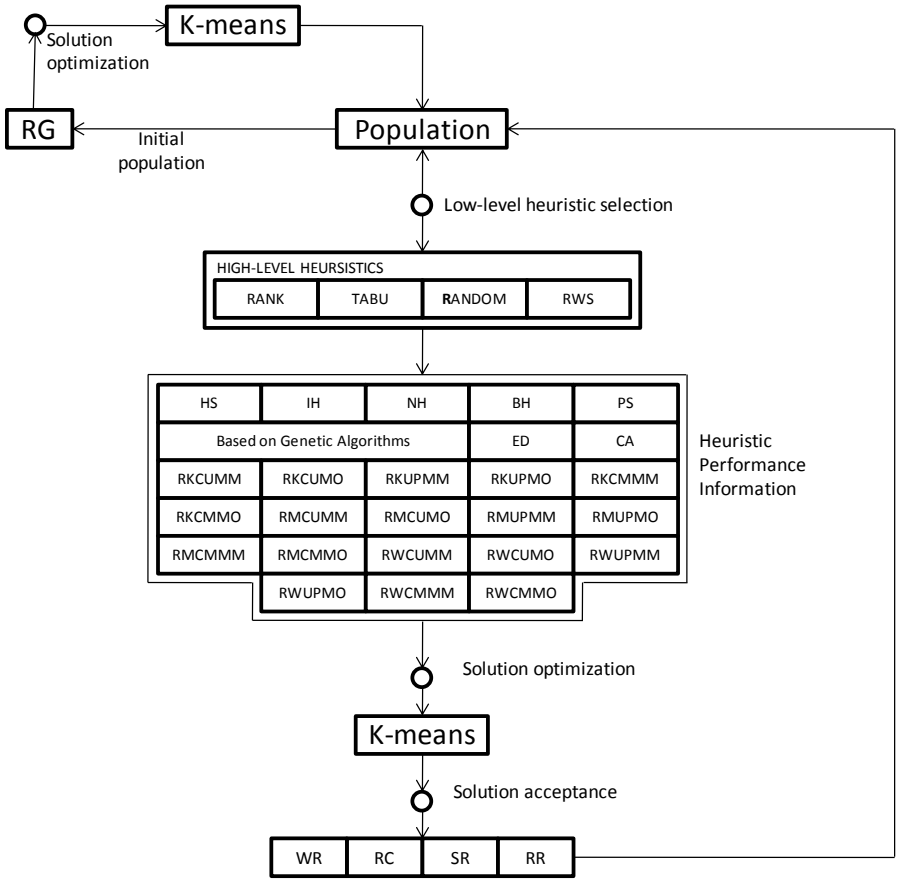
Figure 4-3 shows the relationship between the problem domain and the control domain for the hyper-heuristic environments [155]. In this case, the problem domain focuses on the creation of solutions for the problem of web document clustering, the optimization of such solutions using K-means and their evaluation based on BBIC. The solutions are created with the set of low-level heuristics that are in the control domain. These low-level heuristics are selected by high-level heuristics based on the number of successes of each low level heuristic. A low-level heuristic records a success whenever it achieves that the solution created enters the population based on a specific replacement heuristic.

4.1.1 The K-means algorithm

The K-means algorithm is the simplest and most commonly used algorithm for clustering employing a sum of squared error (SSE) criterion based on (3.5). This algorithm is popular because it finds the local minimum (or maximum) in a search space, it is easy to implement, and its time complexity is O(n). Unfortunately, the quality of the result is dependent on the initial points and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen [14, 81, 92, 115]. K-means inputs are: number of clusters (K value) and a set (table, array or collection) containing n

objects (or registers) in a D-dimensionality feature space, the formality defined by $X = \{x_1, x_2, .., x_n\}$(In our case, xi is a row vector, for implementation reasons). K-means outputs are a set containing K centers. The steps in the procedure of K-means can be summarized as shown in **Figure 4-4**.

Figure 4-1: General diagram of WDC-HH framework



In step 01, there are several approaches for selecting K initial centers [156]. Forgy [62] for example suggested selecting K instances randomly from the data set and McQueen suggested selecting the first K points in the data set as the preliminary seeds and then using an incremental strategy to update and select the real K centers of the initial solution [156]. In step 02, it is necessary to re-compute membership according to the current solution. Several similarity or distance measurements can be used. In this paper, we used cosine similarity formality defined as (3.7).

4.1.2 The fitness function

In the literature of partitional clustering, various criteria have been used to compare two or more solutions and decide which is better [91, 191]. The most popular criteria are based on the within-cluster and between-cluster scatter matrices. In this framework, the Balanced Bayesian Information Criterion (BBIC) [41] was used to define one solution as better than another one and to automatically define the number of clusters. BBIC is expressed by (3.3).

The genetic program used to generate the new fitness function (BBIC) uses a gene representation based on tree of expressions, one and two-point crossover of two parents, three kinds of mutation approaches, rank selection to generate new generation, and random re-initialization when premature convergence is detected. The fitness function of the genetic program is based on maximizing the F-measure (commonly used in information retrieval and classification tasks) extracted from a table of multiple solutions of k-means for several web clustering problems, including the “ideal” solution. In order to evaluate the F-measure (weighted formulas used by Weka [77]) the procedure detailed in section 5.1.2 was followed.

Table 4-1 shows the table that the genetic program seeks to optimize (maximize the average F-measure in all problems). A total of 50 web clustering problems based on DMOZ datasets were used. For each problem, a total of 630 solutions was created using *k*-means (30 with 2 clusters, 30 with 3 clusters, and so on until 30 with 22 clusters).

For each solution, several values were registered, namely: n (number of documents), *k* (number of clusters), SSE (based on formula (3.5)), weighted SSE (WSSE based on formula (4.1)), minimum distance between centroids (MNDBC expressed by formula (4.2)), average distance between centroids (ADBC expressed by formula (3.6)), maximum distance between centroids (MXDBC expressed by formula (4.3)), and F-measure (calculated based on current solution and ideal solution). An additional row for each problem was included; the “ideal” solution with all previously mentioned attributes.

The genetic program seeks to maximize the formula (4.4) and it can be summarized by **Figure 4-2**.

Table 4-1: Dataset to optimize (maximize F-measure) based on attributes. There are 50 problems (P) and 631 solutions (S) for each problem.

P	S	N	K	SSE	WSSE	MNDBC	ADBC	MXDBC	F-measure
1	1	121	2	71.09	36.98	0.89	0.89	0.89	57.93
	2	121	2	71.84	36.04	0.83	0.83	0.83	56.66
	…								
	630	121	22	31.26	2.17	0.54	0.94	0.54	48.75
	ideal	121	4	56.50	17.60	0.91	0.96	0.91	100.00
…									
50	1	132	2	89.65	70.83	0.93	0.93	0.93	9.48
	2	132	2	90.92	73.90	0.85	0.85	0.85	10.86
	…								
	630	132	22	37.24	2.27	0.63	0.97	0.63	54.03
	ideal	132	10	50.69	5.19	0.77	0.96	0.77	100.0

$$WSSE = \sum_{j=1}^{k} |C_j| \sum_{i=1}^{n} \left(P_{i,j} * \left(1 - SimCos(x_i, c_j) \right)^2 \right) \quad (4.1)$$

Where $|C_j|$ is the number of documents in cluster j.

$$\text{MNDBC} = Minimize_{\, i=1,..k-1, j=i+1,..k} \left(d_{i,j} \right)$$
$$Where\ d_{i,j} = 1 - SimCos(c_i, c_j) \quad (4.2)$$

$$\text{MXDBC} = Maximize_{\, i=1,..k-1, j=i+1,..k} \left(d_{i,j} \right)$$
$$Where\ d_{i,j} = 1 - SimCos(c_i, c_j) \quad (4.3)$$

$$FF = \frac{\sum_{i=1}^{P} SelectFBest(p_i, exp)}{P}$$
$$SelectFBest(p_i, exp) = Fmeasure \mid Minimize(exp\ in\ p_i\ over\ all\ S)$$

Where P is the total number of problems, p_i is the problem i, exp is the expression in genetic chromosome, Fmeasure is the value of F-measure in **Table 4-1**, and S is the list of 631 solutions for each problem. SelectFBest is a function that applies the current expression on chromosome to each solution (S), selects the solution with the minimum value for the expression and returns the F-measure for that solution. (4.4)

Figure 4-2: Pseudo-code for the genetic program

```
01  Initialize algorithm parameters.
02  Randomly initialize population, which encode expressions as a Tree.
03  Calculate fitness value for each solution in population using (4.4).
04  For Generation = 1 to MNG
05      For I = 1 to PS step by 2
06          Select chromosome I as parent1 from current population.
07          Select chromosome I+1 as parent2 from current population.
08          Generate two intermediate offspring based on parent1 and parent2 using one or
            two point crossover and include them in population.
09          Calculate fitness value for offspring using (4.4).
10      Next For
11      Apply mutation using usual gene mutation, transposition of insertion sequence (IS)
        elements or root transposition, calculate fitness value for each new solution, and include
        new solutions in current population.
12      Select PS solutions from current population to the new generation using Rank selection.
13      If Premature Convergence then Re-initialize population keeping best solution and calculate
        fitness value for each chromosome in population using (4.4).
14  Next For
15  Select and return best chromosome.
```

Initialize algorithm parameters: In this research, the optimization problem lies in maximizing the FF function expressed by (4.4). The algorithm needs the following parameters: Population Size (PS), Mutation Rate (MR), and Maximum Number of Generations (MNG) to stop the algorithm execution.

Representation and Initialization: each solution has an expression and the objective function value. The expression is a tree of different arguments ($0 for n, $1 for k, $2 for SSE, $3 for WSSE and so on) and functions (+, -, *, /, and ln for natural logarithmic).

Crossover: with 50% of probability a one point crossover is executed, otherwise a two point crossover is executed. In a one point crossover, if the length of the parent's chromosomes is the same, a random point in the first expression is defined, so that the offspring are the results of swapping parts of chromosomes at that point. In a two point crossover, two different points are randomly generated based on the parent's chromosomes, so that the offspring are the results of swapping the content of parents at those points.

Mutation: A low probability of mutation (MR) is applied to solutions in population. If a solution is selected to mutate, one of three different options can be used, namely: usual gene mutation, transposition of IS elements, or root transposition. In usual gene mutation a position in the tree (expression) is randomly selected and changed for another, also

randomly generated (arguments are changed for other arguments and functions are changed for other functions). Transposition of IS elements is done by copying a randomly selected region of genes into the head of the chromosome (into a randomly selected position). The first gene of the chromosome's head is not affected – it cannot be selected as target point. Root transposition is achieved by inserting a new chromosome root and shifting the existing one. The method first of all randomly selects a function gene in the chromosome's head. The starting point of the sequence to be put into chromosome's head is found. It then randomly selects the length of the sequence making sure that the entire sequence is located within the head. Once the starting point and the length of the sequence are known, it is copied into the head of the chromosome, shifting elements already existing.

The genetic algorithm was executed and several expressions obtained an average F-measure of 90%. Several expressions included the relation between SSE and ADBC involved in a natural logarithmic function. With this information, an adaptation of BIC called Balanced BIC was proposed. The Balanced BIC (BBIC) is expressed by (3.3).

Figure 4-3: Classification of HH (adapted from [76])

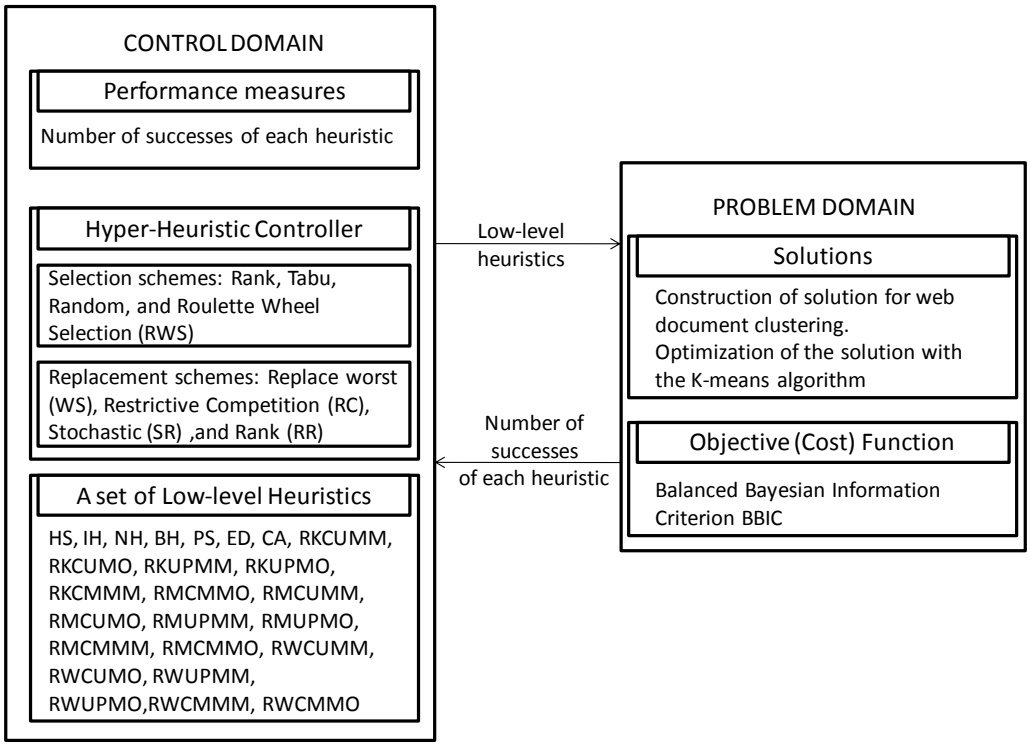


Figure 4-4: The K-means algorithm

```
01    Select an Initial Partition (k centers)
      Repeat
02          Re-compute Membership
03          Update Centers
04    Until (Stop Criterion)
05    Return Solution
```

4.1.3 WDC-HH from an algorithm point of view

Figure 4-5 shows a description of each process executed by de WDC-HH framework. Details of this process include:

01: **Initialize algorithm parameters**: In this research, the optimization problem lies in minimizing the BBIC criterion, called fitness function. WDC-HH needs the following parameters: Best Memory Results Size (BMRS), Population Size (PS), Number of Iterations (NI) or Maximum Execution Time (MET), and other specific parameters from all low-level heuristics (typical values for each parameter are shown in parenthesis): Harmony Memory Considering Rate (HMCR, 0.95), Pitch Adjusting Rate (PAR, between 0.3 and 0.99), Selection Group Size (SGS, 25% of population size) for restricted mating, Replacement Group Size (RGS, 25% of population size) for restricted competition replacement, Mutation Rate (MR, between 0.2% and 0.5%), Minimum Bandwidth (MinB, 0.0005), Maximum Bandwidth (MaxB, 005) for mutation operation. Two parameters for ABC heuristic: Probability Employed Bee (PEB, 10%) and Exploitation Probability Random (EPR, 40%). Mutation Factor (FED, 50%) and Recombination Probability (CR, 20%) in differential evolution heuristic. And three parameters for PSO heuristic: Social scaling parameter (C2, 1.49445) and Particle Inertia Minimum and Maximum (Wmax and Wmin) [69, 117, 118, 142].

02: **Document preprocessing**: Initially, Lucene (http://lucene.apache.org) is used at a document pre-processing stage. The pre-processing stage includes: tokenize, lower case filtering, stop word removal, text stemming based on Porter's algorithm [9], delete documents with empty preprocessed content (snippet) and the building of the Term-Document Matrix (TDM). The TDM matrix is the most widely-used structure for document representation in IR, and is based on the vector space model [9, 78]. In this model, the documents are designed as bags of words; the document collection is represented by a matrix of N-documents (as rows) by D-terms (as columns). Each document is represented by a vector of normalized frequency term (tf_i) by the document inverse frequency for that

term, in what is known as TF-IDF value (expressed by equation (3.2)), and the cosine similarity (see equation (3.7)) is used for measuring the degree of similarity between two documents, between a document and the user's query, or between a document and a cluster centroid.

03: **Initialize the best memory results and call the HHK routine**: The Best Memory Results (BMR) is a memory structure where the best solution vectors of each island are stored (see **Figure 4-6**). Each row in BMR stores the result of one call to the Hyper-Heuristic K-means (HHK) routine, in a basic cycle. Each row vector in BMR has four parts: The centroids, the low-level heuristic used to generate it, the replace heuristic used to enter in population, and the fitness value of that vector. The number of centroids in each row vector in BMR can be different.

Figure 4-5: The WDC-HH algorithm and the HHK routine

WDC-HH algorithm:

01	Initialize algorithm parameters
02	Document preprocessing: Tokenize, stop word removal, text stemming based on Porter's algorithm, delete documents with empty preprocessed content, and Term-Document matrix (TDM) building
03	Initialize the BMR and call the HHK routine **For each** i Є [1, BMRS] **do** BMR[i] = HHK (TDM) **Next-for**
04	Select the best result
05	Assign labels to clusters
06	Overlap clusters

HHK routine:

01	Initialize Population: Define centroids (forgy strategy), Execute K-means (Steps 02-06 of **Figure 4-4**) and Calculate fitness (BBIC) for each solution vector generated in population. A total set of PS solution vectors is created.
02	**Repeat**
03	Generate a new solution vector: Using performance-based rank selection tabu selection, random selection, or performance-based roulette wheel selection to select the low-level heuristics from HS, IH, NH, BH, PS, ED, CA or the other 18 genetic-based low-level heuristics.
04	Execute K-means (Steps 02-06 of **Figure 4-4**) and Calculate fitness (BBIC) for the new solution vector
05	Update population: The solution competes to one solution in population for entering in the population. There are four alternative acceptance (replace) strategies: Replace the worst, stochastic replacement, rank replacement and restricted competition replacement. ABC is the unique low-level heuristic that does not use these replace strategies, because ABD has its own replace strategy.
06	**Until** stopping conditions are satisfied: for example, the maximum number of iterations (NI) is satisfied or the maximum execution time (MET) is reached.
07	Select the best solution in population and return to WDC-HH

Figure 4-6: Best Memory Results

$$BMR = \begin{bmatrix} Centroids_1 & lowHeuristic_1 & repHeuristic_1 & Fitness_1 \\ Centroids_2 & lowHeuristic_2 & repHeuristic_2 & Fitness_2 \\ \vdots & \vdots & \vdots & \vdots \\ Centroids_{BMRS-1} & lowHeuristic_{BMRS-1} & repHeuristic_{BMRS-1} & Fitness_{BMRS-1} \\ Centroids_{BMRS} & lowHeuristic_{BMRS} & repHeuristic_{BMRS} & Fitness_{BMRS} \end{bmatrix}$$

04: **Select the best result**: Find and select the best result from the Best Memory Results (BMR). The best result is the row with the lowest fitness value (minimize *f(x)*). Then return this row as the best clustering solution (centroids and fitness).

05: **Assign labels to clusters**: the WDC-HH framework uses a frequent phrases approach for labeling each cluster. This step corresponds with step 2 called “Frequent Phrase Extraction” in Lingo [150] (with some modifications). See section 3.4.4.2 for more details.

06: **Overlap clusters**: Finally, each cluster includes documents that fall into other clusters too, if these documents are at a distance less than or equal to the average distance of the cluster.

In HHK routine, each solution vector used has a different number of clusters (centroids), and the objective function (BBIC) depends on the centroid location in each solution vector and the number of centroids (K value).

In step 01: **Initialize Population** of HHK routine, the population is a memory structure where all the solution vectors are stored. The general structure of the population is similar to BMR in **Figure 4-6**. Each solution vector is created with a random number of centroids (k < Kmax), a random initial location for each centroid, a value of RG for describing that this solution was generated with a random generation strategy, a value of NR for describing that this solution entered the population without any specific replacement strategy, and the fitness value for this solution. The cluster centers in the solution consist of k_i x D real numbers, where k_i is the number of clusters and D is the total number of terms (words in vocabulary). The Initial centroids are selected randomly from the original

data set. Next, the **K-means algorithm** (steps 02 to 06 of **Figure 4-4**) is executed and then fitness value for this solution calculated.

In summary, PS solution vectors are generated and then the fitness value for each vector is calculated. Initially, each centroid corresponds to a different document randomly selected in the TDM matrix (Forgy strategy in the K-means algorithm). The initial number of clusters k_i, K value, is randomly calculated from 2 to Kmax (inclusive), where K is a natural number and Kmax is the upper limit of the number of clusters and is taken to be $\sqrt{N} + 1$, (where N is the total number of documents in the TDM matrix, but this value cannot be less than eight), which is an adapted rule of thumb used in the clustering literature by many researchers.

In the evolution process, solution vectors change all or most of the original solution vectors in the population. Therefore, it is normal to find vectors with different values of low-level and replacement heuristics. For example, a solution could be <[0.4|0.2|0.7], [0.2|0.3|0.1], [0.1|0.4|0.5], [0.7|0.7|0.7], BH, RK, 0.193>. This sample solution encodes centers of four (K value) clusters in a three dimensional space with a fitness value of 0.193, generated using Global-best Harmony Search low level-heuristic (BH) and finally, this solution uses rank replacement strategy (RK) to enter the population.

4.1.4 High-level heuristics

In step 03: **Generate a new solution vector** of the HHK routine, a new solution vector (centroids) is generated based on low-level heuristics using one of the high-level heuristics, namely: Performance-based Rank selection, Tabu selection, Random selection or Performance-based Roulette Wheel selection.

4.1.4.1 Performance-based rank selection

Rank selection was initially proposed by Baker to eliminate the high convergence presented by proportional selection methods. The selection strategy selects a new low-level heuristic based on a ranking, and this ranking is based on past heuristic success. The heuristics are organized based on the number of successes in descending order, and then the table of rankings is created. The table of rankings contains different probability values for each low-level heuristic. Probability values are calculated based on formula (4.5) [102, 128, 185].

$$\frac{0.25 - 1.5 * (i * 1.0/n - 1)}{n} \quad (4.5)$$

Where n is the total number of low-level heuristics and i (between 0 and n-1) is the order number of each specific low-level heuristic.

4.1.4.2 Tabu selection

The original version of this heuristic avoids exploring previously visited areas using a tabu list [19]. In this research the tabu list has a maximum length of 18% (approximating the largest integer value) of all the low-level heuristics being run. In this way, if only two combined low and replacement heuristics are being run, this selection heuristic behaves as an alternator, selecting first one and then the other. Low level heuristics enter the tabu list when a maximum number of executions have been run. **Figure 4-7** shows a general description of tabu selection in the WDC-HH framework.

4.1.4.3 Random selection

The random selection strategy randomly uses a low-level heuristic to generate a new solution vector. No memory of previous good performance is retained and no learning is attempted.

Figure 4-7: Tabu Selection

```
Initialize tabu parameters: Define the tabu list size and the maximum number of executions.
Repeat
        index ~ U (1…number of heuristics) //Select one of the heuristics that is not in the tabu list
Until tabu list does not contain the heuristic index
Heuristic[index].NumberOfExecutions++
if (tabulist is full) then
        Tabulist[0].delete //Delete the first heuristic in tabu list based on a FIFO behavior.
End if
//If exceed the number of visits
If (Heuristic[index]. NumberOfExecutions > Maximum Number of Executions) then
        Tabulist.Add(index) //Add the heuristic index to tabu list
End if
```

4.1.4.4 Performance-based roulette wheel selection

This strategy was inspired by "Roulette Wheel" and "Stochastic Universal" Sampling [128]. The selection strategy selects a new low-level heuristic based on past heuristic success. This is achieved using a roulette wheel-based selection operator which ensures that low-level heuristics that previously performed well (high success rate) have a higher probability of being selected again. The probability $P_{i,j}$ of selecting low-level heuristic i for creating a new solution at iteration j can be calculated based on Laplace estimator [194], as in formula (4.6).

$$P_{i,j} = \frac{NSH_i + 1}{TE + NH} \quad (4.6)$$

Where NSH_i is the number of times when the heuristic i has been successfully, TE is the total number of times when all heuristics have been successful, and NH is the total number of heuristics being tested.

4.1.5 Low-level heuristics

In step 03: **Generate a new solution vector** of the HHK routine, a new solution vector (centroids) is generated based on the low-level heuristic selected. There are 25 low-level heuristics: harmony search, improved harmony search, novel global harmony search, global-best harmony search, eighteen genetic algorithm variations, particle swarm optimization, artificial bee colony, and differential evolution.

4.1.5.1 Harmony search (HS)

HS is a meta-heuristic algorithm mimicking the improvisation process of musicians (where music players improvise the pitches of their instruments to obtain better harmony) [69, 117]. HS has been successfully applied to many optimization problems: travelling salesman problem, power economic dispatch, and for web document clustering [117], among others. **Figure 4-8** shows a general description of the improvisation step of HS used in the HHK routine as a low-level heuristic.

The HMCR and PAR parameters of HS help the method in searching for globally and locally improved solutions, respectively. PAR has a profound effect on the performance of the HS algorithm. Thus, the fine tuning of this parameter is very important (see [142] for details).

Figure 4-8: Improvisation steps of HS algorithm in HHK routine

```
For i=1 to K (number of centroids) do
        If U (0, 1) ≤ HMCR then
        Begin /*memory consideration*/
                j ~ U (1… PS);
                p ~ U (1… Population [j].K) //selection of centroid
                New [i] = Population [j].Centroid[p]
                If U(0,1) ≤ PAR then
                Begin /*pitch adjustment*/
                        For j=1 to D (number of dimensions) do
                                New [i] = New[i] ± BW
                        Next For
                End if
        Else /*random selection – forgy strategy*/
                j ~ U (1… N);
                New [i] = TDM[j]
        End if
Next for
```

4.1.5.2 Improved harmony search algorithm (IH)

IH uses in general the same logic as the HS algorithm [118], see **Figure 4-8**. The key difference between IHS and the traditional HS method is in the method of adjusting the PAR (Pitch Adjustment Rate) and BW (Bandwidth) parameters in each iteration. PAR is defined based on (4.7) and BW based on (4.8).

$$PAR(iteration) = PAR_{min} + \frac{(PAR_{max} - PAR_{min})}{NI} \times iteration \quad (4.7)$$

Where PAR is the pitch adjustment rate for each iteration, PAR_{min} is the minimum pitch adjustment rate, PAR_{max} is the maximum pitch adjustment rate, NI is the maximum number of iterations, and iteration is the current iteration number.

$$bw(iteration) = bw_{max} \exp(c \times iteration) \text{ and } c = \frac{Ln\left(\frac{bw_{min}}{bw_{max}}\right)}{NI} \quad (4.8)$$

Where bw(iteration) is the bandwidth for each iteration, bw_{min} is the minimum bandwidth, and bw_{max} is the maximum bandwidth.

4.1.5.3 A novel global harmony search algorithm (NH)

NH [212] is inspired by the swarm intelligence of a particle swarm. NH includes two important operations: position updating and genetic mutation with a small probability. **Figure 4-9** shows a general description of the improvisation step of NH used in HHK routine as a low-level heuristic.

Figure 4-9: Improvisation steps of NH algorithm in HHK routine

```
For i=1 to K (number of centroids) do
    Best ~ U (1 … Population [BestSolution].k);
    Worst ~ U (1 … Population [WorstSolution].k);
    For j=1 to D (number of dimensions) do
        high = Population[BestSolution].Centroid[Best][j]
        low = Population[WorstSolution].Centroid[Worst][j]
        X = 2 * high - low
        If X < 0 then X =0
        r ~ U (0, 1)
        New [i][j] = low +     r * (X - low)
        If U(0,1) ≤ PM then
            p ~ U (1… N);
            New [i][j] = TDM[j][p]
        End if
    Next for
Next for
```

4.1.5.4 Global-best harmony search algorithm (BH)

Global-Best Harmony Search [142] is a new variant of HS. BH is inspired by the concept of swarm intelligence as proposed in Particle Swarm Optimization [100]. **Figure 4-10** shows a general description of the improvisation step of BH used in HHK routine as a low-level heuristic.

4.1.5.5 Particle swarm optimization (PS)

PS is a population-based, co-operative search meta-heuristic [54]. In PS, a potential solution to an optimization problem is treated as a bird in a flock, without quality and volume, and referred to as a particle, coexisting and evolving simultaneously based on knowledge shared with neighboring particles. While flying through the problem search space, each particle modifies its velocity to find a better solution (position) by applying its own flying experience and the experience of neighboring particles. Particles update their positions and velocities based on (4.9) and (4.10), respectively.

Figure 4-10: Improvisation steps of BH algorithm in HHK routine

```
For i=1 to K (number of centroids) do
    If U (0, 1) ≤ HMCR then
    Begin /*memory consideration*/
        j ~ U (1… PS);
        p ~ U (1… Population [j].K) //selection of centroide
        New [i] = Population [j].Centroid[p]
        If U(0,1) ≤ PAR then
        Begin /*Particle Swarm Optimization*/
            p ~ U (1… Population [Best].k); // Best is the position of the best solution vector in population
            New [i] = Population [Best].Centroid[p]
        End if
    Else /*random selection – forgy strategy*/
        j ~ U (1… N);
        New [i] = TDM[j]
    End if
Next for
```

$$v_{t+1}^{i} = \omega_t * v_t^i + c_1 * R_1 * \left(p_t^i - x_t^i\right) + c_2 * R_2 * (p_t^g - x_t^i) \tag{4.9}$$

$$x_{t+1}^{i} = x_t^i + v_{t+1}^i \tag{4.10}$$

Where x_t^i represents the current position of particle i in solution space and subscript t indicates an iteration count, p_t^i is the best-found position of particle i up to iteration count t and represents the cognitive contribution to the search velocity v_t^i. p_t^g is the global best-found position among all particles in the swarm up to iteration count t and forms the social contribution to the velocity vector, R_1 and R_2 are random numbers uniformly distributed in the interval (0,1), and c_1 and c_2 are the cognitive and social scaling parameters, respectively. ω_t is the particle inertia, which is reduced dynamically to decrease the search area in a gradual fashion [15]. The variable ω_t is updated by (4.11).

$$\omega_t = (\omega_{max} - \omega_{min}) * \frac{(NI - in)}{NI} + \omega_{min} \tag{4.11}$$

Where ω_{max} and ω_{min} denote the maximum and minimum of ω_t respectively, NI is the maximum number of iterations, and in is the current iteration number.

In this research, c_1 is equal to zero because solution vectors do not evolve. They are replaced by new, better solutions, and each one of them has the best possible value in its neighbourhood. The new solution is based on one solution vector from current population. **Figure 4-11** shows a general description of the PSO algorithm used in HHK routine as a low-level heuristic.

Figure 4-11: PS algorithm in HHR routine

```
/* select particle */
i ~ U (1… PS);
New = Population [i]
For i=1 to K (number of centroids) do
    p ~ U (1… Population [Best].k);
    For j=1 to D (number of dimensions) do
        r ~ U (0… 1)
        velocity = wt * velocity + C2 * r *
        ( Population[Best].Centroid[p][j] - New[i][j])
        New [i][j] = New[i][j] + velocity
    Next for
Next for
```

4.1.5.6 Differential evolution (ED)

In ED [141], three different parents are initially randomly selected from the population. The calculation of the number of centroids to be generated in the new solution is defined by formula (4.12), where Xri with i= 1, 2 and 3 is the number of centroids of each parent and FED is a real constant for mutation between [0,2], which controls the amplification in the differential variation (Xr2 - Xr3). Next, it is ensured that the number of groups is greater than or equal to 2.

$$|Xr1 + FED(Xr2 - Xr3)| \tag{4.12}$$

A fourth solution vector (different to the parents) is then selected from the population, called the base vector. Next, to define each one of the centroids, a random number between 0 and 1 is generated. If the number is less than the probability of reproduction or recombination (CR parameter) the centroid is taken from a vector randomly selected from the centroid base. Otherwise, a random centroid is taken from each of the parents and the attributes of the centroid are calculated based on formula (4.12).

4.1.5.7 Artificial bee colony (CA)

Inspired by [151], a new random number is generated between 0 and 1. If this value is less than or equal to 0.1, a new individual is created using random centroids, similar to the work carried out by an employed bee. If the performance of the new individual is

better than the performance of the worst individual in the population, the new individual replaces the worst in the population.

If the random number generated is between (0.1, 0.4], an individual (solution vector) is chosen randomly from the population. A perturbation is applied to this individual, creating some centroids at random, if the new individual has a better performance than the original individual, the new individual replaces the original.

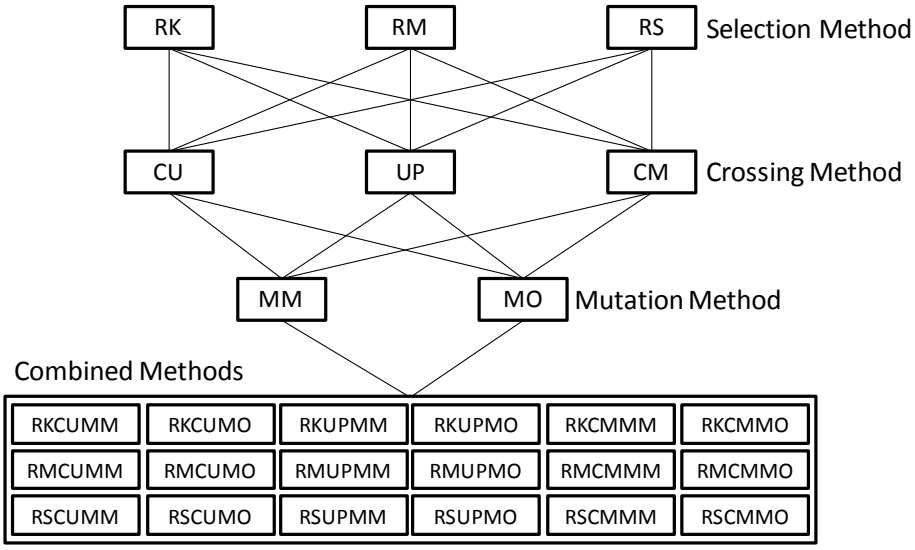
If the randomly generated number is greater than 0.4, one of the best individual is chosen from the population, based on the roulette wheel method. An exploitation is applied to this individual, creating some centroids at random. If the performance of the new individual is better than the performance of the original individual, the new one replaces the original one.

4.1.5.8 Heuristics based on genetic algorithms

Genetic heuristics result from the combination of different selection, crossover and mutation schemes widely used in the literature. The selection schemes were: Restrictive mating (RM), Roulette wheel selection (RW) and Rank selection (RK). The crossover schemes were: One-point crossover (UP), Multi-point crossover (CM) and Uniform crossover (CU). During crossover, the cluster centers (centroids) are considered to be indivisible, so, crossover points can only lie in between two cluster centers. After crossover, with a low probability (Mutation Rate, MR) a mutation operation is applied to the offspring. The mutation schemes were: one-bit uniform mutation (MO) and multi-bit uniform mutation (MM). Mutation between Minimum Bandwidth (MinB) and Maximum Bandwidth (MaxB) (similar to Harmony Search Algorithm [69]) is applied to the chosen cluster dimension/attribute/term [x = x ± Random (MinB, MaxB)]. When the mutation operation generates a value that reaches data boundaries, the mutation value is applied in the opposite way (mirror). **Figure 4-12** shows the eighteen low-level genetic heuristics created.

Rank Selection (RK): Applies the same process explained in section 4.1.4.1 to select the two parents of the new individual, but performance is based on the fitness value of the individuals from the population rather than the number of successes of the heuristics.

Figure 4-12: Heuristics based on genetic algorithms



Restrictive Mating Selection (RM): One parent p1 is randomly selected from the population. Its mate p2 is chosen from a selection group (SGS solution vectors randomly selected from current population) with the most similar number of clusters as for p1. If this results in a group with more than one candidate solution, the similarity of cluster centers (based on cosine similarity) is further used to select the most similar one.

Roulette wheel selection (RW): Applies the same process explained in section 4.1.4.4 to select the two parents of the new individual, but performance is based on the fitness value of the individuals from the population rather than the number of successes of the heuristics.

Traditional crossover schemes produce two offspring, but in the proposed framework just one is generated. The framework generates a random number between 0 and 1 and selects the left offspring if the generated number is less than 0.5 otherwise the framework generates the right offspring.

One-point crossover: first a random cutting point is chosen for both parents. The left offspring will be comprised of the centroids to the left of the first parent and the centroids to the right of second parent using the cutting point as reference, and the right offspring is built with the centroids to the left of the second parent and the right centroids of the first parent using as reference the cutting point [102].

Multi-point crossover: The total crossing points between 1 and the smallest number of the centroid of the two parents are defined. A segment is defined as a set of centroids between two adjacent crossing points. The left offspring is formed by centroids in the left segments of the first parent (p1) and centroids in the right segments of the second parent (p2). The right offspring is formed by centroids in the right segments of p1 and in the left segments of p2 [72].

Uniform crossover: The size of the new offspring is calculated generating a random value between the size of the lesser parent (smaller number of centroids) and the size of the greater parent (greater number of centroids). Subsequently, to build each new centroid the framework generates a random number between 0 and 1. When the number is 0 the centroid is taken from parent 1 and if it is 1 the centroid is taken from parent 2, checking at all times that the centroids are not repeated [102].

One-bit uniform mutation: A centroid of the new individual is randomly selected and one of its attributes modified by adding or subtracting a value that is calculated by formula (4.13), taking into account that the probability of mutation of the attributes is 0.5%

$$(BWmax - BWmin) * RandomDouble + BWmin \tag{4.13}$$

Where $BW_{max} = 0.005$ y $BW_{min} = 0.0005$.

Multi-bit uniform mutation: For each of the centroids of the new individual, a modification of the attributes takes place by adding or subtracting a value resulting from formula (4.13). The probability of an attribute change is 0.05%.

4.1.6 Replacement heuristics

In step 05: **Update population** of the HHK routine. The solution competes with one solution in the population in order to gain entry to the population. There are four alternative replacement (acceptance) strategies: Replace worst, Restricted competition replacement, Stochastic Replacement and Rank Replacement.

4.1.6.1 Replace worst (WR)

In this case, the new solution competes with the worst solution in the population. If the fitness of the new solution is better than its paired solution, then the paired solution is replaced by the new one.

4.1.6.2 Restricted competition replacement (RC)

The new solution is compared with each solution that has the same number of clusters as the new solution in a competition group (RGS solution vectors randomly selected from the current population), and paired with the one with the most similar cluster centers (cosine distance) if such exists; otherwise, it is paired with a solution with the lowest fitness. If the fitness of the new solution is better than its paired solution, the latter is replaced [40].

The extended restricted competition replacement is mostly used to balance competition during replacement among solutions with different numbers of clusters. An appropriate value for RGS should be set to allow both thorough exploration of the search space with the same number of clusters and competition among solutions with different numbers of clusters.

4.1.6.3 Stochastic replacement (SR)

It selects the best individual from the population based on performance and compares it with the new individual generated. If the new individual has a better fitness than the best it replaces the best in the population. Otherwise formula (4.14) is applied to decide whether or not the new individual enters the population [76]. In this method, a new random number is generated between 0 and 1. If it is less than the result of applying formula (4.14), the new individual replaces the worst individual in the population.

$$e^{-\frac{10*ActualIteration}{NI}} \tag{4.14}$$

4.1.6.4 Rank replacement (RR)

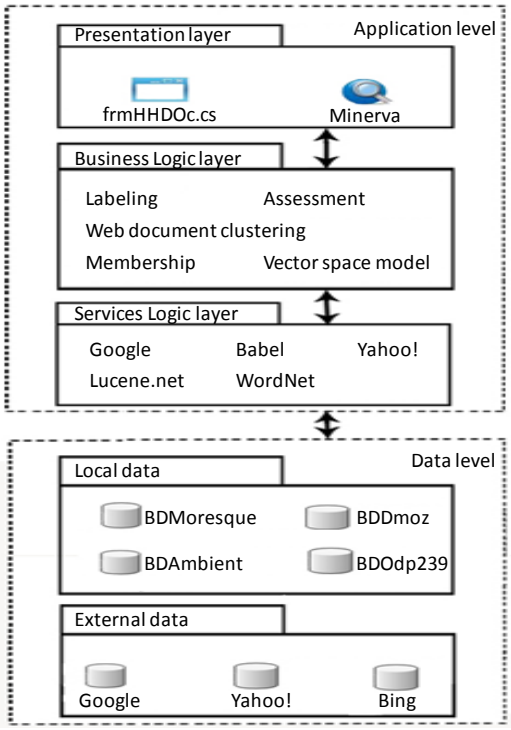
The same process explained in section 4.1.4.4 is applied, but here it works on the fitness of individuals in the population rather than the number of successes of the heuristics. It selects the individual to replace; if the performance of the new individual is better than the performance of the selected individual, the new individual replaces the selected individual in the population.

4.2 Framework implementation

4.2.1 General architecture

The framework is implemented based on a multi-layer architecture among which are: 1) presentation layer, 2) business logic layer, 3) service logic layer and 4) data layer, see **Figure 4-13**. Each layer has a particular function, the presentation layer called Laboratory is responsible for defining the connection to the databases that are part of the data layer: BDDmoz, BDMoresque, BDODP-239 and BDAmbient, requesting information from the test that is going to be run, and finally presenting the results of the execution of such evaluations on a form and saving them in Excel files. If the presentation layer that the user is running is Minerva, this application will allow entering the user query on the web, setting the search options and displaying the results in clusters with the metaphor of folders.

Figure 4-13: General architecture of the framework



The business logic layer is responsible for the web document clustering process spanning such processes as vector space model generation (TDM matrix), use of K-means

algorithm to optimize solutions, group labeling, and calculating evaluation measures, e.g. number of groups (k), F-measure, precision, recall, SSL_k measure, etc.

In the service logic layer are services such as Google, Yahoo! and Bing, used to obtain the results of web searches; Babel that allows recognition of the document language; Lucene.Net that facilitates the process of creating the TDM matrix with the tokenization tasks, removal of stop words, stemming in English and Spanish, among others; and WordNet that provides a lexical database of English; among others.

4.2.1.1 Overview of classes

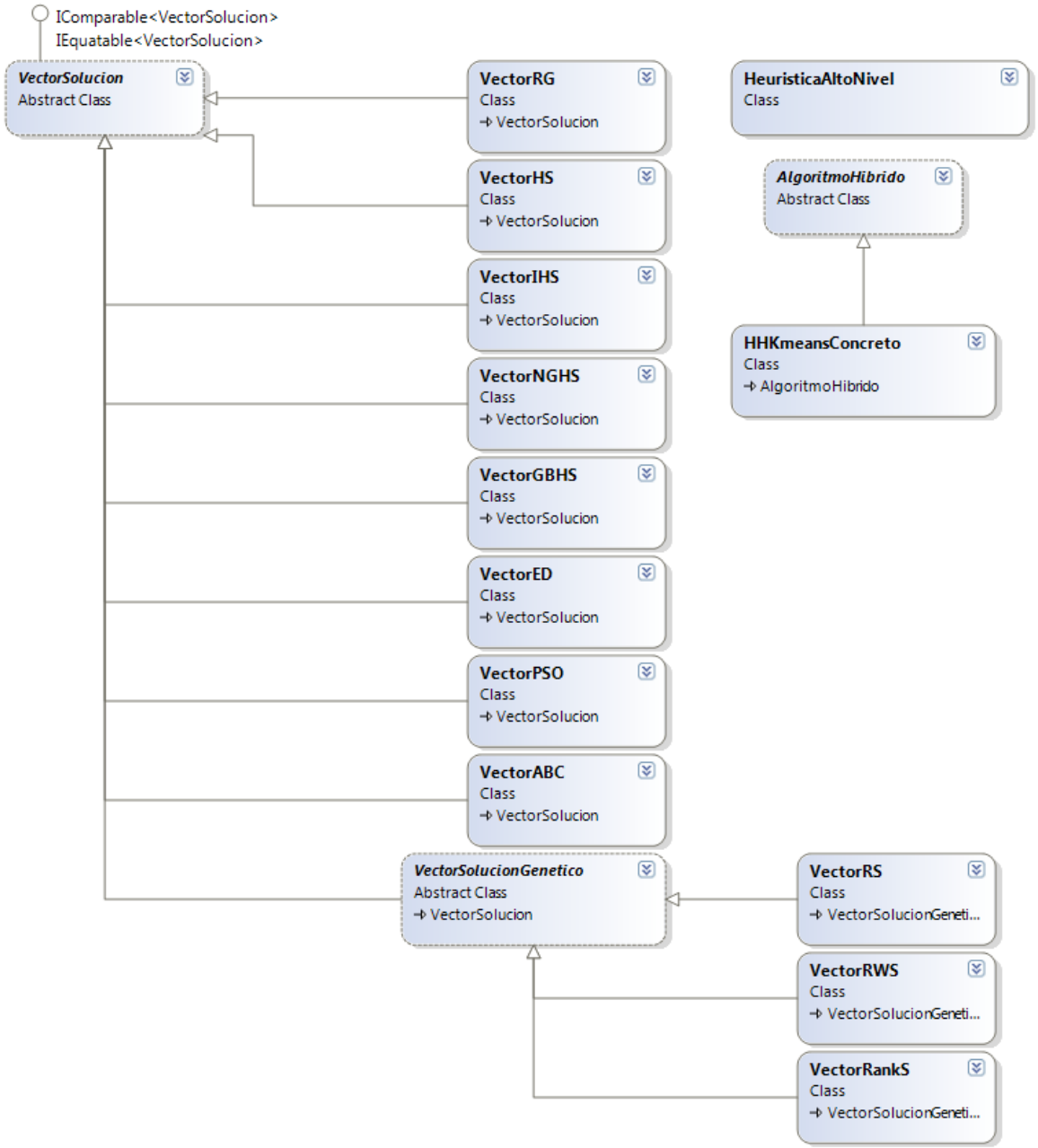
The following is an overview of the most important classes of the business logic layer of the framework, in relation to hyper-heuristics:

- **HHKmeansConcreto.cs:** Responsible for initializing the algorithm parameters, such as: number of iterations, population size. Generates random population, makes the call to the HeuristicaAltoNivel.cs class. If the solution generated is better than other solution previously selected from memory (population) it enters to replace it as long as an identical solution is not found in the vector (to ensure diversity).
- **VectorSolucion.cs:** Vector that stores the centroids and the fitness associated with a solution.
- **VectorRG:** Generates a new random solution vector (random creation of centroids).
- **HeuristicaAltoNivel.cs:** Applies high level heuristics to run: Rank, Random, Tabu or RWS (Roulette wheel), which decides which low-level heuristic to implement.
- **ControllerReplace.cs:** Selects one of the four replacement strategies to run for the new solution: rank replacement, replace the worst, stochastic replacement, or restricted competition replacement. When the low-level heuristic that is running is ABC, depending on each case, replace the worst or restricted competition replacement is performed.
- **Controller.cs:** The logic of each of the low level heuristics are in the classes: VectorABC.cs, VectorED.cs, VectorGBHS.cs, VectorHS.cs, VectorIHS.cs, VectorNGHS.cs, VectorPSO.cs.
- For the low-level heuristics based on genetic algorithms the classes used are: VectorRankS.cs, VectorRS.cs or VectorRWS.cs to select the parents of new solution

vector; for crossing: CruceMultipunto.cs, CruceUniforme.cs or CruceUnPunto.cs; and for the mutation process: MutacionMultibit.cs or MutacionOneBit.cs.

Figure 4-14 shows all the classes inherited from VectorSolucion, where VectorSolución represents the main class.

Figure 4-14: Some classes of the business logic layer

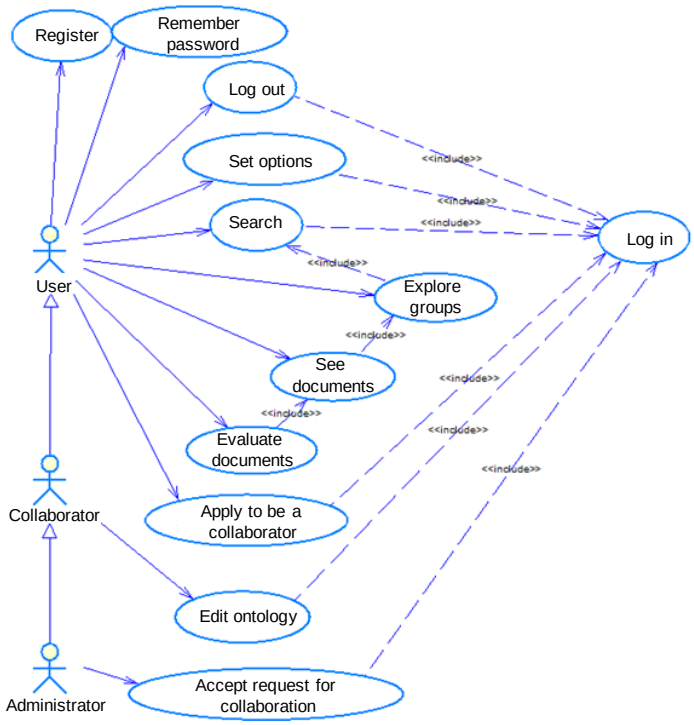


4.2.2 Minerva: The web application

Minerva considers three types of users (roles): Users, Collaborators and Administrators. The use cases for the system for a regular user are (see **Figure 4-15**): register, remember lost password, log in (precondition to use the following use cases), log out, modify options, search, explore groups (requires to search first), see documents (requires to first explore the groups), review the documents (requires to see the documents first).

A regular user can also request to become a collaborator and support the collaborative editing of one or more specific domain ontologies. Once in the role as Collaborator, a user can also collaboratively edit ontologies in the system. Finally, on top of what a collaborator can do, the Administrator is responsible for granting Collaborator privileges to users who so request, amongst other functions (e.g. when a collaborator is doing a bad job, revoke privileges).

Figure 4-15: Minerva use case diagram



Minerva features a web interface centered on the final user, looking to incorporate the main attributes that make it Usable; objective attributes, such as learning ease, memorization ease, efficacy, efficiency or time employed to complete a task, operability

and ease of understanding; and subjective attributes aimed at user satisfaction [61, 71, 130], such as accessibility, functionality, utility, aesthetic and credibility.

Minerva has a simple and usable web interface. This interface consists of a text box for capturing the key words that make up the query, with a built-in capacity to auto-complete and a button that triggers the query process (see **Figure 4-16**).

Figure 4-17 shows the way in which results are displayed once the meta-search engine has completed the recovery, processing, clustering, labeling and overlapping of the documents. In the upper part, from left to right are: the user's nickname, a hyperlink to access the options page, a link to the options page and finally a link to exit the system (log out). Then there is the text box to enter the queries and the button to begin the search. On the left hand side of the page, the clusters appear with their labels and the amount of documents in each group, while the currently selected group appears highlighted. The right side holds the documents for the currently selected group and for each one the title is highlighted with a hyperlink to open the document (in the same tab or window as the browser). Then there are three icons: the first (a magnifying glass over a document) to open the document in a new window; the second (check sign) to mark a document as relevant; and the third (an X) to mark the document as irrelevant. Then the document's snippet appears (as reported by the traditional search engine), the document's URL and the name of the traditional search engines that reported this document (in brackets).

Figure 4-16: Minerva auto-complete option

Figure 4-17: Results display in Minerva

In the options form (see **Figure** 4-18), the user can decide which traditional search engines to use, define whether he is performing the query in English or Spanish, or both, change the target function for the clustering algorithms, select the document representation model, select the clustering algorithm and establish the parameters for the selected algorithm.

Figure 4-18: Minerva configuration options page

5 Experimental Results

This chapter describes the final experiments for the main components of the proposed model - what they comprised, how they developed, and the results they produced. First, the evaluation of the query expansion process in closed test collections is shown. Next, the evaluation of web document clustering algorithms using terms by documents matrices in closed test collections is shown, and the chapter concludes showing an overall evaluation of the model with users.

5.1 Proposed query expansion process

With the objective of evaluating the IDF function proposed in closed test collections two simplified versions of the proposed query expansion model were developed, namely CE-IDF (query expansion model based on keywords) and VP-IDF (query expansion model based on weighted vectors). CE-IDF receives as input the user query and delivers as the result an expanded query with related terms in the user's profile, while VP-IDF in addition returns the weightings of the terms of this expanded query (see [36] for more details).

5.1.1 Data sets for assessment

The data set used for the first experiment was the **CACM IR test collection** available free of charge in http://ir.dcs.gla.ac.uk/resources/test_collections (Test Collections of the R&D Group in Information Retrieval at the University of Glasgow in Scotland, United Kingdom). This data set is a collection of titles and abstracts of articles published in the journal "Communications of the ACM". The collection includes 3,204 documents and 64 queries. For each query, human assessors read all documents and assessed which of them are relevant. In the present investigation, the 52 queries in the collection for which the relevance judgments were complete were taken (see **Table 5-1**). A second experiment was conducted using a different collection of texts called Library & Information Science Abstracts (**LISA IR test collection**), also available at no cost at http://ir.dcs.gla.ac.uk/resources/test_collections. The collection features 6,004 documents

and 35 queries. In this test collection all queries were taken and evaluated (see **Table 5-1**).

Table 5-1: Summary of IR test collections for query expansion process assessment

Dataset	Documents	Queries	Complete Queries	Size (MB)	# of original terms	# of final terms
CACM	3,204	64	52	2.2	49,357	45,414
LISA	5,872	35	35	3.4	20,044	13,924

5.1.2 Metrics for assessment

Given that the basic measures of precision and recall do not take into account the order of relevance of the results, in this evaluation the precision-recall curve is used. The curve represents the precision value at different recall levels [9, 50, 119]. This paper mainly shows tables of precision-recall curve data and some figures that summarize these curves.

5.1.3 Compared systems

In order to verify the performance of the algorithms proposed in this research, the results were compared against the basic ranking (baseline) measure used by Lucene (based on cosine similarity) and the user relevance feedback algorithm proposed by Rocchio [9, 50, 119]. For the latter, the following values are taken for the parameters of this algorithm: α = 50%, β = 50% and γ = 0%. These values reported the best results in four of the five experiments.

5.1.4 Scenarios

Three different scenarios were evaluated, namely 1) with no session memory, 2) with session memory, and 3) with long-term memory.

In the first scenario (**with no session memory**), the execution of each query was simulated five times: the first, called "Basic" or “Baseline”, which uses Lucene similarity (a variant of cosine similarity); the second, a query expansion based on documents relevant or not that showed up in the basic query, called “expansion 1”; then "expansion 2" is performed with the relevance judgments from expansion 1; and in the same way expansions 3 and 4 follow. This has the aim of simulating the process of refining searches that a user carries out when searching repeatedly on a specific topic. It is worth noting

that the memory of the user profile in this case only lasts from one query request to the next (hereafter referred to as no user profile memory).

The second scenario (**with session memory**) follows the same steps as for the previous scenario, but in this case the user profile holds the memory over the five runs of the same query. This process simulates the saving of a user profile during a topic query session.

The third scenario (**with long-term memory**) follows the same steps as for the previous scenarios, but in this case the user profile was maintained throughout all the queries. This process simulates the saving of the user profile throughout its life in the system. This experiment is considered the most important since information retrieval systems or web search generally require keeping a user profile for the entire time that the user is using the system and that this profile adapts itself to the changing search requirements of the users.

5.1.5 Results and discussion

5.1.5.1 CACM IR test collection - with no session memory

Table 5-2 shows the result of the basic query using Lucene (baseline), which starts with a precision value of 56% for a recall level of 10%, and decreases to 7% when the recall level is 100%. It then shows the result of expansion 1, showing a significant improvement in the three algorithms, reaching an average precision of 94% at the first recall level and falling to an average of 16% at the final recall level. This first expansion process shows a precision-recall curve that is much higher at all levels of recall than in the basic query. Furthermore it shows how the three algorithms continue to improve, little by little, in expansions 2, 3 and 4. It can also be seen that VT-IDF achieves from expansion 1 a precision of 94% at 10% recall and in expansion 4 reaches 96%, while Rocchio achieves 93% in the first expansion and a maximum of 98% in expansion 4. Finally, it shows that while CE-IDF achieves an initial and final value of just 94% across the 4 expansions, it performs better than the other algorithms at 20%, 30% and 40% recall.

In this first experiment, it can be seen how using a query expansion based on the relevance of the results previously presented to the user can significantly improve the system's performance. It also shows that for the data collection selected, the Rocchio algorithm achieves better results at the first recall levels, but that VT-IDF and CE-IDF obtain very similar results.

Table 5-2: Precision-recall values for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with no user profile memory (best results are in bold)

	Recall	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Baseline	Lucene	0.56	0.44	0.35	0.26	0.18	0.15	0.12	0.10	0.07	0.07
Expansion 1	CE-IDF	**0.94**	**0.87**	**0.81**	**0.65**	0.48	**0.34**	**0.24**	**0.21**	**0.17**	**0.17**
	VT-IDF	**0.94**	**0.87**	0.78	0.63	**0.50**	**0.34**	0.23	0.18	0.15	0.15
	Rocchio	0.93	0.82	0.72	0.56	0.42	0.32	0.23	0.18	0.16	0.16
Expansion 2	CE-IDF	0.94	**0.93**	0.87	**0.74**	0.56	0.43	0.27	**0.23**	**0.17**	**0.17**
	VT-IDF	0.94	0.92	**0.88**	0.69	**0.58**	**0.44**	**0.29**	0.19	0.15	0.15
	Rocchio	**0.98**	0.91	0.80	0.65	0.48	0.37	0.22	0.17	0.15	0.15
Expansion 3	CE-IDF	0.94	**0.94**	0.89	**0.78**	0.59	0.48	0.31	**0.24**	**0.17**	**0.17**
	VT-IDF	0.94	0.93	**0.90**	0.72	**0.62**	**0.49**	**0.32**	0.19	0.16	0.16
	Rocchio	**0.98**	0.91	0.82	0.67	0.51	0.37	0.22	0.17	0.15	0.15
Expansion 4	CE-IDF	0.94	**0.94**	**0.92**	**0.81**	0.64	0.50	**0.35**	**0.25**	**0.17**	**0.17**
	VT-IDF	0.96	**0.94**	0.90	0.75	**0.66**	**0.54**	0.33	0.20	0.16	0.16
	Rocchio	**0.98**	0.91	0.83	0.67	0.51	0.39	0.22	0.17	0.15	0.15

5.1.5.2 CACM IR test collection - with session memory

Table 5-3 shows a significant improvement by the three algorithms on the basic query (baseline), reaching an average precision of 94% at the first recall level and falling to an average of 16% at the final recall level. This first expansion process shows values in the precision-recall curve that are obviously much higher at all levels of recall than for the basic query. Furthermore it can be seen how Rocchio, VT-IDF and CE-IDF make use of the additional profile information to improve result precision, expansion after expansion, at the different recall levels. It also shows how VT-IDF achieves a precision of 94% at 10% recall from expansion 1 and how in expansion 4 it reaches 96%. Rocchio, meanwhile, reached 93% in the first expansion and a maximum of 98% in expansion 4. Finally, it shows that CE-IDF achieved 94% at the first recall level in all the expansions. Just as in the previous experiment, this algorithm obtains consistently higher precision levels than the others at 20%, 30% and 40% recall. In general, these results are not dissimilar to those obtained in the previous experiment.

5.1.5.3 CACM IR test collection - with long-term memory

Table 5-4 shows that the three algorithms obtained a precision lower than those achieved with basic expansion (baseline). This is due to the weight of the user profile (history of past queries) on the query being performed. But in this case CE-IDF obtains the highest precision values, showing that this algorithm is less sensitive to the history of the user, or

put another way that CE-IDF adapts more quickly to changes in the requirements of the user's queries. In expansion 2, it can be seen how the three algorithms improve precision, but only CE-IDF enhances the basic query (baseline). For expansions 3 and 4 all algorithms progressively improve their results, but only CE-IDF and VT-IDF outperform the basic query, achieving a difference of up to 20% at the first recall level. In all cases, CE-IDF obtains the best results, reaffirming the idea that it is the method that adapts more quickly to new requirements.

Table 5-3: Precision-recall values for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with session memory (best results are in bold)

	Recall	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Baseline	Lucene	0.56	0.44	0.35	0.26	0.18	0.15	0.12	0.10	0.07	0.07
Expansion 1	CE-IDF	**0.94**	**0.87**	**0.81**	**0.65**	0.48	**0.34**	**0.24**	**0.21**	**0.17**	**0.17**
	VT-IDF	**0.94**	**0.87**	0.78	0.63	**0.50**	**0.34**	0.23	0.18	0.15	0.15
	Rocchio	0.93	0.82	0.72	0.56	0.42	0.32	0.23	0.18	0.16	0.16
Expansion 2	CE-IDF	0.94	**0.92**	**0.91**	**0.75**	**0.58**	0.43	0.29	0.23	**0.17**	**0.17**
	VT-IDF	0.94	**0.92**	0.89	0.70	**0.58**	**0.50**	**0.31**	**0.24**	0.16	0.16
	Rocchio	**0.98**	0.91	0.80	0.65	0.48	0.37	0.22	0.17	0.15	0.15
Expansion 3	CE-IDF	0.94	**0.92**	**0.92**	**0.82**	**0.62**	0.47	0.31	0.24	0.17	0.17
	VT-IDF	0.95	**0.92**	0.90	0.76	0.61	**0.52**	**0.36**	**0.26**	**0.18**	**0.18**
	Rocchio	**0.98**	0.91	0.82	0.68	0.51	0.37	0.22	0.17	0.15	0.15
Expansion 4	CE-IDF	0.94	**0.92**	**0.92**	**0.83**	**0.67**	0.49	0.35	0.26	0.17	0.17
	VT-IDF	0.96	**0.92**	0.90	0.81	0.63	**0.52**	**0.37**	**0.28**	**0.20**	**0.20**
	Rocchio	**0.98**	0.91	0.83	0.67	0.51	0.37	0.22	0.17	0.15	0.15

In the values reported for Rocchio, it can be seen that the enhancement process is slower than that obtained with the other two algorithms. Additional evaluations showed that Rocchio can obtain better precision results in this third experiment when $\alpha = 90\%$, $\beta = 10\%$ and $\gamma = 0\%$. In this case, precision ranges between 55% and 62% at the first recall level memory for the four expansions. Unfortunately, using these parameters, precision values for the first two experiments drop to 91% and 94% at the first recall level in the four expansions. With these new values for the parameters it was possible to reduce the weight of history on the user's initial query in the Rocchio algorithm. In addition, it all shows the difficulty that can arise regarding the appropriate tuning of these values in this algorithm.

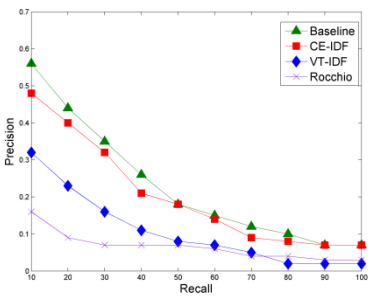
Table 5-4 further shows how CE-IDF achieves from expansion 1 a precision of 48% at 10% recall and how in expansion 4 it reaches 75%, while Rocchio achieves only 16% in the first expansion and 27% in expansion 4. Finally, it shows that VT-IDF despite starting with 32% in the first expansion, manages to equal CE-IDF in expansion 4 with a precision of 75%.

Table 5-4: Precision-recall values for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with long-term memory (best results are in bold)

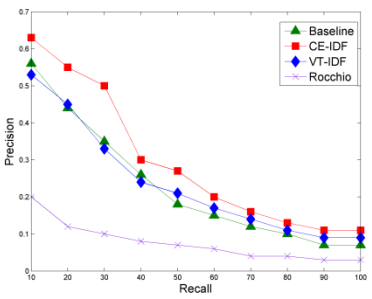
	Recall	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Baseline	Lucene	0.56	0.44	0.35	0.26	0.18	0.15	0.12	0.10	0.07	0.07
Expansion 1	CE-IDF	**0.48**	**0.40**	**0.32**	**0.21**	**0.18**	**0.14**	**0.09**	**0.08**	**0.07**	**0.07**
	VT-IDF	0.32	0.23	0.16	0.11	0.08	0.07	0.05	0.02	0.02	0.02
	Rocchio	0.16	0.09	0.07	0.07	0.07	0.06	0.04	0.04	0.03	0.03
Expansion 2	CE-IDF	**0.63**	**0.55**	**0.50**	**0.30**	**0.27**	**0.20**	**0.16**	**0.13**	**0.11**	**0.11**
	VT-IDF	0.53	0.45	0.33	0.24	0.21	0.17	0.14	0.11	0.09	0.09
	Rocchio	0.20	0.12	0.10	0.08	0.07	0.06	0.04	0.04	0.03	0.03
Expansion 3	CE-IDF	**0.68**	**0.61**	**0.55**	**0.40**	**0.33**	**0.25**	0.18	0.14	**0.11**	**0.11**
	VT-IDF	0.65	0.56	0.46	0.32	0.27	0.22	**0.20**	**0.15**	0.10	0.10
	Rocchio	0.24	0.14	0.13	0.08	0.08	0.07	0.04	0.04	0.04	0.04
Expansion 4	CE-IDF	**0.75**	**0.69**	**0.63**	**0.46**	**0.37**	**0.27**	0.18	0.15	0.11	**0.11**
	VT-IDF	**0.75**	0.67	0.56	0.38	0.33	0.26	**0.22**	**0.17**	**0.12**	**0.11**
	Rocchio	0.27	0.16	0.13	0.10	0.09	0.08	0.05	0.05	0.04	0.04

Finally, **Figure 5-1** shows the precision-recall curve of the four expansions in the last experiment and allows a visual comparison of the results obtained with the three algorithms. In general, the results show that CE-IDF is a better algorithm when taking into account a long-term profile, followed by VT-IDF and lastly Rocchio.

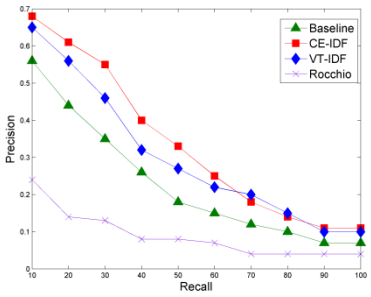
Figure 5-1: Precision-recall curves for Rocchio, VT-IDF and CE-IDF on CACM IR test collection with long-term memory in four expansions



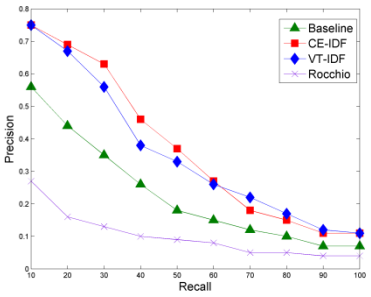
(a) Expansion 1



(b) Expansion 2



(c) Expansion 3



(d) Expansion 4

5.1.5.4 LISA IR test collection - with no session memory

Table 5-5 shows the result of the basic query using Lucene (baseline), which starts at a 55% precision for a recall level of 10%, and decreases to 9% when the recall level is 100%. It also shows the result of expansion 1, demonstrating a significant improvement in the three algorithms, reaching an average of 91.3% precision at the first recall level and falling to an average of 20.3% at the last recall level. This first expansion process displays values for the precision-recall curve that are much higher at all recall levels than the basic query (baseline). Furthermore it shows how the three algorithms slowly improve in expansions 2, 3 and 4. **Table 5-5** also shows how VT-IDF from expansion 1 achieves a precision of 92% at 10% recall and in expansion 4 reaches 94%. Meanwhile, Rocchio achieved 88% in the first expansion and a maximum of 89% in expansion 4. Finally, it shows that CE-IDF achieves an initial and final value of 91% in the 4 expansions, but achieves the best results at recall levels from 20% to 100% from the first expansion. This experiment shows that for the data collection selected, the VT-IDF algorithm performs better at all recall levels for expansions 2, 3 and 4, followed by CE-IDF, but CE-IDF generally shows the best results for expansion 1, followed by VT-IDF, leaving Rocchio last.

5.1.5.5 LISA IR test collection - with session memory

Table 5-6 shows a significant improvement of the three algorithms compared to the basic query (baseline), achieving an average precision of 90% at the first recall level and falling to an average of 16% at the last recall level. This first expansion process displays values for the precision-recall curve that are clearly much higher at all levels of recall than for the basic query (baseline). It also shows how Rocchio, VT-IDF and CE-IDF make use of the additional profile information to improve the accuracy of results, expansion after expansion, at the different recall levels.

Table 5-5: Precision-recall values for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with no session memory (best results are in bold)

	Recall	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Baseline	Lucene	0.55	0.45	0.36	0.31	0.28	0.19	0.10	0.10	0.09	0.09
	CE-IDF	0.91	**0.82**	**0.63**	**0.53**	**0.49**	**0.31**	**0.22**	**0.18**	**0.18**	**0.18**
Expansion 1	VT-IDF	**0.92**	0.78	0.60	0.49	0.46	0.30	0.18	**0.18**	0.15	0.15
	Rocchio	0.88	0.68	0.50	0.41	0.38	0.24	0.19	0.17	0.15	0.15
	CE-IDF	0.91	0.83	0.70	0.59	0.53	0.33	0.27	0.21	0.20	0.20
Expansion 2	VT-IDF	**0.94**	**0.84**	**0.71**	**0.60**	**0.54**	**0.40**	**0.29**	**0.22**	**0.22**	**0.22**
	Rocchio	0.89	0.72	0.54	0.47	0.43	0.31	0.23	0.21	0.21	0.21
	CE-IDF	0.91	0.83	0.71	0.60	0.53	0.34	0.28	0.23	0.20	0.20
Expansion 3	VT-IDF	**0.94**	**0.85**	**0.74**	**0.61**	**0.54**	**0.38**	**0.31**	**0.24**	**0.23**	**0.23**
	Rocchio	0.89	0.72	0.54	0.47	0.44	0.30	0.22	0.21	0.20	0.20
	CE-IDF	0.91	0.83	0.74	0.62	0.52	0.38	0.29	0.22	0.19	0.19
Expansion 4	VT-IDF	**0.94**	**0.86**	**0.75**	**0.63**	**0.54**	**0.41**	**0.35**	**0.24**	**0.22**	**0.22**
	Rocchio	0.89	0.72	0.54	0.47	0.44	0.30	0.22	0.21	0.20	0.20

Table 5-6 also shows how VT-IDF from expansion 1 achieves an accuracy of 92% at 10% recall and in expansion 4 reaches 94%. Rocchio meanwhile achieved 88% in the first expansion and a maximum of 89% in expansion 4. Finally, it shows that CE-IDF achieved 91% at the first recall level in all expansions. Just as in the previous experiment, in general it secures the best results in expansion 1, but the behavior of the three algorithms is maintained, leaving Rocchio in last place once more.

5.1.5.6 LISA IR test collection - with long-term memory

Table 5-7 shows how VT-IDF and Rocchio obtained lower precision values than those achieved with the basic expansion (baseline), particularly in the case of Rocchio, whose value was really low even in relation to VT-IDF, due to the weight of user profile (the history of past queries) on the query being performed. But in this case the CE-IDF algorithm yields a higher precision value, showing that this algorithm is less sensitive to the history of the user, i.e. CE-IDF adapts quickly to the changes in the requirements of the user queries, similar to the results obtained with CACM IR test collection.

In expansion 1, it can be seen how CE-IDF improves on the basic query (baseline). For expansion 3, VT-IDF reaches a value higher than that of the basic expansion and in expansion 4 Rocchio continues to fall below that of the basic (baseline), since its improvement with respect to the previous expansions is very small, generally obtaining

very low values, failing to climb above 3% precision at the 10% recall level. In all cases, CE-IDF obtains the best results, reaffirming the idea that this is a method that quickly adapts to the new user requirements.

Table 5-6: Precision-recall values for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with session memory (best results are in bold)

	Recall	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Baseline	Lucene	0.55	0.45	0.36	0.31	0.28	0.19	0.10	0.10	0.09	0.09
Expansion 1	CE-IDF	0.91	**0.82**	**0.63**	**0.53**	**0.49**	**0.31**	**0.22**	**0.18**	**0.18**	**0.18**
	VT-IDF	**0.92**	0.78	0.60	0.49	0.46	0.30	0.18	**0.18**	0.15	0.15
	Rocchio	0.88	0.68	0.50	0.41	0.38	0.24	0.19	0.17	0.15	0.15
Expansion 2	CE-IDF	0.91	0.83	**0.71**	0.59	0.54	0.35	0.27	0.20	0.20	0.20
	VT-IDF	**0.94**	**0.87**	0.70	**0.62**	**0.56**	**0.42**	**0.31**	**0.23**	0.20	0.20
	Rocchio	0.89	0.71	0.53	0.47	0.43	0.31	0.23	0.21	**0.21**	**0.21**
Expansion 3	CE-IDF	0.91	0.83	0.71	0.59	**0.56**	0.38	0.29	0.20	0.20	0.20
	VT-IDF	**0.94**	**0.89**	**0.74**	**0.62**	**0.56**	**0.45**	**0.36**	**0.25**	**0.21**	**0.21**
	Rocchio	0.89	0.71	0.54	0.47	0.44	0.30	0.22	0.21	0.20	0.20
Expansion 4	CE-IDF	0.91	0.83	**0.71**	0.59	0.56	0.39	0.29	0.20	0.20	0.20
	VT-IDF	**0.94**	**0.89**	**0.71**	**0.62**	**0.57**	**0.48**	**0.37**	**0.26**	**0.23**	**0.23**
	Rocchio	0.89	0.71	0.54	0.47	0.44	0.30	0.22	0.21	0.20	0.20

It is further noted that the process of improvement in Rocchio is much slower than that obtained with the other two algorithms. Additional tests showed that Rocchio can achieve better precision in this third experiment when $\alpha = 90\%$, $\beta = 10\%$ and $\gamma = 0\%$. In this case the precision ranges between 4.3% and 5.7% at the first recall level during the four expansions. Unfortunately, with these parameters the precision values for the first two experiments decreased to 71% at the first recall level in the four expansions. With these new values for the parameters is possible to reduce the weight of history on the user's initial query in the Rocchio algorithm. The above also confirms the obvious difficulty that the proper definition of these values can present in this algorithm.

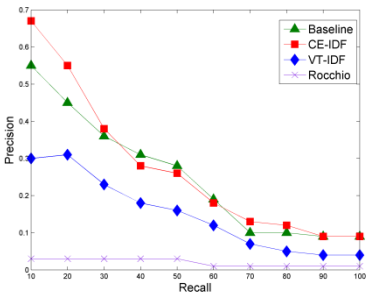
Table 5-7 also shows how CE-IDF achieves a precision of 67% at 10% recall from expansion 1 and in expansion 4 reaches 83%. Rocchio meanwhile achieves only 3% in all expansions. Finally, it shows that VT-IDF despite starting with 30% in the first expansion reaches 69% precision in expansion 4.

Table 5-7: Precision-recall values for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with long-term memory (best results are in bold)

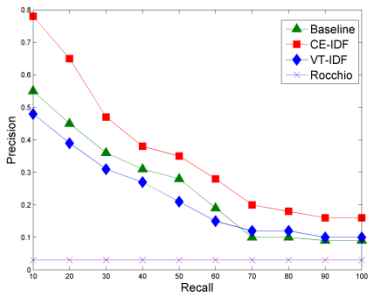
	Recall	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Baseline	Lucene	0.55	0.45	0.36	0.31	0.28	0.19	0.10	0.10	0.09	0.09
Expansion 1	CE-IDF	**0.67**	**0.55**	**0.38**	**0.28**	**0.26**	**0.18**	**0.13**	**0.12**	**0.09**	**0.09**
	VT-IDF	0.30	0.31	0.23	0.18	0.16	0.12	0.07	0.05	0.04	0.04
	Rocchio	0.03	0.03	0.03	0.03	0.03	0.01	0.01	0.01	0.01	0.01
Expansion 2	CE-IDF	**0.78**	**0.65**	**0.47**	**0.38**	**0.35**	**0.28**	**0.20**	**0.18**	**0.16**	**0.16**
	VT-IDF	0.48	0.39	0.31	0.27	0.21	0.15	0.12	0.12	0.10	0.10
	Rocchio	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Expansion 3	CE-IDF	**0.83**	**0.67**	**0.50**	**0.41**	**0.37**	**0.31**	**0.21**	**0.18**	**0.17**	**0.17**
	VT-IDF	0.59	0.53	0.41	0.34	0.28	0.20	0.16	0.14	0.12	0.12
	Rocchio	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Expansion 4	CE-IDF	**0.83**	**0.70**	**0.52**	**0.44**	**0.41**	**0.33**	**0.23**	**0.21**	**0.20**	**0.20**
	VT-IDF	0.69	0.61	0.47	0.42	0.36	0.26	0.18	0.16	0.14	0.14
	Rocchio	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03

Figure 5-2 shows the precision-recall curve of the four expansions in the last experiment and allows a visual comparison of the results obtained with the three algorithms. As in CACM IR test collection, CE-IDF shows its superiority in LISA IR test collection when taking the long-term profile into account, followed by VT-IDF and lastly Rocchio with values still well below the basic query (baseline) in all expansions.

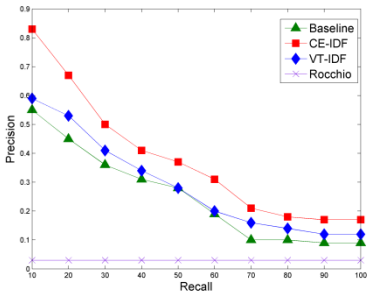
Figure 5-2: Precision-recall curves for Rocchio, VT-IDF and CE-IDF on LISA IR test collection with long-term memory in four expansions



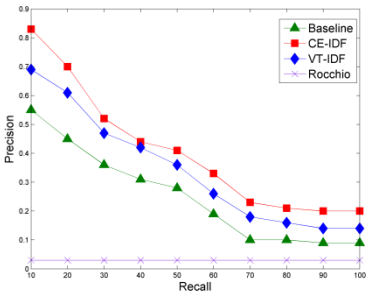
(a) Expansion 1



(b) Expansion 2



(c) Expansion 3



(d) Expansion 4

5.2 Proposed web document clustering algorithms

5.2.1 Data sets for assessment

The proposed algorithms and hyper-heuristic framework were used for clustering of web results on four traditional benchmarking data sets, namely: DMOZ-50, AMBIENT, MORESQUE and ODP-239. These data sets correspond to a total of 447 queries with their ideal solutions (see a summary of data sets in **Table 5-8**).

DMOZ-50 data set consists of 50 queries derived from the Open Directory Project (acronym for Mozilla's Directory). Each query has on average 129.14 documents, 6.02 subtopics (meanings from very different subjects), and 22.62 relevant results per retrieved subtopic. Each query is a large collection of documents, each with a comparatively small set of classes and large number of documents per class. In this data set, query keywords are not available. The collection is available for download at http://artemisa.unicauca.edu.co/~ccobos/wdc/wdc.htm.

AMBIENT (AMBIguous ENTries) data set consists of 44 queries extracted from *ambiguous* Wikipedia entries. Each query has on average 50.55 ranked search results collected from Yahoo! (manually annotated with document-level relevance judgments per subtopic), 7.91 subtopics, and 7.72 relevant results per retrieved subtopic. Most of the queries in AMBIENT data set are of single word (1 keyword) and they are all available. AMBIENT data set measures the ability to retrieve subtopics contained in the search results (documents retrieved by Yahoo!), not all possible subtopics of a query. This data set can be downloaded at http://credo.fub.it/ambient.

MORESQUE (MORE Sense-tagged QUEry results) data set consists of 114 ambiguous queries which were conducted as a complement to AMBIENT data set. This data set tests the behavior of web search algorithms on queries of different lengths, ranging from 1 to 4 words. MORESQUE data set provides 114 queries of length 2, 3 and 4 (all of them are available), together with an average of 53.54 top results (documents) from Yahoo!, 3.82 subtopics, and 19.43 relevant results per retrieved subtopic. This data set can be downloaded at http://lcl.uniroma1.it/moresque.

ODP-239 data set consists of 239 queries derived from Open Directory Project (http://www.dmoz.org). Each query has on average 106.95 documents (each document consists of a URL, title and a very short description), 9.56 subtopics, and 11.38 relevant results per retrieved subtopic. ODP-239 consists of many small collections, each with a comparatively large set of classes, as opposed to having one large collection of documents with a small number of classes. The topics, subtopics, and their associated documents were selected in such a way that the distribution of documents across subtopics reflects the relative importance of subtopics. The collection is available for download at http://credo.fub.it/ODP-239.

Table 5-8: Summary of data sets for query web document clustering assessment

Dataset	Documents	Queries	Documents by query	Subtopics	Relevant results per subtopic	Average number of processed terms by query
DMOZ-50	6,457	50	129.14	6.02	22.62	643.92
AMBIENT	2,224	44	50.55	7.91	7.72	381,8
MORESQUE	6,104	114	53.54	3.82	19.43	342,2
ODP-239	25,561	239	106.95	9.56	11.38	188,6

5.2.2 Metrics for assessment

The assessment included two aspects: Ground-truth validation and Assessment of user behavior. Ground-truth validation is aimed at assessing how good a clustering method is at recovering known clusters (referred to as classes) from a gold standard partition. Several evaluation measures are available for this task, including precision, recall, F-measure, Fall-out, and Accuracy (Rand index) [116]. In this research, the weighted Precision, weighted Recall, weighted F-measure (the harmonic means of precision and

recall), weighted Fall-out and weighted Accuracy measures are used to evaluate the quality of solution.

Given a collection of clusters, $\{C_1, C_2, \dots C_k\}$, to evaluate its weighted Precision, weighted Recall and weighted F-measure with respect to a collection of ideal clusters $\{C_1^i, C_2^i, \dots C_h^i\}$, these steps are followed: (a) find for each ideal cluster C_n^i a distinct cluster C_m that best approximates it in the collection being evaluated, and evaluate $P(C, C^i)$, $R(C, C^i)$, and $F(C, C^i)$ as defined by (5.1) and (5.2). (b) Calculate the weighted Precision (P), weighted Recall (R) and weighted F-measure (F) based on (5.3).

$$P(C, C^i) = \frac{|C \cap C^i|}{|C|} \; and \; R(C, C^i) = \frac{|C \cap C^i|}{|C^i|} \tag{5.1}$$

Where C is a cluster of documents and cluster C^i is an ideal cluster of documents

$$F(C, C^i) = \frac{2 * P(C, C^i) * R(C, C^i)}{P(C, C^i) + R(C, C^i)} \tag{5.2}$$

$$P = \frac{1}{T}\sum_{j=1}^{h}|C_j^i| * P(C_m, C_j^i), \; R = \frac{1}{T}\sum_{j=1}^{h}|C_j^i| * R(C_m, C_j^i), \text{ and } \; F = \frac{2*P*R}{P+R} \quad \text{where} \; T = \sum_{j=1}^{h}|C_j^i| \tag{5.3}$$

In relation of the Assessment of user behavior, the Subtopic Search Length under k document sufficiency (SSL_k) metric was used for assessing the ease in which users can use clustering results, in summary, assessment of user behavior [16, 26, 167]. This measure is defined as the average number of items (cluster labels or search results) that must be examined before finding a sufficient number (k) of documents relevant to any of the query subtopics, assuming that both cluster labels and search results are read sequentially from top to bottom, and that only cluster with labels relevant to the subtopic at hand are opened. SSL_k allows an evaluation of full-subtopic retrieval (i.e., retrieval of multiple documents relevant to any subtopic) rather than focusing on subtopic coverage (i.e., retrieving at least one relevant document for some subtopics). SSL_k also allows a realistic modelization of the user search behavior because the role played by cluster labels is taken into account.

5.2.3 Compared systems

All algorithms and results of the hyper-heuristic framework were compared with Bisecting K-means, STC and Lingo from two perspectives, the quality of the clustering results and the ease with which users can use clustering results. **Suffix Tree Clustering** (STC) [144] is the original web search clustering approach based on suffix trees and frequent phrases,

while **Lingo** [150] is a well-known successor of STC. In this web clustering algorithm (implemented in the Carrot^2 open source framework) frequent phrases of documents are extracted first using suffix arrays, then the best frequent phrases are selected using Singular Value Decomposition (SVD), and finally documents are allocated to such frequent phrases.

All algorithms and the best result of the hyper-heuristic framework was also compared to Lingo3G, KeySRC, OPTIMSRC, and Yahoo! results using previous reported results. **Lingo3G** is a commercial web clustering algorithm also available on Carrot^2. This algorithm is very different to Lingo, it uses a custom-built meta-heuristic to select well-defined and diverse cluster labels. **KeySRC** [16] is a web clustering engine built on top of STC with part-of-speech pruning and dynamic selection of the cut-off level of the clustering dendrogram. **OPTIMSRC** [26] is a web document clustering algorithm based on generation of the meta partition with stochastic hill climbing followed by meta labeling based on Lingo, STC, KeySRC labels. And **Yahoo!** results which are the original search results returned by Yahoo! search engine. In reference [26], SSL_k results for Yahoo! on AMBIENT data set are presented.

5.2.4 Results and discussion

In order to select the best web document clustering heuristic several tests were executed using the WDC-HH framework. All individual heuristics and several combinations (pairs, thirds, quartets, quintets, groups of ten, and groups of fifteen) of best heuristics were evaluated using all data sets during one (1) second of execution time. For all tests, the assessment metrics were calculated and best results were summarized in **Table** 5-9.

The best heuristic obtained in the hyper-heuristic framework was WDC-HH-BHRK (Global-best Harmony Search as low-level heuristic and Rank replacement heuristic). This combination corresponds to the IGBHSK algorithm presented in section 3.4.1. Results of IGBHSK were also compared with results reported by other state of the art algorithms based on F-measure and SSL_k (see **Table 5-11** and **Table 5-12**).

Table 5-10 shows that on the DMOZ-50 data set, WDC-CSK outperforms other algorithms in all evaluation measures, except Lingo in precision. On the AMBIENT data set, IGBHSK outperforms other algorithms in recall, F-measure, and accuracy. Fall-out is

also competitive for IGBHSK on this data set, but precision favors Lingo. Results on MORESQUE data set are favorable for STC but precision favors Lingo and fall-out WDC-MA. On the ODP-239 data set, IGBHSK outperforms all other algorithms in recall and precision, WDC-CSK outperforms the others in F-measure and fall-out, and Lingo outperforms all others in precision. In general, Bisecting k-means obtains the worst results on all data sets.

IGBHSK, WDC-MA, and WDC-CSK obtain a better number of clusters on all data sets, and the difference is highly significant in comparison with Lingo and STC algorithms. On average, IGBHSK differs from the ideal number of clusters (7.54) by around 0.17 clusters; WDC-MA by 0.89; WDC-CSK by 1.57; Bisecting K-means by 3.73; Lingo by 20.28; and STC by 6.73. Also IGBHSK outperforms Lingo and STC in recall, F-measure, and fall-out to, and its accuracy is very similar to STC. Finally WDC-CSK is very competitive with IGBHSK results but Lingo is best in terms of precision.

Average rankings of precision using the Friedman test show that Lingo is better than other algorithms, with a Friedman statistic (distributed according to chi-square with 5 degrees of freedom) equal to 575.971556 and p-value equal to 2.1964130514362523E-10 (see **Table 5-11**). Additionally, Lingo is an improvement on all other algorithms; WDC-CSK is an improvement on other algorithms except Lingo; WDC-MA is an improvement on IGBHSK, Bisecting K-means; and STC is an improvement on Bisecting K-means with a level of significance equal to 0.95 in the Wilcoxon test.

It is important to highlight that precision in Lingo is biased, because the number of clusters is excessively high and the value of precision only shows that Lingo is able to allocate small number of documents relating to the same topic in small number of generated clusters.

Average rankings of recall using the Friedman test shows that IGBHSK is better than all other algorithms, with a Friedman statistic equal to 1071.990093 and p-value equal to 0.0 (see **Table 5-11**). Additionally, IGBHSK is an improvement on all other algorithms, while WDC-CSK is an improvement on the rest of the algorithms; WDC-MA is an improvement on STC, Lingo and Bisecting K-means; STC is an improvement on Lingo and Bisecting K-

means; and Lingo is an improvement on Bisecting K-means with a level of significance equal to 0.95 in the Wilcoxon test.

Table 5-9: Consolidated results of best heuristics (best results are in bold)

Heuristic	Estimated K	Precision	Re-call	F-measure	Accuracy	Fall-Out	SSL_1	SSL_2	SSL_3	SSL_4	Sum of SSL
BHRK o **IGBHK algorithm**	7.37	69.27	**49.18**	**52.40**	**0.78**	0.05	**16.55**	**25.41**	**32.92**	**40.89**	**115.8**
Tabu: BHRK-BHWR	**7.39**	69.28	**49.16**	**52.38**	**0.78**	0.05	16.71	25.57	33.11	41.02	124.3
BHWR	7.36	69.26	**49.16**	**52.37**	**0.78**	0.05	16.71	25.64	33.09	41	**116.4**
Rank: BHRK BHWR	**7.39**	69.29	**49.10**	**52.33**	**0.78**	0.05	16.58	25.49	33.05	41.06	124.3
Tabu: BHRK BHWR HSRK	7.69	70.09	48.66	52.27	**0.78**	0.05	16.67	25.61	33.13	41.1	124.3
Rank: BHRK BHWR HSRK	7.71	70.15	48.66	52.24	**0.78**	0.05	16.71	25.56	33.06	40.99	124.3
Rank: BHRK BHWR HSWR	7.74	70.26	48.59	52.23	**0.78**	0.05	16.8	25.64	33.08	40.98	124.3
Tabu: BHRK-BHWR-HSWR	7.70	70.12	48.63	52.22	**0.78**	0.05	16.61	25.48	32.99	40.94	124.3
BHSR	**7.50**	69.55	48.81	52.19	**0.78**	0.05	16.63	25.58	33.13	41.11	**116.4**
Tabu: BHWR-HSRK	7.83	70.46	48.44	52.19	0.77	0.05	16.79	25.65	33.13	40.99	124.3
Tabu: BHRK-HSRK	7.83	70.48	48.42	52.17	0.77	0.05	16.79	25.73	33.19	41.08	124.3
Tabu: BHWR-HSWR	7.86	70.50	48.40	52.16	0.77	0.05	16.78	25.64	33.15	41.11	124.3
Tabu: BHRK-HSWR	7.85	70.48	48.36	52.15	0.77	0.05	16.65	25.52	33.13	41.09	124.3
Tabu: BHRK BHWR HSWR HSRK	7.85	70.43	48.39	52.13	0.77	0.05	16.74	25.67	33.14	41.09	124.3
BHRC	**7.39**	69.07	48.89	52.13	**0.78**	0.05	**16.55**	25.49	33	40.97	**116**
Rank: BHRK BHWR HSWR HSRK	7.87	70.52	48.32	52.10	0.77	0.05	16.74	25.69	33.2	41.18	124.3
Tabu: BHWR HSWR HSRK	8.02	70.95	48.15	52.09	0.77	0.05	16.83	25.75	33.26	41.1	124.3
Tabu: BHRK BHWR RWCUMMRK	7.84	70.65	48.32	52.08	0.77	0.05	16.77	25.65	33.11	41.02	124.3
Tabu: BHRK BHWR HSWR RWCUMMRK	7.94	70.89	48.20	52.08	0.77	0.05	16.76	25.63	33.11	40.99	124.3
Rank: BHRK BHWR HSRK RWCUMMRK	8.00	71.11	48.07	52.05	0.77	0.05	16.85	25.75	33.27	41.22	124.3
Tabu: BHRK BHWR HSRK RWCUMMRK	7.93	70.86	48.16	52.04	0.77	0.05	16.74	25.59	33.08	41.03	124.3
Rank: BHWR HSWR	8.11	71.08	48.01	52.04	0.77	0.05	16.83	25.64	33.15	41.11	124.3
Rank: BHWR HSWR HSRK	8.07	71.00	48.07	52.03	0.77	0.05	16.8	25.64	33.11	41.14	124.3
Rank: BHRK BHWR HSWR RWCUMMRK	8.00	71.07	48.04	52.01	0.77	0.05	16.79	25.6	33.1	41.07	124.3
…											
RWCUMMRK o **WDC-MA algorithm**	8.43	**71.69**	46.74	51.18	0.77	0.05	16,88	25,81	33,27	41,23	**117,2**

Table 5-10: Ground-Truth Validation Results (best results are in bold)

Data set	Algorithm	Estimated k	Difference to ideal k	Precision	Recall	F-measure	Accuracy	Fall-out
DMOZ-50 Ideal K 6,02	IGBHSK*	**8.19**	**2.17**	84.03	**70.00**	74.25	91.65	0.03
	WDC-CSK	9.22	3.20	**90.34**	**70.26**	**76.77**	**92.13**	**0.01**
	WDC-MA	9.05	3.03	86.38	67.43	73.48	**91.19**	**0.02**
	Bisecting K-means	10.98	4.96	70.94	43.37	50.32	84.42	0.05
	Lingo	34.29	28.27	83.85	37.88	48.23	83.41	0.05
	STC	16.00	9.98	84.82	57.85	65.12	88.81	0.03
AMBIENT Ideal K 7,91	IGBHSK	5.82	2.09	74.11	**62.36**	**63.21**	**84.30**	0.04
	WDC-CSK	**7.39**	**0.52**	78.13	58.77	61.79	82.78	0.04
	WDC-MA	6.75	1.16	75.68	58.35	60.77	82.75	0.04
	Bisecting K-means	11.39	3.48	76.46	40.65	45.97	77.12	0.04
	Lingo	20.86	12.95	**86.75**	50.21	58.68	80.43	**0.03**
	STC	11.00	3.09	72.40	53.14	55.38	81.89	0.06
MORESQUE Ideal K 3,81	IGBHSK	**6.09**	**2.27**	86.81	43.30	52.43	60.22	0.05
	WDC-CSK	7.92	4.10	88.33	39.58	49.30	58.48	0.06
	WDC-MA	6.94	3.13	87.54	40.11	49.59	58.50	**0.04**
	Bisecting K-means	10.36	6.55	87.36	30.05	38.69	53.47	0.04
	Lingo	20.16	16.34	**90.50**	39.35	50.55	59.18	0.06
	STC	11.17	7.35	82.83	**49.96**	**57.18**	**65.45**	0.13
ODP-239 Ideal K 9,56	IGBHSK	8.09	1.47	56.93	**45.21**	45.83	**81.92**	0.06
	WDC-CSK	**9.78**	**0.22**	60.90	43.93	**46.18**	81.68	**0.05**
	WDC-MA	9.32	0.24	60.33	43.43	45.51	81.50	0.05
	Bisecting K-means	11.75	2.19	55.60	32.12	34.92	78.16	0.06
	Lingo	31.39	21.83	**71.56**	32.93	41.01	79.15	0.07
	STC	15.98	6.42	57.33	39.74	41.80	80.65	0.10
Average Ideal K 7,54	IGBHSK	**7.37**	**0.17**	69.27	**49.18**	**52.40**	**77.71**	**0.05**
	WDC-CSK	9.01	1.57	72.88	47.22	**51.93**	**77.04**	**0.05**
	WDC-MA	8.43	0.89	71.69	46.74	**51.18**	76.83	**0.05**
	Bisecting K-means	11.27	3.73	67.47	33.69	38.69	72.46	0.05
	Lingo	27.81	20.28	**79.26**	36.82	45.99	74.66	0.06
	STC	14.27	6.73	68.39	45.69	49.67	**77.81**	0.09

* IGBHSK correspond to WDC-HH-BHRK

Table 5-11: Ground-Truth Friedman Test Rankings for all algorithms (best results are in bold)

	Precision		Recall		F-measure		Accuracy		Fall-out	
Algorithm	Ranking	Position	Ranking	Position	Ranking	Position	Ranking	Position	Ranking	Position
IGBHSK	4.5257	6	**1.9172**	**1**	**2.5145**	**1**	**2.472**	**1**	4.2584	6
WDC-CSK	2.7595	2	2.7036	2	2.6275	2	2.7148	2	**2.6745**	**1**
WDC-MA	3.3535	3	2.9362	3	2.9821	3	2.8904	3	2.8512	2
Bisecting K-means	4.4575	5	5.3468	6	5.5168	6	5.2528	6	3.7237	4
Lingo	**2.1242**	**1**	4.7282	5	3.8758	5	4.1309	5	3.274	3
STC	3.7796	4	3.368	4	3.4832	4	3.5391	4	4.2181	5

Average rankings of F-measure using the Friedman test show that IGBHSK is better than all other algorithms, with a Friedman statistic equal to 793.042506 and p-value equal to 2.8720026357120787E-10 (see **Table 5-11**). Additionally, IGBHSK is an improvement on all other algorithms; WDC-CSK is an improvement on the rest of the algorithms; WDC-MA is an improvement on STC, Lingo and Bisecting K-means; STC is an improvement on Lingo and Bisecting K-means; and Lingo is an improvement on Bisecting K-means with a level of significance equal to 0.95 in the Wilcoxon test

Average rankings of accuracy using the Friedman test show that IGBHSK is better than all other algorithms with a Friedman statistic equal to 704.57015 and p-value equal to 2.593960601871004E-10 (see **Table 5-11**). Additionally, IGBHSK improves upon all other algorithms; WDC-CSK improves upon WDC-MA, Lingo, and Bisecting K-means; WDC-MA improves upon Lingo and Bisecting K-means; STC improves upon Lingo and Bisecting K-means; and Lingo improves upon Bisecting K-means with a level of significance equal to 0.95 in the Wilcoxon test. Also WDC-CSK and WDC-MA improve upon STC with a level of significance equal to 0.90 in the same test.

Average rankings of fall-out using the Friedman test show that WDC-CSK is better than other algorithms with a Friedman statistic equal to 293.016299 and p-value equal to 1.4519774271803954E-10 (see **Table 5-11**). Additionally, WDC-CSK improves upon all other algorithms; WDC-MA improves upon the remaining algorithms; Bisecting K-means improves upon IGBHSK and STC; and IGBHSK and Lingo improve upon STC with a level of significance equal to 0.95 in the Wilcoxon test.

In **Figure 5-3** the curves for precision, recall and F-measure through different numbers of iterations are shown. All values increase with the number of iterations. Therefore, when users can wait longer for results, IGBHSK organizes clusters of documents better and proved the best option. BBIC with cosine similarity is a good option for clustering of web results because precision, recall, and F-measure all increase when IGBHSK evolves ($\mathrm{Fmeasure} = 2.0306 * \ln(\mathrm{iteration}) + 53.654$ with $R^2 = 0.9739$), but in some iterations (e.g. 50 to 55 iterations) this positive relation fails. The research group thus plans to define a better fitness function for evolutionary algorithms in the clustering of web results based on multiobjective genetic programming. Further analysis showed that in general IGBHSK increases cluster quality (based on precision, recall, and F-measure) when it uses more iterations regardless of the number of documents, number of topics, or number of attributes in the data set. Only MORESQUE data set does not comply with this rule.

New solution vectors generated using the IGBHSK algorithm (BH-RK combined heuristic) increase its effectiveness over iterations. **Figure 5-4** shows 64% of effectiveness of the new solution in the first five iterations, i.e. the new solution is better than other solutions in the population 64% of the time. The effectiveness then increases to 81% in five more iterations, to 89% in iteration 15, and finally to around 98% in the sixtieth iteration (the generated solution vector is almost always better than the other solution vector selected from the population). The behavior in **Figure 5-4** is for the AMBIENT data set, but it is similar for other data sets.

In **Figure 5-5**, the curves of precision, recall and F-measure through different numbers of iterations are shown. All values increase with the number of iterations. Therefore, when users also can wait longer for results, WDC-CSK organized clusters of documents better and proved the best option. BBIC with cosine similarity is a good option for clustering of web results because precision and recall both increase when WDC-CSK optimizes BBIC ($\mathrm{Fmeasure} = 1.0995 * \ln(\mathrm{iteration}) + 57.016$ with $R^2 = 0.89$), but in some iterations (e.g. 40 to 50 iterations) this positive relation fails (similar to IGBHSK behavior). The figure also shows the F-measure when WDC-CSK works together with BIC. Results with BBIC are better than with BIC in all iterations.

New nests generated using the abandon, split and merge methods from WDC-CSK increase its effectiveness over iterations. **Figure** 5-6 shows 32% effectiveness of the new

solution in the first five iterations, i.e. the new solution is better than other solutions in nest population. Effectiveness then increases to 54% in five more iterations, to 67% in iteration 15, and finally reaches around 90% in the eightieth iteration. The behavior in **Figure** 5-6 is for the AMBIENT data set, but is similar for other data sets. The behavior of WDC-CSK using BIC as a fitness function is similar but on average is 2.7% less effective. IGBHSK reports better effectiveness than WDC-CSK, so it can report better results in early stages of the evolution process.

Figure 5-3: Precision, Recall and F-Measure for WDC-HH-BHRK (IGBHSK) through the various iterations on the AMBIENT data set

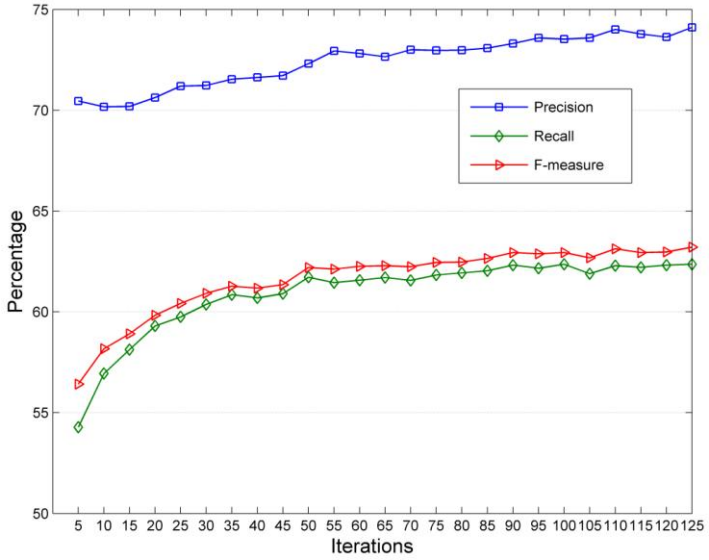


Figure 5-4: Effectiveness of new solution vectors generated at different number of iterations on AMBIENT data set for WDC-HH-BHRK (IGBHSK) algorithm

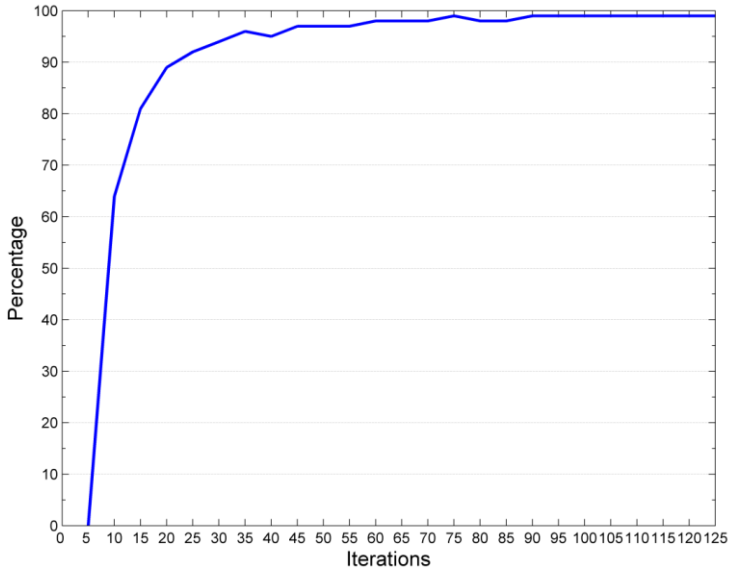


Figure 5-5: Precision, Recall and F-Measure for WDC-CSK through different iterations on AMBIENT data set

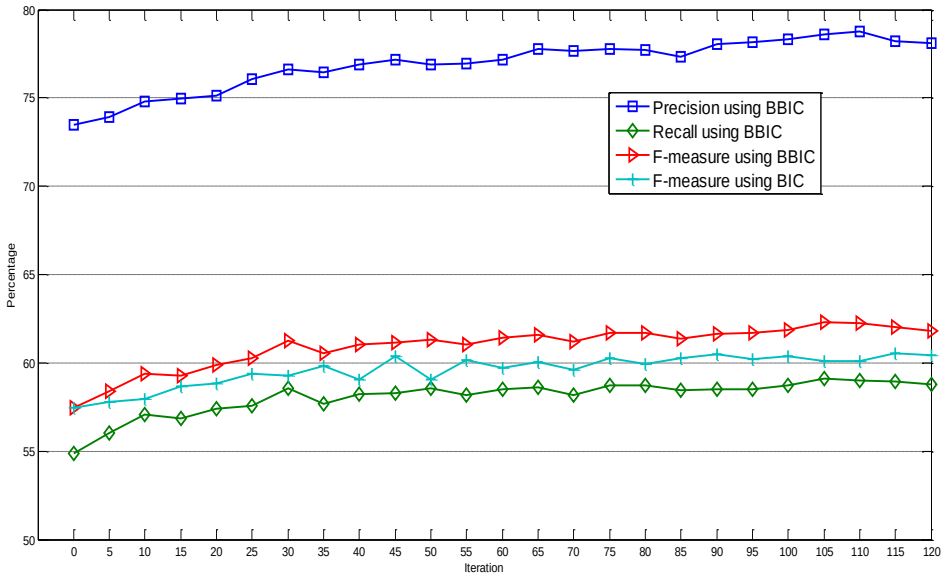


Figure 5-6: Effectiveness of new nest generated at different number of iterations on AMBIENT data set for WDC-CSK algorithm

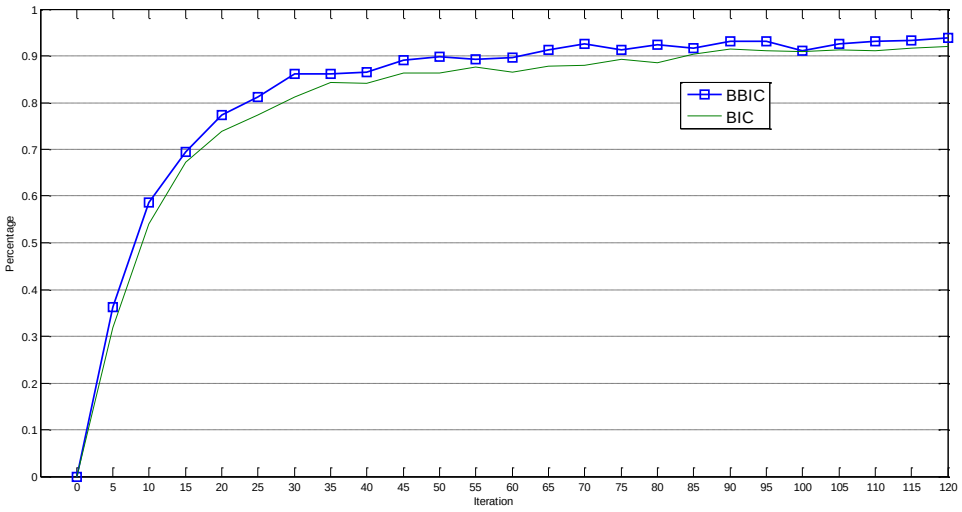


Table 5-12 shows results on SSL_k (with k=1, 2, 3, 4) measure for all data sets. On AMBIENT, MORESQUE and ODP-239, IGBHSK outperforms all other algorithms in SSL_1, SSL_2, SSL_3, SSL_4 and Sum of SSL_k, but WDC-CSK and WDC-MA are very competitive. DMOZ data set gives poor results for IGBHSK, WDC-CSL, and WDC-MA because this

data set has no queries, while MORESQUE has 2, 3 or 4 keywords to describe the query and therefore IGBHSK, WDC-CSL, and WDC-MA improves results of SSL_k by a greater amount. Keywords in queries are very important for the labeling step in the IGBHSK, WDC-CSK, and WDC-MA algorithms (traditional scenario of clustering of web results).

Table 5-12: User Behavior Evaluation (best results are in bold)

Data set	Algorithm	SSL_1	SSL_2	SLL_3	SSL_4	Sum of SSL_k
DMOZ-50	WDC-HH-BHRK	15.1	19.1	22.1	24.6	80.9
	WDC-CSK	17.0	20.6	23.2	25.4	86.1
	WDC-MA	15.8	19.9	22.8	25.1	83.7
	Lingo	14.2	16.6	**18.5**	21.9	71.2
	STC	**12.1**	**16.4**	18.6	**21.3**	**68.4**
AMBIENT	WDC-HH-BHRK	**14.6**	**26.0**	**32.5**	**37.1**	**110.2**
	WDC-CSK	15.5	26.8	33.4	37.9	113.6
	WDC-MA	14.9	26.4	32.9	37.6	112.0
	Lingo	22.4	36.5	47.2	54.3	160.4
	STC	27.2	44.9	54.8	60.4	187.3
	Best combination*	21.7	29.3	33.2	37.3	121.5
	OPTIMSRC*	20.6	28.9	34.1	38.9	122.5
	Lingo*	24.4	30.6	36.6	40.7	132.3
	KeySRC*	24.1	32.4	38.2	42.1	136.8
	Lingo3G*	24.0	32.4	39.6	43.0	139.0
	Yahoo!*	21.6	35.5	42.0	47.6	146.7
MORESQUE	WDC-HH-BHRK	**11.1**	**18.6**	**24.1**	**27.8**	**81.6**
	WDC-CSK	11.8	19.4	24.9	28.6	84.7
	WDC-MA	11.2	19.0	24.5	28.2	83.1
	Lingo	16.5	26.4	33.9	39.2	116.0
	STC	19.6	32.3	40.2	45.2	137.3
ODP-239	WDC-HH-BHRK	**19.8**	**29.9**	**39.5**	**51.2**	**140.4**
	WDC-CSK	20.2	30.2	39.7	51.6	141.7
	WDC-MA	20.1	30.1	39.6	51.4	141.3
	Lingo	25.6	38.1	51.4	66.4	181.5
	STC	26.3	43.1	60.7	78.4	208.5
	Lingo**	22.0	35.0	48.3	63.8	169.1
	Lingo3G**	21.5	34.4	48.2	63.3	167.4
	KeySRC**	22.8	40.1	57.3	75	195.2

* Taken from [26]

** Taken from [24]

5.3 Experiments with Users

On completion of the process of defining, creating and evaluating in the laboratory the best algorithm obtained integrated in the finished model, three blind experiments were performed with 90 users from the final semesters (VIII, IX and X) of the systems engineering program at the University of Cauca. Tests were conducted on the Lingo algorithm and Minerva (using the IGBHSK algorithm, the BBIC function and the CDM matrix) in two parallel groups of 15 students each. In both experiments, the results of a query were evaluated, to measure the quality of the results in terms of clarity and usefulness of the labels, the relevance of the documents to the groups and the order of these within the groups, using a survey (**Table 5-13**) with three sections: 1) specific questions applied to the first ten groups generated by each algorithm, 2) general questions, and 3) observations where the students can give suggestions and insights about the behavior of the algorithm evaluated.

In the survey the possible answers are: SA = Strongly Agree, A = Agree, PA = Partially agree, PD = Partially disagree and SD = Strongly Disagree. Each answer has a weight of 1-5 with 1 being the lowest score and 5 the highest.

Based on the results of the algorithms evaluated, the average of each of the specific questions for each group evaluated was calculated. Next, the general average for each question was calculated (the same process, this time applied to the results for the general questions). The results obtained were those presented in **Table 5-14**.

Figure 5-7 shows the results obtained for the questions put for each of the algorithms. On all questions, the completed model (Minerva) performs better, but in the general questions (5 and 6) the difference between Lingo and Minerva is more marked, indicating that the users considered the number of groups and label quality of Minerva to be more appropriate than those of Lingo.

In **Figure 5-8** and **Figure 5-9** below an overview of the survey results can be found, where Lingo has a higher percentage of negative findings (strongly and partially disagree), while Minerva has a higher percentage of positive findings (agree and strongly

agree). The intermediate findings (somewhat agree) are very similar between the two systems, the difference being just 0.7%.

Table 5-13: Survey form for testing with users

Specific questions

Group #

Is the group label representative of the documents in the group?

☐ SA A ☐ PA ☐ PD ☐ SD ☐

Is the label useful for choosing the specific sub-topic of the query?

☐ SA A ☐ PA ☐ PD ☐ SD ☐

Are the group documents related to the label of the group to which they belong?

☐ SA A ☐ PA ☐ PD ☐ SD ☐

Is the relevance of the documents (position or order) in the group as it should be?

☐ SA A ☐ PA ☐ PD ☐ SD ☐

General questions

1. Is the number of groups appropriate?

☐ SA A ☐ PA ☐ PD ☐ SD ☐

2. Is the quality of the labeling generally high?

☐ SA A ☐ PA ☐ PD ☐ SD ☐

Table 5-14: Average results of the survey (best results are in bold)

	Specific questions				**General questions**	
	Question 1	**Question 2**	**Question 3**	**Question 4**	**Question 1**	**Question 2**
IGBHSK	**3.92**	**3.73**	**3.79**	**3.65**	**3.89**	**3.71**
Lingo	3.35	3.29	3.54	3.17	2.64	2.87

Based on the results of the algorithms evaluated, the Fleiss Kappa test was carried out to measure the degree of agreement between the responses of different students, producing a result of only very slight overall agreement regarding evaluation of the systems, meaning that the test results are not conclusive and that the students (judges) generally express different opinions.

Figure 5-7: Comparative chart of the results for each question (char of values in Table 5-14)

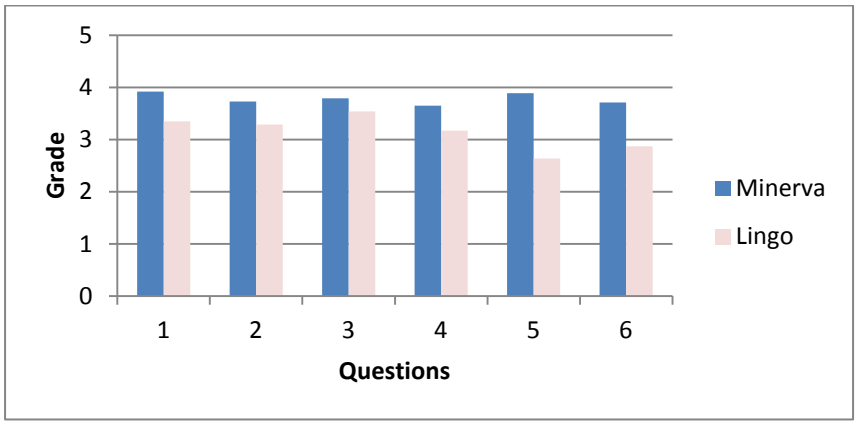


Figure 5-8: Overall comparison of the survey results by responses to each algorithm

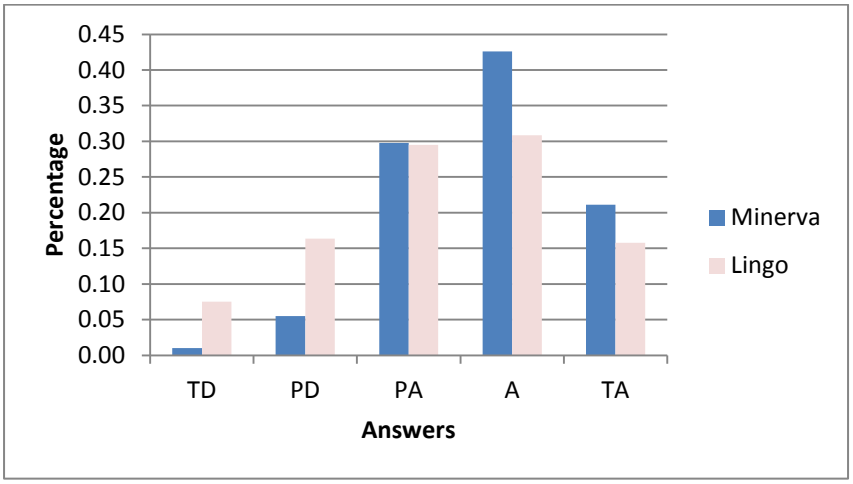
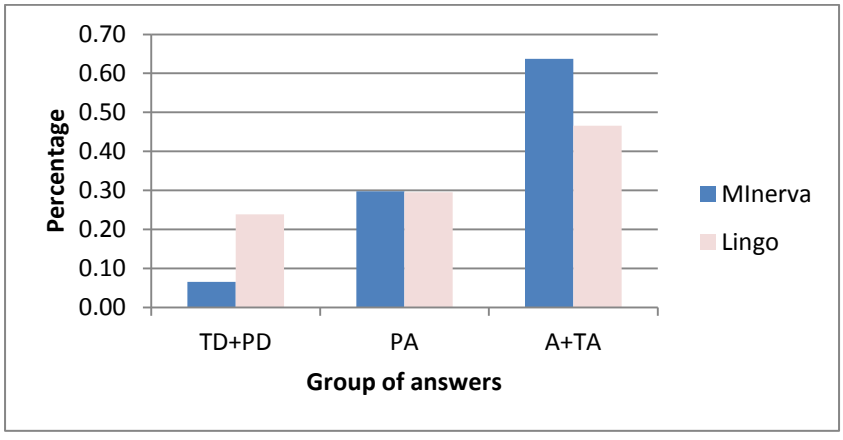


Figure 5-9: Comparative grouping of survey results by response for each algorithm



6 Conclusions, Recommendations and Future Work

6.1 Conclusions

A web document clustering meta search model was successfully proposed, implemented and evaluated. It includes five main components. The first component is responsible for supporting the query expansion of the user based on the semantic relationship (extracted from ontologies that are organized in a taxonomic hierarchy) of the terms that each user has stored in their profile. The second component is responsible for search result acquisition from traditional web search engines (Google, Yahoo! and Bing). The third component is responsible for pre-processing documents and generating two representations of them, one based on vector space model and another based on frequent phrases. The fourth component is responsible for cluster construction and labeling, for which there are three heuristic algorithms that perform clustering based on vector space representation of the results, and labeling based on frequent phrases representation. The fifth component is responsible for visualization of the resulting clusters, which involves the presentation of search results organized into thematic groups (folders) and updating of the user profile based on the feedback registered (relevant or not relevant).

The query expansion process proposed achieves better results than Rocchio with short-term and long-term memories on the data sets used for the test, namely CACM IR test collection and LISA IR test collection. One of the major strengths of the proposed model is its ability to adapt to the changes in the user profile (local needs). Another strength is its simplicity, which facilitated its inclusion in the meta search model proposed.

The proposed web document clustering meta search model uses two document representation models: the vector space model, used in the preprocessing and clustering

components, and the frequent phrases model, used in the labeling component. In this model, the general taxonomy of knowledge is used to hierarchically organize a set of domain-specific ontologies. The domain-specific ontologies are modeled as a related set of terms in multiple languages (English and Spanish), which are used to expand the queries according to specific user requirements. To organize the ontologies, an structure called Inverted Concepts Index is used, which facilitates access to the terms in the ontologies (avoiding direct handling of the OWL content) and the relation of the ontologies to the profile of each user. In the user profile statistics are recorded regarding the terms and their concepts present in the documents that the user evaluates as relevant or not relevant. The profile is modeled in such a way to allow its efficient updating because, as reported in previous studies, it is one of the most complex structures to handle.

Using genetic programming, a function was proposed whose objective is to guide the optimization process of web document clustering heuristic algorithms using K-means as local improvement strategy. This function was called Balanced Bayesian Information Criterion (BBIC) and its formulation involves maximizing cluster cohesion (internal similarity to each cluster) expressed by minimizing the sum of squared errors of the dissimilarities of the documents with respect to each centroid of the cluster to which it belongs and maximizing the separability (difference) between the clusters, expressed by the maximization of the average distance of the centroids of the different clusters in the solution. This function, BBIC, was used and evaluated on a wide set of algorithms and data sets, and in all of these it achieved better F-measure results in fewer iterations than BIC, this latter the best index reported previously for web document clustering.

Three algorithms for clustering web documents were successfully modeled, implemented and evaluated, namely:

- An algorithm called IGBHSK (Iterative Global-best Harmony Search with the K-means algorithm) using meta heuristic Global-best Harmony Search as a global search strategy, K-means as local optimizer or intensification strategy and a Rank-based replacement mechanism allowing the new individuals created in the evolution process to enter the population without producing too much selective pressure and improving the population in each iteration.
- An algorithm called WDC-MA that carries out selection by roulette in order to choose the parents that perform the reproduction process. This algorithm in each iteration

produces only one new agent, which is the result of a uniform crossover and the mutation of multiple bits of each one of the centroids of which it is composed. Then the offspring goes through an optimization process using K-means and becomes part of the population based on a strategy of replacement by Rank.

- An algorithm called WDC-CSK (Web document clustering based on cuckoo search and the K-means algorithm) that uses meta heuristic of cuckoo search as an effective strategy for exploration of the search space and K-means as an intensification strategy. In cuckoo search, Lévy flights required to be modified by Split and Merge operations on centroids in the nests. This is due to the fact that Lévy flights do not have a metaphor that easily supports their transfer to the clustering problem. The Split and Merge operations allow solutions to be found in the neighborhood of a nest in the population, which is then optimized by the K-means algorithm and enters the nest population if its fitness value is better than that reported for a nest selected at random from the population. This algorithm also incorporated the abandonment of nests as a search strategy around all of the search space (exploration), thereby achieving a balance between exploration and exploitation.

The three algorithms proposed (IGBHSK, WDC-MA and WDC-CSK) share certain characteristics:

- They contemplate an evolutionary process of independent islands in parallel, which allow for the harnessing of existing computational resources in order to find a better solution to the problem of web document clustering in the same execution time as for a single island.
- They use the BBIC function, a function that reports the best results and is therefore recommended, but may be executed with other fitness functions such as BIC, Davies-Bouldin index, as well as others.
- They use a matrix of input data, which can be the term-document matrix (TDM), concept-document matrix (CDM) or frequent concept-document matrix (FCDM).
- They are compact algorithms, i.e. they generate in each iteration only one new solution vector (agent, nest or harmony), helping to improve progressively the population in each iteration and exploiting at each stage the best features of the individuals in the population to generate the new individuals. This achieves a better control of the evolutionary process and its relationship to the short run time that the algorithms have to deliver the results.

The results of the evaluation of the algorithms showed that IGBHSK is the best solution in terms of the reports of recall, F-measure, precision, and the estimated number of clusters, but also showed that WDC-CSK is a very competitive solution. The evaluation process also included evaluation of user behavior through the SSL_k measure, for which these algorithms report far superior results to those of the state of the art.

In the evaluation process, which included evaluation of the quality of the clustering process and user behavior (browsing on the results) the non-parametric statistical tests of Friedman and Wilcoxon were used with the aim of defining the confidence level of the results. The findings of the experiments are supported by a 95% confidence in most cases and a 90% confidence for a few exceptions. Results for IGBHSK and WDC-CSK show an improvement of between 4.5% and 13% on F-measure, between 3.5% and 28% on recall, between 1% and 3% on accuracy, and between 18% and 49% on fall-out. Experiments also show improvements on accumulate SSLk values of between 21% and 31%.

The successful modeling, implementation and evaluation of a hyper heuristic framework specifically for the problems related to web document clustering were achieved. The HH framework can be run directly for web document clustering and it works in the same way as description-centered algorithms. It uses four high-level selection strategies: random selection, tabu selection, rank selection and roulette wheel selection based on the performance of low-level heuristics. It also employs a wide set of low-level heuristics: harmony search, improved harmony search, new global harmony search, global-best harmony search, particle swarm optimization, artificial bee colony, differential evolution, and a further eighteen heuristics based on genetic algorithms, each a product of the combination of micro-heuristics: restricted pairing selection (RM), roulette wheel selection (RW), rank selection, one-point crossover (UP), uniform crossover (CU), multi-point crossover (CM), one-bit uniform mutation (MU) and multi-bit uniform mutation (MM). It also uses the K-means algorithm as a strategy for improving the solution at the local level and, based on the Balanced Bayesian Information Criterion; it is able to automatically define the number of groups. Finally it uses four replacement strategies: replace worst, restricted competition replacement, stochastic replacement and rank replacement.

The heuristics that make up IGBHSK (Global-best harmony search and Rank replacement) and WDC-MA (Performance-based roulette wheel selection, uniform crossover and multi-bit uniform mutation) were included in the hyper-heuristic framework. This allowed previously developed proposals to be compared against a much larger number of heuristics, dynamically constructed within the framework. The results of the evaluation of the framework show that IGBHSK is the best solution found up until now and that heuristics composed of Global-best Harmony Search with other replacement methods are equally competitive. WDC-MA was outperformed by more than 20 heuristics, although the results obtained with this algorithm are better than those reported by other state of the art algorithms.

Although the framework evaluation process was not exhaustive, since this requires a total of 1.58456E+29 ($\sum_{i=1}^{97}(97!/(i! \times (97-i)!))$) evaluations, the evaluations also showed that the combinations of heuristics occupy important places in the ranking of the best results obtained (positions 2, 4, 5, 6, 7 and more). Such combinations include the use, for example, of Global-best Harmony Search along with two replacement strategies simultaneously, e.g. rank replacement and replace worst. Also using Harmony Search with Rank replacement combined with the two mentioned above. Therefore, it is necessary to conduct more systematic evaluations with the framework in order to find better results or re-affirm that that found up until now is the best solution from that universe of heuristics.

The methodology used to carry out the project, Iterative Research Pattern, with the additional phase of documentation and disclosure of results was appropriate for the task. Development of each of the instances and their specific products allowed achieving iteratively the objectives, obtaining early disclosure of the results and receiving feedback from each of the sub products. It also means that each instance can fit the specific requirements of the product to be brought to fruition and permits the project progress to be more easily controlled.

6.2 Recommendations and Future work

Given that experiments with users did not yield conclusive results or that could be generalized, it is planned to conduct the evaluation of the proposed model using an experimental design of multiple time series with multiple post-tests over a long period of

time. This would help to better define the actual behavior of the model in a real use scenario.

Given that carrying out a thorough evaluation of the hyper-heuristic framework involves an unfeasible number of evaluations, the use of covering arrays is recommended in order to significantly reduce the number of assessments and obtain results that have a greater coverage of the total possible evaluations. Another option is to take advantage of the new super-computers being installed in the country - or making use of those already found in other countries - to conduct the greatest possible number of evaluations.

Future work will include proposing an objective function for evolutionary algorithms that perform the clustering of web results from a multiobjective genetic programming approach that can outperform results reported for BBIC. In the present research, a naive approach was taken that enabled optimization of precision and recall using F-measure, but it is necessary to evaluate other objectives, for example SSL_k measure.

There remain other tasks. Among these are: 1) using disambiguation techniques in order to improve the quality of clustering results and the comparison of results with other algorithms, 2) designing other high, low and replacement heuristics in the proposed framework and comparing results with the state of the art in web result clustering, and 3) designing other bio-inspired algorithms for web result clustering and comparing results with IGBHSK.

References

[1] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences, 178 (2008) 37-51.

[2] R.M. Aliguliyev, Clustering of document collection - A weighting approach, Expert Systems with Applications, 36 (2009) 7904-7916.

[3] Amazon, Sitio web de Amazon, in.

[4] N.O. Andrews, E.A. Fox, Recent Developments in Document Clustering, in: Technical Report TR-07-35, Computer Science, Virginia Tech, 2007.

[5] F. Archetti, P. Campanelli, E. Fersini, E. Messina, A Hierarchical Document Clustering Environment Based on the Induced Bisecting k-Means, in: Flexible Query Answering Systems, 2006, pp. 257-269.

[6] L.G. Astaiza A., A practical approach to scheduling examinations., Ing. Investig., 25 (2005) 92-100.

[7] A. Asuncion, D.J. Newman, UCI Machine Learning Repository in, University of California, School of Information and Computer Science, Irvine, CA, 2007.

[8] M.A. Aufaure, R. Soussi, H. Baazaoui, SIRO: On-line semantic information retrieval using ontologies, in: Digital Information Management, 2007. ICDIM '07. 2nd International Conference on, 2007, pp. 321-326.

[9] R. Baeza-Yates, A., B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., 1999.

[10] R. Baeza-Yates, C. Castillo, B. Keith, Web Searching, in: Encyclopedia of Language & Linguistics, Elsevier, Oxford, 2006, pp. 527-538.

[11] C.L. Barry, User-Defined Relevance Criteria: An Exploratory Study, JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE-A, 45 (1994) 149-159.

[12] H.W. Beck, T. Anwar, S.B. Navathe, A conceptual clustering algorithm for database schema design, Knowledge and Data Engineering, IEEE Transactions on, 6 (1994) 396-411.

[13] F. Beil, M. Ester, X. Xu, Frequent term-based text clustering, in: KDD '02: International conference on Knowledge discovery and data mining (ACM SIGKDD), ACM, Edmonton, Alberta, Canada, 2002, pp. 436-442.

[14] P. Berkhin, Survey Of Clustering Data Mining Techniques, in, Accrue Software, Inc., 2002.

[15] P. Berkhin, J. Kogan, C. Nicholas, M. Teboulle, A Survey of Clustering Data Mining Techniques, in: Grouping Multidimensional Data, Springer-Verlag, 2006, pp. 25-71.

[16] A. Bernardini, C. Carpineto, M. D'Amico, Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering, in: WI-IAT '09: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, 2009, pp. 206-213.

[17] S.K. Bhatia, J.S. Deogun, Conceptual clustering in information retrieval, Systems, Man, and Cybernetics, Part B, IEEE Transactions on, 28 (1998) 427-436.

[18] C. Biancalana, A. Micarelli, Social Tagging in Query Expansion: A New Way for Personalized Web Search, in: SocialCom-09 the 2009 IEEE International Conference on Social Computing, Vancouver, Canada, 2009, pp. 1060-1065.

[19] L. Bianchi, M. Dorigo, L.M. Gambardella, W. J. Gutjahr, A survey on metaheuristics for stochastic combinatorial optimization, Natural Computing: an international journal, 8 (2009) 239-287.

[20] G. Bin, L. Tie-Yan, F. Guang, A.T.Q. Tao Qin, A.Q.-S.C. Qian-Sheng Cheng, A.W.-Y.M. Wei-Ying Ma, Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph copartitioning, Knowledge and Data Engineering, IEEE Transactions on, 17 (2005) 1263-1273.
[21] M. Blanco, Estudio de buscadores, in.
[22] BrightPlanet, The Deep Web: Surfacing Hidden Value, (2000).
[23] F. Can, R. Nuray, A.B. Sevdik, Automatic performance evaluation of Web search engines, Information Processing & Management, 40 (2004) 495-514.
[24] C. Carpineto, M. D'Amico, G. Romano, Evaluating subtopic retrieval methods: Clustering versus diversification of search results, Information Processing & Management, 48 (2012) 358-373.
[25] C. Carpineto, S. Osiński, G. Romano, D. Weiss, A survey of Web clustering engines, ACM Comput. Surv., 41 (2009) 1-38.
[26] C. Carpineto, G. Romano, Optimal meta search results clustering, in: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, ACM, Geneva, Switzerland, 2010, pp. 170-177.
[27] M. Carullo, E. Binaghi, I. Gallo, An online document clustering technique for short web contents, Pattern Recognition Letters, 30 (2009) 870-876.
[28] M. Centelles, Taxonomías para la categorización y la organización de la información en sitios, Hipertext.net, 3 (2005).
[29] S. Chakrabarti, Web Search and Information Retrieval, in: Mining the Web, Morgan Kaufmann, San Francisco, 2003, pp. 45-76.
[30] M.H. Chehreghani, H. Abolhassani, M.H. Chehreghani, Density link-based methods for clustering web pages, Decision Support Systems, 47 (2009) 374-382.
[31] L.-C. Chen, C.-J. Luh, C. Jou, Generating page clippings from web search results using a dynamically terminated genetic algorithm, Information Systems, 30 (2005) 299-316.
[32] V. Cherkassky, The Nature Of Statistical Learning Theory~, Neural Networks, IEEE Transactions on, 8 (1997) 1564-1564.
[33] R.H.L. Chiang, C.E.H. Chua, V.C. Storey, A smart web query method for semantic retrieval of web data, Data & Knowledge Engineering, 38 (2001) 63-84.
[34] W. Chih-Hung, Y. Cheng-Jer, L. Shie-Jue, An entropy-based evaluation function for conceptual clustering, in: Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on, 1995, pp. 4307-4312 vol.4305.
[35] C. Cobos, J. Andrade, W. Constain, M. Mendoza, E. León, Web document clustering based on Global-Best Harmony Search, K-means, Frequent Term Sets and Bayesian Information Criterion, in: 2010 IEEE Congress on Evolutionary Computation (CEC), IEEE, Barcelona, Spain, 2010, pp. 4637-4644.
[36] C. Cobos, E. Estevez, M. Mendoza, L. Gomez, E. Leon, Algoritmos de Expansión de Consulta basados en una Nueva Función Discreta de Relevancia, Revista UIS Ingenierías, 10 (2011) 9-22.
[37] C. Cobos, E. León, M. Mendoza, A harmony search algorithm for clustering with feature selection, Rev. Fac. Ing. Univ. Antioquia, 55 (2010) 153-164.
[38] C. Cobos, M. Mendoza, E. Leon, A hyper-heuristic approach to design and tuning heuristic methods for web document clustering, in: 2011 IEEE Congress on Evolutionary Computation (CEC), IEEE, New Orleans, USA., 2011, pp. 1350-1358.
[39] C. Cobos, M. Mendoza, E. León, M. Manic, E. Herrera-Viedma, Clustering of Web Search Results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion, in: 2013 IFSA-NAFIPS Joint Congress, International Fuzzy Systems Association, Edmonton, Canada, 2013, pp. 6.
[40] C. Cobos, C. Montealegre, M. Mejía, M. Mendoza, E. León, Web Document Clustering based on a New Niching Memetic Algorithm, Term-Document Matrix and Bayesian Information Criterion, in: 2010 IEEE Congress on Evolutionary Computation (CEC), IEEE, Barcelona, Spain, 2010, pp. 4629-4636.

[41] C. Cobos, L. Muñoz, M. Mendoza, E. León, E. Herrera-Viedma, Fitness Function Obtained from a Genetic Programming Approach for Web Document Clustering Using Evolutionary Algorithms, in: J. Pavón, N. Duque-Méndez, R. Fuentes-Fernández (Eds.) Advances in Artificial Intelligence – IBERAMIA 2012, Springer Berlin Heidelberg, 2012, pp. 179-188.
[42] C. Cobos, O. Rodriguez, J. Rivera, J. Betancourt, M. Mendoza, E. León, E. Herrera-Viedma, A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes, Information Processing & Management, 49 (2013) 607-625.
[43] C. Cobos, J. Zuñiga, J. Guarin, E. León, M. Mendoza, CMIN – A Case Tool Based on CRISP-DM to Support Data Mining Projects, Revista Ingeniería e Investigación de la Universidad Nacional de Colombia, 30 (2010) 45-56.
[44] M.B. Dale, On the Comparison of Conceptual Clustering and Numerical Taxonomy, Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-7 (1985) 241-244.
[45] F.d.A.T. de Carvalho, Fuzzy c-means clustering methods for symbolic interval data, Pattern Recognition Letters, 28 (2007) 423-437.
[46] F.d.A.T. De Carvalho, Y. Lechevallier, Partitional clustering algorithms for symbolic interval data based on single adaptive distances, Pattern Recognition, In Press, Corrected Proof.
[47] Dmoz, Open Directory Project Web Site (Dmoz), in.
[48] Dogpile.com, Different Engines, Different Results: Web Searchers Not Always Finding What They're Looking for Online, in, 2007.
[49] S. Dominich, PageRank: Quantitative Model of Interaction Information Retrieval, in: 12th International World Wide Web Conference WWW '03 International Workshop on Mobile Web Technologies WF7, World Wide Web Consortium WWW-C, Institute for Electrical and Electronics Engineers IEEE, John von Neumann Computer Society, Budapest, Hungary, 2003, pp. 20-24
[50] S. Dominich, The Modern Algebra of Information Retrieval, Springer-Verlag Berlin Heidelberg, 2008.
[51] Z. Dongsheng, W. Liqing, Study on Key Techniques of Query Expansion Based on Ontology and Its Application, in: Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on, 2009, pp. 1-4.
[52] A. Douzal-Chouakria, A. Diallo, F. Giroud, Adaptive clustering for time series: Application for identifying cell cycle expressed genes, Computational Statistics & Data Analysis, 53 (2009) 1414-1426.
[53] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Second ed., John Wiley & Sons Inc., 2001.
[54] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on, 1995, pp. 39-43.
[55] E.N. Efthimiadis, Query Expansion, in, Information School, University of Washington 1996, pp. in: Williams, Martha E., ed. Annual Review of Information Systems and Technology (ARIST), v31, pp 121-187, 1996.
[56] C. Fellbaum, WordNet: An Electronic Lexical Database, MIT Press, 1998.
[57] P. Ferragina, A. Gullì, The Anatomy of SnakeT: A Hierarchical Clustering Engine for Web-Page Snippets, in: Knowledge Discovery in Databases: PKDD 2004, 2004, pp. 506-508.
[58] Ferretsoft, Webferret Web Site, in, 2008.
[59] E. Fersini, E. Messina, F. Archetti, A probabilistic relational approach for web document clustering, Information Processing & Management, 46 (2010) 117-130.
[60] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognition, 41 (2008) 176-190.
[61] D.H. Fisher, Knowledge acquisition via incremental conceptual clustering, Machine Learning, 2 (1987) 139-172.
[62] E.W. Forgy, Cluster analysis of multivariate data: efficiency versus interpretability of classifications, Biometrics, 21 (1965) 768-769.

[63] R. Forsati, M.R. Meybodi, M. Mahdavi, A.G. Neiat, Hybridization of K-Means and Harmony Search Methods for Web Page Clustering, in: WI-IAT '08: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008, pp. 329-335.
[64] W.B. Frakes, R.A. Baeza-Yates, Information Retrieval Data Structures & Algorithms Prentice-Hall, 1992.
[65] Y. Fu, Z. Li, T.S. Huang, A.K. Katsaggelos, Locally adaptive subspace and similarity metric learning for visual data clustering and retrieval, Computer Vision and Image Understanding, 110 (2008) 390-402.
[66] R. Fuentes, J. Pavón, Agentes para la recuperación de información especializada en Internet, in: II Taller en Desarrollo de Sistemas Multiagente, Granada, Spain, 2005.
[67] B. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, in: Proceedings of the SIAM International Conference on Data Mining, 2003, pp. 59-70.
[68] E. Garcia, RSJ-PM Tutorial: A Tutorial on the Robertson-Sparck Jones Probabilistic Model for Information Retrieval, in, 2009.
[69] Z. Geem, J. Kim, G.V. Loganathan, A New Heuristic Optimization Algorithm: Harmony Search, Simulation, 76 (2001) 60-68.
[70] F. Geraci, M. Pellegrini, M. Maggini, F. Sebastiani, Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution, in: String Processing and Information Retrieval, 2006, pp. 25-36.
[71] M. Giugni O., R. Loaiza B., Metodología para el desarrollo de portales centrada en el usuario: una evaluación empírica, Revista electrónica de estudios telemáticos, 7 (2008) 17.
[72] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., 1989.
[73] E. González, J. Turmo, Non-Parametric Document Clustering by Ensemble Methods, Procesamiento del lenguaje natural, 40 (2008).
[74] Google, Google Directory Web Site, in.
[75] Google, Google Personalized Web Search, in.
[76] J. Grobler, A.P. Engelbrecht, G. Kendall, V.S.S. Yadavalli, Alternative hyper-heuristic strategies for multi-method global optimization, in: 2010 IEEE Congress on Evolutionary Computation (CEC), IEEE, Barcelona, Spain, 2010, pp. 826-833.
[77] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA Data Mining Software: An Update, SIGKDD Explorations, 11 (2009).
[78] K. Hammouda, Web Mining: Clustering Web Documents A Preliminary Review, in, 2001, pp. 1-13.
[79] K.M. Hammouda, M.S. Kamel, Efficient phrase-based document indexing for Web document clustering, Knowledge and Data Engineering, IEEE Transactions on, 16 (2004) 1279-1296.
[80] J. Han, M. Kamber, Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufman Publishers, 2006.
[81] J. Han, M. Kamber, A.K.H. Tung, Spatial Clustering Methods in Data Mining: A Survey, in: Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2001, pp. 1-29.
[82] L. Han, G. Chen, A fuzzy clustering method of construction of ontology-based user profiles, Advances in Engineering Software, 40 (2009) 535-540.
[83] L. Han, G. Chen, HQE: A hybrid method for query expansion, Expert Systems with Applications, 36 (2009) 7985-7991.
[84] X. He, J.-B. Wang, Z.-X. Zhang, Y.-R. Cai, Clustering web documents based on Multiclass spectral clustering, in: Machine Learning and Cybernetics (ICMLC), 2011 International Conference on, 2011, pp. 1466-1471.
[85] M. Hemalatha, D. Sathyasrinivas, Hybrid neural network model for web document clustering, in: Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the, 2009, pp. 531-538.

[86] T.B. Ho, N.B. Nguyen, S. Kawasaki, Tolerance Rough Set Model Approach to Document Clustering, in: The Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2000, Boston, 2000, pp. 89-90.
[87] M.S. Hossain, R.A. Angryk, GDClust: A Graph-Based Document Clustering Technique, in: Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on, 2007, pp. 417-422.
[88] C. Hsinchun, C. Wingyan, Q. Yi, C. Michael, X. Jennifer Jie, W. Gang, Z. Rong, A. Homa, Crime data mining: an overview and case studies, in: Proceedings of the 2003 annual national conference on Digital government research, Digital Government Research Center, Boston, MA, 2003, pp. 1-5.
[89] InfoSpace, Dogpile Web Site, in, 2008.
[90] K. Inna Gelfer, K. Oren, Cluster-based query expansion, in: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, Boston, MA, USA, 2009.
[91] A.K. Jain, R.C. Dubes, Algorithms for clustering data, Prentice-Hall, Inc., 1988.
[92] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv., 31 (1999) 264-323.
[93] N.S. Jaishankar, A fast `parsing' algorithm for conceptual clustering, in: Applied Computing, 1991., [Proceedings of the 1991] Symposium on, 1991, pp. 471.
[94] L. Jing, Survey of Text Clustering, in, 2008.
[95] T. Joachims, F. Radlinski, Search Engines that Learn from Implicit Feedback, Computer, 40 (2007) 34-40.
[96] S. Jung, J.L. Herlocker, J. Webster, Click data as implicit relevance feedback in web search, Information Processing & Management, 43 (2007) 791-807.
[97] M. Kantardzic, Data Mining: Concepts, Models, Methods and Algorithms, John Wiley & Sons, 2003.
[98] H. Karanikas, B. Theodoulidis, Knowledge Discovery in Text and Text Mining Software, in, UMIST - CRIM, Manchester, 2002.
[99] G. Karypis, CLUTO - Software for Clustering High-Dimensional Datasets, Release 2.1.2., in, Department of Computer Science, University of Minnesota, 2006.
[100] J. Kennedy, R.C. Eberhart, Particle Swarm Optimization, in: IEEE Int'l. Conf. on Neural Networks., IEEE Press, Perth, Australia, 1995, pp. 1942–1948.
[101] W. Kim, L. Kerschberg, A. Scime, Learning for automatic personalization in a semantic taxonomy-based meta-search agent, Electronic Commerce Research and Applications, 1 (2002) 150-173.
[102] A. Kuri, J. Galaviz, Algoritmos Genéticos, Fondo de Cultura Económica/UNAM/IPM, 2002.
[103] R.K. Lai, C.-Y. Fan, W.-H. Huang, P.-C. Chang, Evolving and clustering fuzzy decision tree for financial time series data forecasting, Expert Systems with Applications, 36 (2009) 3761-3773.
[104] N.C. Lang, N.H. Son, A tolerance rough set approach to clustering web search results, in: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Springer-Verlag New York, Inc., Pisa, Italy, 2004.
[105] D.T. Larose, Discovering Knowledge in Data. An Introduction to Data Mining, John Wiley & Sons, Inc., 2005.
[106] D.T. Larose, Data Mining Methods and Models, John Wiley & Sons, Inc., 2006.
[107] I. Lee, B.-W. On, An effective web document clustering algorithm based on bisection and merge, Artif. Intell. Rev., 36 (2011) 69-85.
[108] B. Leuf, The Semantic Web: crafting infrastructure for agency, John Wiley & Sons, England, 2006.
[109] Y. Li, S.M. Chung, J.D. Holt, Text document clustering based on frequent word meaning sequences, Data & Knowledge Engineering, 64 (2008) 381-404.
[110] Y. Li, C. Luo, S.M. Chung, Text Clustering with Feature Selection by Using Statistical Data, Knowledge and Data Engineering, IEEE Transactions on, 20 (2008) 641-652.
[111] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman & Hall/CRC, 2007.

[112] X. Liu, P. He, A Study on Text Clustering Algorithms Based on Frequent Term Sets, in: Advanced Data Mining and Applications, 2005, pp. 347-354.
[113] A. Lozano, Ontologías en la Web Semántica, in: I Jornadas de Ingeniería Web' 01, 2001.
[114] J. Madrid, S. Gauch, Incorporating Conceptual Matching in Search, in: Conference on Information and Knowledge Management McLean, VA 2002.
[115] G.H.O. Mahamed, P.E. Andries, S. Ayed, An overview of clustering methods, Intell. Data Anal., 11 (2007) 583-605.
[116] M. Mahdavi, H. Abolhassani, Harmony K-means algorithm for document clustering, Data Mining and Knowledge Discovery, 18 (2009) 370-391.
[117] M. Mahdavi, M.H. Chehreghani, H. Abolhassani, R. Forsati, Novel meta-heuristic algorithms for clustering web documents, Applied Mathematics and Computation, 201 (2008) 441-451.
[118] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Applied Mathematics and Computation, 188 (2007) 1567-1579.
[119] C. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, in, Cambridge University Press, Cambridge, England, 2008.
[120] B. Marin, G. Rachid, L. Vincent, K. Anne-Marie, Toward personalized query expansion, in: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, ACM, Nuremberg, Germany, 2009.
[121] T. Matsumoto, E. Hung, Fuzzy clustering and relevance ranking of web search results with differentiating cluster label generation, in: Fuzzy Systems (FUZZ), 2010 IEEE International Conference on, 2010, pp. 1-8.
[122] G. Mecca, S. Raunich, A. Pappalardo, A new algorithm for clustering search results, Data & Knowledge Engineering, 62 (2007) 504-522.
[123] Merlot, MERLOT Multimedia Educational Resource for Learning and Online Teaching, in.
[124] D. Miao, Q. Duan, H. Zhang, N. Jiao, Rough set based hybrid algorithm for text classification, Expert Systems with Applications, 36 (2009) 9168-9174.
[125] C. Michael, F. Xiao, R.L.S. Olivia, Analysis of the query logs of a web site search engine, J. Am. Soc. Inf. Sci. Technol., 56 (2005) 1363-1376.
[126] W.B. Michael, C. Malu, Survey of Text Mining II: Clustering, Classification, and Retrieval, 2008.
[127] R.S. Michalski, R.E. Stepp, Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy, Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-5 (1983) 396-410.
[128] M. Mitchell, An Introduction to Genetic Algorithms, The MIT Press, Cambridge, MA, USA, 1999.
[129] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press 1999.
[130] Y.H. Montero, Factores del Diseño Web Orientado a la Satisfacción y No-Frustración de Uso, Revista Española de Documentación Científica, (2006) 239-257.
[131] J. Moore, E.-H.S. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering, in: Workshop on Information Technologies and Systems, 1997.
[132] J. Mustafa, S. Khan, K. Latif, Ontology based semantic information retrieval, in: Intelligent Systems, 2008. IS '08. 4th International IEEE Conference, 2008, pp. 22-14-22-19.
[133] R. Navigli, G. Crisafulli, Inducing word senses to improve web search result clustering, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Cambridge, Massachusetts, 2010, pp. 116-126.
[134] Q.H. Nguyen, Y.S. Ong, N. Krasnogor, A study on the design issues of Memetic Algorithm, in: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, 2007, pp. 2390-2397.
[135] T.C. Nguyen, T.T. Phan, An Ontology-Based Approach of Query Expansion, in: G. Kotsis, D. Taniar, E. Pardede, I. Khalil Ibrahim (Eds.) iiWAS'2007 - The Ninth International

Conference on Information Integration and Web-based Applications Services, Jakarta, Indonesia, 2007.
[136] J. Nielsen, When Search Engines Become Answer Engines, in, 2004.
[137] N.F. Noy, L. Deborah, Ontology Development 101: A Guide to Creating Your First Ontology, in.
[138] K. O'Hara, N. Shadbolt, Knowledge Technologies and the Semantic Web in, 2004.
[139] OCLC, OCLC Online Computer Library Center, in.
[140] OCLC, OCLC Online Computer Library Center Dewey Decimal Classification, in.
[141] M.G.H. Omran, A.P. Engelbrecht, A. Salman, Bare bones differential evolution, European Journal of Operational Research, 196 (2009) 128-139.
[142] M.G.H. Omran, M. Mahdavi, Global-best harmony search, Applied Mathematics and Computation, 198 (2008) 643-656.
[143] H. Ordoñez, C. Cobos, E. León, Modelo de un meta-buscador web semántico basado en una taxonomía general de conocimiento, una ontología de dominio general, ontologías específicas y perfil de usuario, Revista UIS Ingenierías, 10 (2011) 23-38.
[144] Z. Oren, E. Oren, Web document clustering: a feasibility demonstration, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, Melbourne, Australia, 1998, pp. 46-54.
[145] S. Osiński, An Algorithm for clustering of web search results, in, Poznań University of Technology, Poland, 2003, pp. 91.
[146] S. Osiński, Improving quality of search results clustering with approximate matrix factorizations, in: 28th European Conference on IR Research (ECIR 2006), London, UK, 2006, pp. 167-178.
[147] S. Osiński, J. Stefanowski, D. Weiss, Lingo: Search results clustering algorithm based on Singular Value Decomposition, in: Proceedings of the International Conference on Intelligent Information Systems (IIPWM), Springer, 2004, pp. 359-368.
[148] S. Osiński, D. Weiss, Conceptual clustering using Lingo algorithm: Evaluation on Open Directory Project data, in: Proceedings of the International Conference on Intelligent Information Systems (IIPWM), 2004, pp. 369-377.
[149] S. Osiński, D. Weiss, Carrot 2: Design of a Flexible and Efficient Web Information Retrieval Framework, in: Advances in Web Intelligence, 2005, pp. 439-444.
[150] S. Osiński, D. Weiss, A concept-driven algorithm for clustering search results, Intelligent Systems, IEEE, 20 (2005) 48-54.
[151] B. Panigrahi, V. Pandi, S. Das, A. Abraham, Population Variance Harmony Search Algorithm to Solve Optimal Power Flow with Non-Smooth Cost Function, in: Recent Advances In Harmony Search Algorithm, Springer Berlin / Heidelberg, 2010, pp. 65-75.
[152] R.Y. Pawan Lingras, Chad West, Interval Set Clustering of Web Users with Rough K-Means, in, Saint Mary's University, 2003.
[153] K.S. Pratt, Design Patterns for Research Methods: Iterative Field Research, in: Association for the Advancement of Artificial Intelligence, 2009.
[154] H. Ralambondrainy, A conceptual version of the K-means algorithm, Pattern Recognition Letters, 16 (1995) 1147-1157.
[155] P. Rattadilok, An Investigation and Extension of a Hyper-heuristic Framework, Informatica, 34 (2010) 523-534.
[156] S.J. Redmond, C. Heneghan, A method for initialising the K-means clustering algorithm using kd-trees, Pattern Recognition Letters, 28 (2007) 965-973.
[157] S.C.f.B.I. Research, The Protégé Ontology Editor and Knowledge Acquisition System, in, 2008.
[158] C.J.V. Rijsbergen, Information Retrieval, Butterworth-Heinemann, 1979.
[159] S.E. Robertson, K. Sparck-Jones, Relevance weighting of search terms, in: Document retrieval systems, Taylor Graham Publishing, 1988, pp. 143-160.
[160] J.a.S. Rocchio, G., Relevance feedback in information retrieval, 1971.
[161] T. Rolleke, T. Tsikrika, G. Kazai, A general matrix framework for modeling Information Retrieval, Information Processing & Management, 42 (2006) 4-30.

[162] R.C. Romero-Zaliz, C. Rubio-Escudero, J.P. Cobb, F. Herrera, O. Cordon, I. Zwir, A Multiobjective Evolutionary Conceptual Clustering Methodology for Gene Annotation Within Structural Databases: A Case of Study on the <emphasis emphasistype="italic">Gene Ontology</emphasis> Database, Evolutionary Computation, IEEE Transactions on, 12 (2008) 679-701.
[163] K. Ron, H.J. George, Wrappers for feature subset selection, Artif. Intell., 97 (1997) 273-324.
[164] G. Sacco, Dynamic taxonomies: a model for large information bases, Knowledge and Data Engineering, IEEE Transactions on, 12 (2000) 468-479.
[165] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing & Management, 24 (1988) 513-523.
[166] D. Sánchez, J. Cavero, E. Marcos, Ontologías y MDA: una revisión de la literatura, in.
[167] U. Scaiella, P. Ferragina, A. Marino, M. Ciaramita, Topical clustering of search results, in: Proceedings of the fifth ACM international conference on Web search and data mining, ACM, Seattle, Washington, USA, 2012, pp. 223-232.
[168] C. Scott Shaobing, P.S. Gopalakrishnan, Clustering via the Bayesian information criterion with applications in speech recognition, in: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, 1998, pp. 645-648 vol.642.
[169] J. Sedding, D. Kazakov, WordNet-based text document clustering, in: V.P.a.A. Todirascu (Ed.) Proceedings of COLING 2004 3rd Workshop on Robust Methods in Analysis of Natural Language Data, COLING, Geneva, Switzerland, 2004, pp. 104-113.
[170] B. Sheth, P. Maes, Evolving agents for personalized information filtering, in: Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on, Orlando, FL, USA, 1993, pp. 345-352.
[171] O.M. Shir, Niching in Derandomized Evolution Strategies and its Applications in Quantum Control, in, Leiden, 2008, pp. 256.
[172] J.-f. Song, W.-m. Zhang, W.-d. Xiao, G.-h. Li, Z.-n. Xu, Ontology-Based Information Retrieval Model for the Semantic Web, in: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service, IEEE Computer Society, 2005.
[173] W. Song, C.H. Li, S.C. Park, Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures, Expert Systems with Applications, 36 (2009) 9095-9104.
[174] W. Song, S. Park, Genetic Algorithm-Based Text Clustering Technique, in: Advances in Natural Computation, 2006, pp. 779-782.
[175] A. Spink, M. Park, B.J. Jansen, J. Pedersen, Multitasking during Web search sessions, Information Processing & Management, 42 (2006) 264-275.
[176] A. Spink, M. Park, S. Koshman, Factors affecting assigned information problem ordering during Web search: An exploratory study, Information Processing & Management, 42 (2006) 1366-1378.
[177] A. Spink, J.L. Xu, Selected results from a large study of Web searching: the Excite study, Information Research, 6 (2000).
[178] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: KDD workshop on text mining, ACM Boston, MA, USA., 2000, pp. 1-20.
[179] Y. Stekh, F.M.E. Sardieh, M. Lobur, M. Dombrova, Algorithm for clustering web documents, in: Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2010 Proceedings of VIth International Conference on, 2010, pp. 187-187.
[180] K.P. Supreethi, E.V. Prasad, Web Document Clustering Technique Using Case Grammar Structure, in: Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on, 2007, pp. 98-102.
[181] G. Susan, BDEI: Biodiversity Information Organization using Taxonomy (BIOT), in: Proceedings of the 2002 annual national conference on Digital government research, Digital Government Research Center, Los Angeles, California, 2002.

[182] G. Susan, C. Jason, P. Alexander, Ontology-based personalized search and browsing, Web Intelli. and Agent Sys., 1 (2003) 219-234.
[183] B.S. Sven Meyer zu Eissen, Martin Potthast, The Suffix Tree Document Model Revisited, Journal of Universal Computer Science, 596-603.
[184] L. Talavera, J. Bejar, Generality-based conceptual clustering with probabilistic concepts, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23 (2001) 196-206.
[185] Y. Tan, Y. Shi, K. Tan, H. Jiang, Y. Liu, L. Zheng, Design and Simulation of Simulated Annealing Algorithm with Harmony Search, in: Advances in Swarm Intelligence, Springer Berlin / Heidelberg, 2010, pp. 454-460.
[186] S. Tekir, F. Mansmann, D. Keim, Geodesic distances for web document clustering, in: Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on, 2011, pp. 15-21.
[187] R.G. Thomas, Toward principles for the design of ontologies used for knowledge sharing, Int. J. Hum.-Comput. Stud., 43 (1995) 907-928.
[188] USA.GOV, Web Site of the Library of Congress of United States of America, in.
[189] S. Vijay, M. Maheshwari, R. Ali, Search Engine: A Review., Oriental Journal of Computer Science & Technology, 6 (2013) 341.
[190] J. Wang, Y. Mo, B. Huang, J. Wen, L. He, Web Search Results Clustering Based on a Novel Suffix Tree Structure, in: Autonomic and Trusted Computing, 2008, pp. 540-554.
[191] A. Webb, Statistical Pattern Recognition, 2nd Edition, {John Wiley & Sons}, 2002.
[192] X. Wei, L. Xin, G. Yihong, Document clustering based on non-negative matrix factorization, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, Toronto, Canada, 2003, pp. 267-273.
[193] C. Welty, The Ontological Nature of Subject Taxonomies, in: Formal Ontology in Information Systems. IOS Press Frontiers in AI Applictions Series, N. Guarino, Trento, Italy, 1998, pp. 317-327.
[194] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Francisco, CA, 2005.
[195] P. Wohl, T.W. Christopher, A parallel processing approach to incremental conceptual clustering, in: Parallel Processing Symposium, 1991. Proceedings., Fifth International, 1991, pp. 240-245.
[196] D. Wolfram, Search characteristics in different types of Web-based IR environments: Are they the same?, Information Processing & Management, 44 (2008) 1279-1292.
[197] L. Xiang-Wei, H. Pi-Lian, W. Hui-Ying, The research of text clustering algorithms based on frequent term sets, in: Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, 2005, pp. 2352-2356 Vol. 2354.
[198] Y. Xin-She, S. Deb, Cuckoo Search via Lévy flights, in: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, 2009, pp. 210-214.
[199] Yahoo, Yahoo Directory Web Site, in.
[200] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, (2008) 128.
[201] L. Yongli, L. Chao, Z. Pin, X. Zhang, A Query Expansion Algorithm Based on Phrases Semantic Similarity, in: Proceedings of the 2008 International Symposiums on Information Processing, IEEE Computer Society, 2008.
[202] X. Yuni, X. Bowei, Conceptual Clustering Categorical Data with Uncertainty, in: Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, 2007, pp. 329-336.
[203] D. Zhang, Y. Dong, Semantic, Hierarchical, Online Clustering of Web Search Results, in: Advanced Web Technologies and Applications, 2004, pp. 69-78.
[204] Q. Zhang, Q. Peng, T. Xu, DNA splice site sequences clustering method for conservativeness analysis, Progress in Natural Science, In Press, Corrected Proof.
[205] S. Zheng, X. Zhao, B. Zhang, H. Bu, Web Document Clustering Research Based on Granular Computing, in: Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on, 2009, pp. 446-450.

[206] C. Zhi, G. Huan-Tong, Z. Xin, C. Qing-Sheng, An algorithm for conceptual clustering of Chinese text, in: Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, 2004, pp. 3035-3039 vol.3035.

[207] Z. Zhong-Yuan, J. Zhang, Survey on the Variations and Applications of Nonnegative Matrix Factorization, in: ISORA'10: The Ninth International Symposium on Operations Research and Its Applications, ORSC & APORC, Chengdu-Jiuzhaigou, China, 2010, pp. 317–323.

[208] L. Zhuhadar, O. Nasraoui, Semantic Information Retrieval for Personalized E-Learning, in: Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on, 2008, pp. 364-368.

[209] L. Zhuhadar, O. Nasraoui, R. Wyatt, Dual Representation of the Semantic User Profile for Personalized Web Search in an Evolving Domain, in: AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0, 2009, pp. 84-89.

[210] L. Zhuhadar, O. Nasraoui, R. Wyatt, Visual Ontology-Based Information Retrieval System, in: Proceedings of the 2009 13th International Conference Information Visualisation, IEEE Computer Society, 2009, pp. 419-426.

[211] L. Zhuhadar, O. Nasraoui, R. Wyatt, E. Romero, Multi-model Ontology-Based Hybrid Recommender System in E-learning Domain, in: Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on, 2009, pp. 91-95.

[212] D. Zou, L. Gao, J. Wu, S. Li, Y. Li, A novel global harmony search algorithm for reliability problems, Computers & Industrial Engineering, 58 (2010) 307-316.

Modelo de un Meta Buscador que Realiza Agrupación de Documentos Web, Enriquecido con una Taxonomía, Ontologías e Información del Usuario

Anexos

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá D.C., Colombia

2013

Appendix A

Title	A harmony search algorithm for clustering with feature selection
Journal	Rev. Fac. Ing. Univ. Antioquia - Universidad de Antioquia
Class	ISI Journal (rated category A1 by PUBLINDEX-COLCIENCIAS)
Send date	April 29, 2009
Status	Published (September 2010)
Note	Volume 55. pp. 153-164. ISSN: 0120-6230

A harmony search algorithm for clustering with feature selection

Un algoritmo de búsqueda armónica para clustering con selección de características

Carlos Cobos[1,2], Elizabeth León[2], Martha Mendoza[1]*

[1]Information Technology Research Group (GTI), Electronic and Telecommunications Engineering Faculty, University of Cauca, Sector Tulcán Office 422 FIET, Popayán, Colombia.

[2]Research Laboratory of Intelligent Systems (LISI), National University of Colombia, Bogotá, Colombia.

(Recibido el 29 de Abril de 2009. Aceptado el 6 de abril de 2010)

Abstract

This paper presents a new clustering algorithm, called IHSK, with feature selection in a linear order of complexity. The algorithm is based on the combination of the harmony search and K-means algorithms. Feature selection uses both the concept of variability and a heuristic method that penalizes the presence of dimensions with a low probability of contributing to the current solution. The algorithm was tested with sets of synthetic and real data, obtaining promising results.

----- *Keywords:* harmony search, clustering, feature selection

Resumen

En este artículo se presenta un nuevo algoritmo de clustering denominado IHSK, con la capacidad de seleccionar características en un orden de complejidad lineal. El algoritmo es inspirado en la combinación de los algoritmos de búsqueda armónica y K-means. Para la selección de las características se usó el concepto de variabilidad y un método heurístico que penaliza la presencia de dimensiones con baja probabilidad de aportar en la solución actual. El algoritmo fue probado con conjuntos de datos sintéticos y reales, obteniendo resultados prometedores.

----- *Palabras clave*: búsqueda armónica, agrupamiento, selección de características

* Autor de correspondencia: teléfono: + 57 + 2 + 820 98 00 ext. 2119, fax: + 57 + 2 + 820 98 00 ext. 2102, correo electrónico: ccobos@unicauca.edu.co. (C. Cobos)

Appendix B

Title	Algoritmos de Expansión de Consulta basados en una Nueva Función Discreta de Relevancia
Journal	Revista UIS Ingenierías - Universidad Industrial de Santander
Class	EBSCO Journal (rated category B by PUBLINDEX-COLCIENCIAS)
Send date	March 31, 2011
Status	Published (June 2011)
Note	Volume 10. Issue 1. pp. 7-20. ISSN: 1657-4583

Algoritmos de expansión de consulta basados en una nueva función discreta de relevancia

CARLOS ALBERTO COBOS LOZADA
Ingeniero de Sistemas, Magíster en Informática, Ph.D. (c) en Ingeniería de Sistemas y Computación
Profesor Titular, Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones
Director del Grupo de I+D en Tecnologías de la Información, Universidad del Cauca
ccobos@unicauca.edu.co
Popayán, Cauca, Colombia

EDUARDO ESTEVEZ MENDOZA
Estudiante de Ingeniería de Sistemas
Programa de Ingeniería de Sistemas, Escuela de Ingeniería de Sistemas e Informática
Miembro del Grupo de I+D en Sistemas y Tecnologías de la Información, Universidad Industrial de Santander
eestevez25@hotmail.com
Bucaramanga, Santander, Colombia

MARTHA ELIANA MENDOZA BECERRA
Ingeniera de Sistemas, Magíster en Informática, Estudiante de Doctorado en Ingeniería de Sistemas y Computación
Profesora Titular, Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones
Miembro del Grupo de I+D en Tecnologías de la Información, Universidad del Cauca
mmendoza@unicauca.edu.co
Popayán, Cauca, Colombia

LUIS CARLOS GÓMEZ FLÓREZ
Ingeniero de Sistemas, Magíster en Informática
Profesor Titular, Escuela de Ingeniería de Sistemas e Informática, Facultad de Ingenierías Físico Mecánicas
Director del Grupo de I+D en Sistemas y Tecnologías de la Información, Universidad Industrial de Santander
lcgomezf@uis.edu.co
Bucaramanga, Santander, Colombia

ELIZABETH LEÓN GUZMÁN
Ingeniera de Sistemas, Magíster en Ingeniería de Sistemas, Ph.D. in Computer Science and Computer Engineering
Profesora Asistente, Departamento de Ingeniería de Sistemas e Industrial, Facultad de Ingeniería
Directora del Grupo de Investigación en Minería de Datos, Universidad Nacional de Colombia
eleonguz@unal.edu.co
Bogotá D.C., Colombia

Fecha de recibido: 31/03/2011
Fecha de aprobación: 15/06/2011

Appendix C

Title	Modelo de un Meta Buscador Web Semántico Basado en una Taxonomía General de Conocimiento, una Ontología de Dominio General, Ontologías Especificas y Perfil de Usuario
Journal	Revista UIS Ingenierías - Universidad Industrial de Santander
Class	EBSCO Journal (rated category B by PUBLINDEX-COLCIENCIAS)
Send date	October 12, 2010
Status	Published (June 2011)
Note	Volume 10. Issue 1. pp. 23-38. ISSN: 1657-4583

Modelo de un meta-buscador web semántico basado en una taxonomía general de conocimiento, una ontología de dominio general, ontologías específicas y perfil de usuario

HUGO ORDOÑEZ ERASO
Ingeniero de Sistemas, Magíster en Computación
Profesor, Facultad de Ingeniería, Universidad Mariana
Miembro del Grupo de I+D en Tecnologías de la Información (GTI), Universidad del Cauca
hugoeraso@gmail.com
San Juan de Pasto, Nariño, Colombia

CARLOS ALBERTO COBOS LOZADA
Ingeniero de Sistemas, Magíster en Informática, Ph.D. (c) en Ingeniería de Sistemas y Computación
Profesor Titular, Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones
Director del Grupo de I+D en Tecnologías de la Información (GTI), Universidad del Cauca
ccobos@unicauca.edu.co
Popayán, Cauca, Colombia

ELIZABETH LEÓN GUZMÁN
Ingeniera de Sistemas, Magíster en Ingeniería de Sistemas, Ph.D. en Ciencias e Ingeniería de la Computación
Profesora Asistente, Departamento de Ingeniería de Sistemas e Industrial, Facultad de Ingeniería
Directora del Grupo de I+D en Minería de Datos (MIDAS), Universidad Nacional de Colombia sede Bogotá
eleonguz@unal.edu.co
Bogotá, Colombia

*Fecha de recibido:*12/10/2010
Fecha de aprobación: 15/06/2011

RESUMEN

La búsqueda web en los últimos años se ha convertido en una de las áreas de investigación más importantes del mundo, debido entre otras cosas: al crecimiento acelerado de las fuentes de información, a la necesidad de contar con información más relevante a los requerimientos específicos de cada usuario, a la exploración de menores tiempos de búsqueda y a la falta de usar la semántica de los términos implicados en las consultas. En este artículo se presenta el modelo de un meta-buscador (usa los recursos indexados por Google, Yahoo! y Bing) web semántico llamado XGhobi, que incorpora una taxonomía general de conocimiento, una ontología de dominio general (WordNet), un conjunto de ontologías de dominio específico y el perfil de los usuarios para mejorar la relevancia de los documentos recuperados tanto en inglés como en español. Se describe en detalle los componentes del meta-buscador, algunas interfaces de usuario y los resultados de su evaluación. La evaluación del sistema muestra la precisión obtenida en pruebas realizadas con usuarios.

PALABRAS CLAVE: Meta-buscador web, Taxonomía, Ontología, WordNet, Perfil de usuario.

ABSTRACT

Web search has become one of the most important fields of research around the world. They are many reasons including: the fast-growing nature of information sources; the search necessity for information closer to specific user requirements; the need to reduce search time; and the desire to take into account the semantics of terms used when doing search queries. This paper shows a semantic meta-web search model called XGhobi which uses indexed resources by Google, Yahoo! and Bing. The XGhobi engine combines a general taxonomy of knowledge, a general domain ontology –WordNet-, a set of specific domain ontologies, and user profile management to improve the relevance of recovered documents in both English and Spanish. A detailed description of the meta-web search engine's components, some user interfaces and its results and its assessments are shown. The assessment covers the obtained precision on tests done by users.

KEYWORDS: Meta-web searcher, Taxonomy, Ontology, WordNet, User profile.

Appendix D

Title	Fitness Function Obtained from a Genetic Programming Approach for Web Document Clustering Using Evolutionary Algorithms
Journal	Lecture Notes in Computer Science (LNCS), Subseries: Lecture Notes in Artificial Intelligence (LNAI)
Class	Springer Journal (rated category C by PUBLINDEX-COLCIENCIAS)
Send date	May 20, 2012
Status	Published (November 2012)
Note	Volume LNAI 7637. pp. 179-188. ISSN: 0302-9743 (Print) 1611-3349 (Online).

Fitness Function Obtained from a Genetic Programming Approach for Web Document Clustering Using Evolutionary Algorithms

Carlos Cobos[1], Leydy Muñoz[1], Martha Mendoza[1], Elizabeth León[2], and Enrique Herrera-Viedma[3]

[1] Computer Science Department, Universidad del Cauca, Colombia
{ccobos,cmunoz,mmendoza}@unicauca.edu.co
[2] Systems and Industrial Engineering Department, Engineering Faculty, Universidad Nacional de Colombia, Colombia
eleonguz@unal.edu.co
[3] Department of Computer Science and Artificial Intelligence, University of Granada, Spain
viedma@decsai.ugr.es

Abstract. Web document clustering (WDC) is an alternative means of searching the web and has become a rewarding research area. Algorithms for WDC still present some problems, in particular: inconsistencies in the content and description of clusters. The use of evolutionary algorithms is one approach for improving results. It uses standard index to evaluate the quality (as a fitness function) of different solutions of clustering. Indexes such as Bayesian Information Criteria (BIC), Davies-Bouldin, and others show good performance, but with much room for improvement. In this paper, a modified BIC fitness function for WDC based on evolutionary algorithms is presented. This function was discovered using a genetic program (from a reverse engineering view). Experiments on datasets based on DMOZ show promising results.

Keywords: genetic programming, web document clustering, clustering of web results, Bayesian information criteria.

1 Introduction

In recent years, web document clustering (WDC) -clustering of web results- has become a very interesting research area [1]. Web document clustering systems seek to increase the coverage (amount) of documents presented for the user to review, while reducing the time spent in reviewing documents [2]. Web document clustering systems are called web clustering engines. Among the most prominent are Carrot, SnakeT, Yippy, KeySRC and iBoogie [3]. Such systems usually consist of four main components: search results acquisition, preprocessing of input, construction and labeling of clusters, and visualization of resulting clusters [1].

The **search results acquisition** component begins with a query defined by the user. Based on this query, a document search is conducted in diverse data sources, in this case in traditional web search engines such as Google, Yahoo! and Bing. In

J. Pavón et al. (Eds.): IBERAMIA 2012, LNAI 7637, pp. 179–188, 2012.
© Springer-Verlag Berlin Heidelberg 2012

Appendix E

Title	TopicSearch - Personalized Web Clustering Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents
Journal	Polibits Journal
Class	Scielo Journal (rated category A1 by PUBLINDEX-COLCIENCIAS)
Send date	March 13, 2013
Status	Published
Note	Volume 47. pp. 33-46. ISSN: 1870-9044.

TopicSearch - Personalized Web Clustering Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents

Carlos Cobos, Martha Mendoza, Elizabeth León, Milos Manic, Enrique Herrera-Viedma

***Abstract*— As resources become more and more available on the Web, so the difficulties associated with finding the desired information increase. Intelligent agents can assist users in this task since they can search, filter and organize information on behalf of their users. Web document clustering techniques can also help users to find pages that meet their information requirements. This paper presents a personalized web document clustering called TopicSearch. TopicSearch introduces a novel inverse document frequency function to improve the query expansion process, a new memetic algorithm for web document clustering, and frequent phrases approach for defining cluster labels. Each user query is handled by an agent who coordinates several tasks including query expansion, search results acquisition, preprocessing of search results, cluster construction and labeling, and visualization. These tasks are performed by specialized agents whose execution can be parallelized in certain instances. The model was successfully tested on fifty DMOZ datasets. The results demonstrated improved precision and recall over traditional algorithms (k-means, Bisecting k-means, STC y Lingo). In addition, the presented model was evaluated by a group of twenty users with 90% being in favor of the model.**

***Index Terms*— Web Document Clustering; Intelligent Agents; Query Expansion; WordNet; Memetic Algorithm; User Profile.**

I. Introduction

IN recent years, web document clustering has become a very interesting research area among academic and scientific communities involved in information retrieval (IR) and web search [1]. Web document clustering systems seek to increase the coverage (amount) of documents presented for the user to review, while reducing the time spent in reviewing documents [2]. In IR, these web document clustering systems are called web clustering engines. Among the most prominent ones are Carrot (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, originally named as Vivisimo and then as Clusty), iBoogie (www.iboogie.com), and KeySRC (http://keysrc.fub.it) [3]. Such systems usually consist of four main components: search results acquisition, preprocessing of input, cluster construction and labeling, and visualization of resulting clusters [1] (see **Fig 1**).

The **search results acquisition** component begins with a query defined by the user. Based on this query, a document search is conducted in diverse data sources, in this case in the traditional web search engines such as Google, Yahoo! and Bing. In general, web clustering engines work as meta search engines and collect between 50 to 200 results from traditional search engines. These results contain as a minimum a URL, a snippet and a title [1].

The **preprocessing** of search results comes next. This component converts each of the search results (as snippets) into a sequence of words, phrases, strings or general attributes or characteristics, which are then used by the clustering algorithm. There are a number of tasks performed on the search results, including: removing special characters and accents, the conversion of the string to lowercase, removing stop words, stemming of the words and the control of terms or concepts allowed by a vocabulary [1].

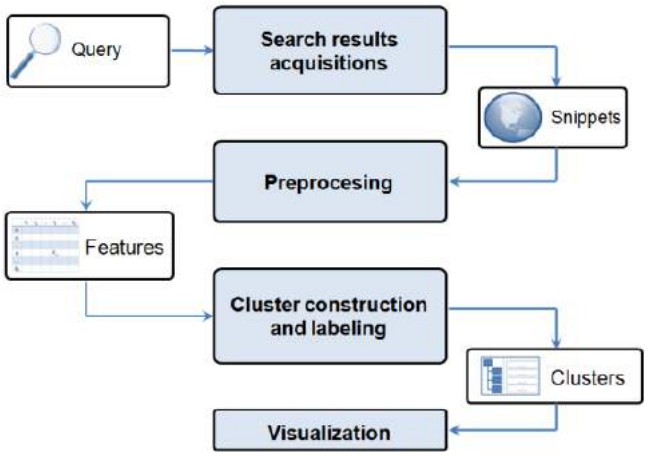


Fig 1.The components of a web clustering engine (adapted from [1])

Once the preprocessing is finished, **cluster construction and labeling** is begun. This stage makes use of three types of

This work was supported by a Research Grant from the University of Cauca under Project VRI-2560 and the National University of Colombia.

Carlos Cobos is with University of Cauca (phone: 57-2-8209800x2119; fax: 57-2-8209810; e-mail: ccobos@unicauca.edu.co).

Martha Mendoza is with University of Cauca (e-mail: mmendoza@unicauca.edu.co).

Elizabeth León is with Universidad Nacional de Colombia (e-mail: eleonguz@unal.edu.co).

Milos Manic is with University of Idaho at Idaho Falls (e-mail: misko@uidaho.edu)

Enrique Herrera-Viedma is with University of Granada (e-mail: viedma@decsai.ugr.es)

Appendix F

Title	Clustering of Web Search Results based on the Cuckoo Search Algorithm and Balanced Bayesian Information Criterion
Journal	Information Sciences
Class	ISI Journal (rated category A1 by PUBLINDEX-COLCIENCIAS)
Send date	April 26, 2013
Status	In the evaluation process (second revision)
Note	Volume. pp. ISSN: 0020-0255.

Clustering of Web Search Results based on the Cuckoo Search Algorithm and Balanced Bayesian Information Criterion

Carlos Cobos [a,b,*], Henry Muñoz-Collazos [a], Richar Urbano-Muñoz [a], Martha Mendoza [a,b], Elizabeth León [c], Enrique Herrera-Viedma [d]

[a] *Information Technology Research Group (GTI) members, Universidad del Cauca, Sector Tulcán Office 422 FIET, Popayán, Colombia*

[b] *Full time professor, Computer Science Department, Electronic and Telecommunications Engineering Faculty, Universidad del Cauca, Colombia*

[c] *Full time professor, Systems and Industrial Engineering Department, Engineering Faculty, Universidad Nacional de Colombia, Colombia*

[d] Full time professor, Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain

[*] Corresponding author: Carlos Cobos; Tel: 57-2-8209800 #2119; Fax: 57-2-8209810; e-mail: ccobos@unicauca.edu.co

Abstract

The clustering of web search results –or web document clustering- has become a very interesting research area among academic and scientific communities involved in information retrieval. Clustering of web search result systems, also called Web Clustering Engines, seeks to increase the coverage of documents presented for the user to review, while reducing the time spent reviewing them. Several algorithms for clustering of web results already exist, but results show there is room for more to be done. This paper introduces a new description-centric algorithm for clustering of web results, called WDC-CSK, which is based on the cuckoo search meta-heuristic algorithm, k-means algorithm, Balanced Bayesian information criterion, split and merges methods on clusters, and frequent phrases approach for cluster labeling. The cuckoo search meta-heuristic provides a combined global and local search strategy in the solution space. Split and merge methods replace the original Lévy flights operation and they try to improve existing solutions (nests), so they can be considered as local search methods. WDC-CSK includes an abandon operation which provides diversity and prevents the population nests converging too quickly. Balanced Bayesian information criterion is used as a fitness function and it allows defines the number of clusters automatically. WDC-CSK was tested with four data sets, namely: DMOZ-50, AMBIENT, MORESQUE and ODP-239 over 447 queries. The algorithm was also compared against other established web document clustering algorithms, among them: Suffix Tree Clustering (STC), Lingo, and Bisecting k-means. Results show a considerable improvement measured by recall, F-measure, fall-out, accuracy and SSL_k, over the other algorithms.

© 2014 Elsevier Ltd. All rights reserved.

Keywords: Cuckoo search algorithm, clustering of web results, web document clustering, balanced Bayesian information criterion, k-means.

1 Introduction

In recent years, clustering of web results has become a very interesting research area among academic and scientific communities involved in information retrieval (IR) and web search [12] since it is very likely that the results relevant to the user are close to each other in the document space, and thus tending to fall into a relatively small number of clusters [44], and thereby achieve a significant reduction of search time. In IR, these clustering of web result systems are called web clustering engines and the main exponents in the field are Carrot2 (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, originally known as Vivisimo and later as Clusty), iBoogie (www.iboogie.com), and KeySRC (http://keysrc.fub.it) [11]. Such systems usually consist of four main components, namely: search results acquisition, processing of input, cluster construction and labeling, and visualization of resulting clusters [12] (see Fig 1).

Appendix G

Title	Algorithm for clustering of web search results from a hyper-heuristic approach
Journal	Applied Soft Computing
Class	ISI Journal (rated category A1 by PUBLINDEX-COLCIENCIAS)
Send date	June 5, 2013
Status	In the evaluation process
Note	Volume. pp. ISSN: 1568-4946.

Algorithm for clustering of web search results from a hyper-heuristic approach

Carlos Cobos [a,b,*], Andrea Duque [a], Jamith Bolaños [a], Martha Mendoza [a,b], Elizabeth León [c]

[a] *Information Technology Research Group (GTI) members, Universidad del Cauca, Sector Tulcán Office 422 FIET, Popayán, Colombia*

[b] *Full time professor, Computer Science Department, Electronic and Telecommunications Engineering Faculty, Universidad del Cauca, Colombia*

[c] *Full time professor, Systems and Industrial Engineering Department, Engineering Faculty, Universidad Nacional de Colombia, Colombia*

[*] Corresponding author: Carlos Cobos; Tel: 57-2-8209800 #2119; Fax: 57-2-8209810; e-mail: ccobos@unicauca.edu.co

Abstract

The clustering of web search results - or web document clustering - has become a very interesting research area among academic and scientific communities involved in information retrieval. Systems for the clustering of web search results, also called Web Clustering Engines, seek to increase the coverage of documents presented for the user to review, while reducing the time spent reviewing them. Several algorithms for clustering of web results already exist, but results show there is room for more to be done. This paper introduces a hyper-heuristic framework called WDC-HH, which allows the defining of the best algorithm for web document clustering. The hyper-heuristic framework uses four high-level-heuristics (performance-based rank selection, tabu selection, random selection and performance-based roulette wheel selection) for selecting low-level heuristics (used to solve the specific problem of web document clustering). As a low level heuristics the framework considers: harmony search, improved harmony search, novel global harmony search, global-best harmony search, eighteen genetic algorithm variations, particle swarm optimization, artificial bee colony, and differential evolution. The framework uses the k-means algorithm as a local solution improvement strategy and based on the Balanced Bayesian Information Criterion it is able to automatically define the appropriate number of clusters. The framework also uses four acceptance/replacement strategies (replacement heuristics): Replace the worst, Restricted Competition Replacement, Stochastic Replacement and Rank Replacement. WDC-HH was tested with four data sets: DMOZ-50, AMBIENT, MORESQUE and ODP-239, for a total of 447 queries with their ideal solutions. As a main result of the framework assessment, a new algorithm based on global-best harmony search and rank replacement strategy obtained the best results in web document clustering problem. This new algorithm was called WDC-HH-BHRK and was also compared against other established web document clustering algorithms, among them: Suffix Tree Clustering (STC) and Lingo. Results show a considerable improvement -measured by recall, F-measure, fall-out, accuracy and SSL_k- over the other algorithms.

© 2013 Elsevier Ltd. All rights reserved.

Keywords: Hiper-heuristics, clustering of web results, web document clustering, balanced Bayesian information criterion, k-means.

1 Introduction

In recent years, clustering of web results has become a very interesting research area among academic and scientific communities involved in information retrieval (IR) and web search [1] since it is very likely that the results relevant to the user are close to each other in the document space, thus tending to fall into a relatively small number of clusters [2] and thereby achieve a significant reduction of search time. In IR, these clustering of web result systems are called web clustering engines and the main exponents in the field are Carrot^2 (http://www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, originally known as Vivisimo and later as Clusty), iBoogie (http://www.iboogie.com), and KeySRC

Appendix H

Title	Web document clustering based on Global-Best Harmony Search, K-means, Frequent Term Sets and Bayesian Information Criterion
Event	WCCI 2010 IEEE World Congress on Computational Intelligence IEEE Congress on Evolutionary Computation (IEEE CEC 2010)
Ranking	"A" by CORE (Computing Research and Education Association of Australasia) in 2010
Send date	January 30, 2010
Status	Published (July 18-23, 2010)
Note	Barcelona, Spain pp. 4637-4644 ISBN: 978-1-4244-6910-9

Web document clustering based on Global-Best Harmony Search, K-means, Frequent Term Sets and Bayesian Information Criterion

Carlos Cobos, Jennifer Andrade, William Constain, Martha Mendoza, Elizabeth León

Abstract— **This paper introduces a new description-centric algorithm for web document clustering based on the hybridization of the Global-Best Harmony Search with the K-means algorithm, Frequent Term Sets and Bayesian Information Criterion. The new algorithm defines the number of clusters automatically. The Global-Best Harmony Search provides a global strategy for a search in the solution space, based on the Harmony Search and the concept of swarm intelligence. The K-means algorithm is used to find the optimum value in a local search space. Bayesian Information Criterion is used as a fitness function, while FP-Growth is used to reduce the high dimensionality in the vocabulary. This resulting algorithm, called IGBHSK, was tested with data sets based on Reuters-21578 and DMOZ, obtaining promising results (better precision results than a Singular Value Decomposition algorithm). Also, it was also then evaluated by a group of users.**

I. Introduction

In recent years, web document clustering has become a very interesting research field. This is an alternative presentation of results based on what is known as the cluster hypothesis [1], according to which the clustering of documents may be beneficial to users of an information retrieval system, since it is likely that the results relevant to the user are close to each other in the document space, and therefore tend to fall into a relatively reduced number of clusters [2] allowing reductions in the search time.

To obtain good results in web document clustering the algorithms must meet the following specific requirements [3, 4]: Automatically define the number of clusters that are going to be created; generate relevant clusters for the user and assign these documents to appropriate clusters; define labels or names for the clusters that are easily understood for system users; handle overlapping clusters (this means that documents can belong to multiple clusters); reduce the high dimension that is presented in the management of document collections; handle the processing time, which means for example that the algorithm must be able to work with snippets and not only with the full text of the document; and handle the noise that is very common in the collection of documents.

Manuscript received January 30, 2010. This work was supported by a Research Grant from the University of Cauca under Project VRI-2560 and the National University of Colombia (Bogotá).

Carlos Cobos is with University of Cauca (phone: 57-2-8209800x2119; fax: 57-2-8209800x2102; e-mal: ccobos@unicauca.edu.co).

Jennifer Andrade, William Constain and Martha Mendoza are with University of Cauca (e-mal: {jandrade, wconstain, mmendoza}@unicauca.edu.co).

Elizabeth León is with National University of Colombia (e-mail: eleonguz@unal.edu.co).

Another important aspect when studying or proposing an algorithm to perform web document clustering is the document representation model. The most widely used models are [5]: ***Vector space model*** [1, 6], in which the documents are designed as bags of words, the document collection is represented by a matrix of D-terms by N-documents, each document is represented by a vector of normalized frequency term (tf_i) by the document inverse frequency for that term, in what is known as TF-IDF value, and the cosine distance is used for measuring the degree of similarity between two documents or between a document and the user's query. A process of stop word removal and stemming [1] should be done before re-presenting the document.

Several algorithms for web document clustering already exist, but results show there is still room for much to be done. These algorithms, by example, report precision and recall values between only 0.6 and 0.8, when the goal is 1.0 and their cluster labels are confused. This is the main motivation of the present work, in which a new algorithm that obtains better results for web document clustering is proposed.

The remainder of the paper is organized as follows. Section 2 presents some related work, the Global-Best Harmony Search algorithm and the K-means clustering algorithm. The proposed new algorithm is described in detail in Section 3. Section 4 shows the experimental results. Finally, some concluding remarks and suggestions for future work are presented.

II. Related Work

In general, clustering algorithms can be classified into [7], [8]: hierarchical, partitional, density-based, grid-based, and model-based algorithms, among others. The algorithms most commonly used for web document clustering have been the hierarchical and the partitional ones [6]. The hierarchical algorithms generate a dendogram or a tree of groups. This tree starts from a similarity measure, among which are: single link, complete link and average link. In relation to web document clustering, the hierarchical algorithm that brings the best results in accuracy is called UPGMA (Unweighted Pair-Group Method using Arithmetic averages) [7, 9].

In partitional clustering, the algorithms perform an initial division of the data in the clusters and then move the objects from one cluster to another based on the optimization of a predefined criterion or objective function [8]. The most representative algorithms using this technique are: K-means, K-medois, and Expectation Maximization. In 2000, a

Appendix I

Title	Web Document Clustering based on a New Niching Memetic Algorithm, Term-Document Matrix and Bayesian Information Criterion
Event	WCCI 2010 IEEE World Congress on Computational Intelligence IEEE Congress on Evolutionary Computation (IEEE CEC 2010)
Ranking	"A" by CORE (Computing Research and Education Association of Australasia) in 2010
Send date	January 30, 2010
Status	Published (July 18-23, 2010)
Note	Barcelona, Spain. pp. 4629-4636 ISBN: 978-1-4244-6910-9

Web Document Clustering based on a New Niching Memetic Algorithm, Term-Document Matrix and Bayesian Information Criterion

Carlos Cobos, Claudia Montealegre, María-Fernanda Mejía, Martha Mendoza, Elizabeth León

Abstract— **This paper introduces a new description-centric algorithm for web document clustering based on Memetic Algorithms with Niching Methods, Term-Document Matrix and Bayesian Information Criterion. The algorithm defines the number of clusters automatically. The Memetic Algorithm provides a combined global and local strategy for a search in the solution space and the Niching methods to promote diversity in the population and prevent the population from converging too quickly (based on restricted competition replacement and restrictive mating). The Memetic Algorithm uses the K-means algorithm to find the optimum value in a local search space. Bayesian Information Criterion is used as a fitness function, while FP-Growth is used to reduce the high dimensionality in the vocabulary. This resulting algorithm, called WDC-NMA, was tested with data sets based on Reuters-21578 and DMOZ, obtaining promising results (better precision results than a Singular Value Decomposition algorithm). Also, it was also then initially evaluated by a group of users.**

I. Introduction

In recent years, web document clustering has become a very interesting research field. This is an alternative presentation of results based on what is known as the cluster hypothesis [1], according to which the clustering of documents may be beneficial to users of an information retrieval system, since it is likely that the results relevant to the user are close to each other in the document space, and therefore tend to fall into a relatively reduced number of clusters [2] allowing reductions in the search time.

To obtain good results in web document clustering the algorithms must meet the following specific requirements [3, 4]: Automatically define the number of clusters that are going to be created; generate relevant clusters for the user and assign these documents to appropriate clusters; define labels or names for the clusters that are easily understood for system users; handle overlapping clusters (this means that documents can belong to multiple clusters); reduce the high dimension that is presented in the management of document collections; handle the processing time, which means for example that the algorithm must be able to work with snippets and not only with the full text of the document; and handle the noise that is very common in the collection of documents.

Another important aspect when studying or proposing an algorithm to perform web document clustering is the document representation model. The most widely used models are [5]: ***Vector space model*** [1, 6], in which the documents are designed as bags of words, the document collection is represented by a matrix of D-terms by N-documents, each document is represented by a vector of normalized frequency term by the document inverse frequency for that term, in what is known as TF-IDF value, and the cosine distance is used for measuring the degree of similarity between two documents or between a document and the user's query. Other models are ***Latent Semantic Indexing*** (LSI) [1, 7], ***Ontology-based model*** [8, 9], ***N-gram*** [4], ***Phrase-based model*** [4], and ***Frequent Word (Term) Sets model*** [9, 10]. In most of the previous representation models, a process of stop word removal (that reduces the dimensionality by more than 40%) and stemming (that reduces words to their canonical stem or root form) [1] should be done before re-presenting the document.

Several algorithms for web document clustering already exist, but results show there is still room for much to be done. These algorithms, by example, report precision and recall values between only 0.6 and 0.8, when the goal is 1.0 and their cluster labels are confused. This is the main motivation of the present work, in which a new algorithm that obtains better results for web document clustering is proposed.

The remainder of the paper is organized as follows. Section 2 presents some related work and a summary of the K-means clustering algorithm. The proposed new algorithm is described in detail in Section 3. Section 4 shows the experimental results. Finally, some concluding remarks and suggestions for future work are presented.

II. Related Work

In general, clustering algorithms can be classified into [11]: hierarchical, partitional, density-based, grid-based, and model-based algorithms, among others. The algorithms most commonly used for web document clustering have been the hierarchical and the partitional ones [6]. The hierarchical algorithms generate a dendogram or a tree of groups. This tree starts from a similarity measure, among which are:

Manuscript received January 30, 2010. This work was supported by a Research Grant from the University of Cauca under Project VRI-2560 and the National University of Colombia (Bogotá).

Carlos Cobos is with University of Cauca (phone: 57-2-8209800x2119; fax: 57-2-8209800x2102; e-mal: ccobos@unicauca.edu.co).

Claudia Montealegre, María-Fernanda Mejía and Martha Mendoza are with University of Cauca (e-mal: {cmontealegre, mmejia, mmendoza}@unicauca.edu.co).

Elizabeth León is with National University of Colombia (e-mail: eleonguz@unal.edu.co).

Appendix J

Title	A hyper-heuristic approach to design and tuning heuristic methods for web document clustering
Journal	IEEE Congress on Evolutionary Computation (IEEE CEC 2011)
Ranking	"A" by CORE (Computing Research and Education Association of Australasia) in 2010
Send date	January 28, 2011
Status	Published (June 5-8, 2011)
Note	New Orleans, USA. pp. 1350-1358. ISBN: 978-1-4244-7833-0.

A Hyper-Heuristic Approach To Design And Tuning Heuristic Methods For Web Document Clustering

Carlos Cobos
Computer Science Department
Universidad del Cauca
Popayán, Colombia
ccobos@unicauca.edu.co

Martha Mendoza
Computer Science Department
Universidad del Cauca
Popayán, Colombia
mmendoza@unicauca.edu.co

Elizabeth León
Systems and Industrial Department
Universidad Nacional de Colombia
Bogotá, Colombia
eleonguz@unal.edu.co

***Abstract*—This paper introduces a new description-centric algorithm for web document clustering called HHWDC. The HHWDC algorithm has been designed from a hyper-heuristic approach and allows defining the best algorithm for web document clustering. HHWDC uses as heuristic selection methodology two options, namely: random selection and roulette wheel selection based on performance of low-level heuristics (harmony search, an improved harmony search, a novel global harmony search, global-best harmony search, restrictive mating, roulette wheel selection, and particle swarm optimization). HHWDC uses the k-means algorithm for local solution improvement strategy, and based on the Bayesian Information Criteria is able to automatically define the number of clusters. HHWDC uses two acceptance/replace strategies, namely: Replace the worst and Restricted Competition Replacement. HHWDC was tested with data sets based on Reuters-21578 and DMOZ, obtaining promising results (better precision results than a Singular Value Decomposition algorithm).**

Keywords-web document clustering; hyper-heuristic; genetic algorithm; memetic algorithm; harmony search; particle swarm

I. INTRODUCTION

In recent years, web document clustering has become a very interesting research area among academic and scientific communities involved in information retrieval (IR) and web search [1]. Web document clustering systems seek to increase the coverage (amount) of documents presented for the user to review, while reducing the time spent reviewing them [2]. In IR, these web document clustering systems are called web clustering engines and the main exponents in the field are Carrot, Vivisimo, SnakeT, Dynamic SVD and STC [1]. Such systems usually consist of four main components: search results acquisition, preprocessing of input, ***cluster construction and labeling***, and visualization of resulting clusters [1].

To obtain good results in web document clustering the algorithms must meet the following specific requirements [1, 3]: Automatically define the number of clusters that are going to be created; generate relevant clusters for the user and assign these documents to appropriate clusters; define labels or names for the clusters that are easily understood for system users; handle overlapping clusters (this means that documents can belong to multiple clusters); handle short input data descriptions (document snippets); reduce the high-dimension that is presented in the management of document collections; handle the processing time (the algorithm must be able to work with snippets and not only with the full text of the document); and handle the noise that is very common in the collection of documents. Several algorithms for web document clustering already exist, but results show there is still room for much to be done. There are three types of algorithms [1]: data-centric, description-aware and description-centric. Each of these builds ***clusters*** of documents and assigns a label to the groups.

Data-centric algorithms are the algorithms traditionally used for data clustering (partitional, hierarchical, density-based, etc.) [1, 4-7]. They seek the best solution in data clustering, but are not so strong on the presentation of the labels or in the explanation of the groups obtained. They address the problem of web document clustering as merely another data clustering problem. In relation to web document clustering, the hierarchical algorithm that brings the best results in accuracy is called UPGMA (Unweighted Pair-Group Method using Arithmetic averages) [5, 6]. In partitional clustering, the most representative algorithms are: k-means, k-medois, and Expectation Maximization. On the other hand, Bisecting k-means [4, 8] is an algorithm that combines the strengths of the hierarchical and partitional methods reporting better results concerning the accuracy and the efficiency of the UPGMA and the k-means algorithms.

Description-aware algorithms give greater weight to one specific feature of the clustering process than to the rest. For example, they make as their priority the quality of the labeling of groups and as such achieve results that are more easily interpreted by the user. Their quality drops however in the cluster creation process. An example of this type of algorithm is Suffix Tree Clustering (STC) [3], which incrementally creates labels easily understood by users, based on common phrases that appear in the documents.

Description-centric algorithms [1, 8-13] are designed specifically for web document clustering, seeking a balance between the quality of clusters and the description (labeling) of them. An example of such algorithms is Lingo [9] (implemented by www.carrot2.org in 2001), which makes use of Singular Value Decomposition (SVD) to find the best relationships between terms, but groups the documents based

This work was supported by a Research Grant from the University of Cauca under Project VRI-2560 and the National University of Colombia (Bogotá).

Appendix K

Title	CMIN – A Case Tool Based on CRISP-DM to Support Data Mining Projects
Journal	Revista Ingeniería e Investigación - Universidad Nacional de Colombia
Class	ISI Journal (rated category A1 by PUBLINDEX-COLCIENCIAS)
Send date	July 21, 2009
Status	Published (December 2010)
Note	Volume 30. Issue 3. pp. 45-56. ISSN: 0120-5609.

En español

CMIN - herramienta case basada en CRISP-DM para el soporte de proyectos de minería de datos

Carlos Cobos[1], Jhon Zuñiga[2], Juan Guarin[3], Elizabeth León[4] y Martha Mendoza[5]

RESUMEN

En este artículo se presenta la CMIN, una herramienta CASE (*Computer Aided Software Engineering*) integrada (que soporta todas las fases de un proceso) basada en CRISP-DM 1.0 (*Cross – Industry Standard Process for Data Mining*) para soportar el desarrollo de proyectos de minería de datos. Primero se expone la funcionalidad general de CMIN, lo que incluye la gestión de procesos, plantillas y proyectos, y se destaca la capacidad de CMIN para realizar el seguimiento de los proyectos de una forma fácil e intuitiva y la manera como CMIN posibilita que el usuario incremente su conocimiento en el uso de CRISP-DM o de cualquier otro proceso que se defina en la herramienta a través de las ayudas e información que se ofrece en cada paso del proceso. Después, se detalla cómo CMIN permite enlazar en tiempo de ejecución (sin necesidad de volver a compilar la herramienta) nuevos algoritmos de minería de datos que apoyen la labor de modelado (basada en un flujo de trabajo o *workflow*) en un proyecto de minería de datos. Finalmente, se ofrecen los resultados de dos evaluaciones de la herramienta, las conclusiones y el trabajo futuro.

Palabras clave: minería de datos, CRISP-DM, herramientas CASE, *workflow*, reflexión.

Recibido: julio 21 de 2009
Aceptado: noviembre 15 de 2010

In English

CMIN – a CRISP-DM-based case tool for supporting data mining projects

Carlos Cobos[6], Jhon Zuñiga[7], Juan Guarin[8], Elizabeth León[9], Martha Mendoza[10]

ABSTRACT

This paper introduces CMIN, an integrated computer aided software engineering (CASE) tool based on cross-industry standard process for data mining (CRISP-DM) 1.0 designed to support carrying out data mining projects. It is "integrated" in the sense that it supports all phases of a process. A general overview of how CMIN works is presented first, including a treatment of processes, templates and project management. CMIN's capacity for easily and intuitively monitoring projects is highlighted, as is the manner in which CMIN allows a user to increase knowledge regarding using CRISP-DM or any other process defined in the CASE tool through the help and information presented in each step. Next, it is shown how CMIN can bind new data mining algorithms in runtime (without the need to recompile the tool) to support modelling tasks (based on a Workflow) and evaluate data mining projects. Finally, the results of two evaluations of the tool, some conclusions and suggestions for future work are presented.

Keywords: Data mining, CRISP-DM, CASE tools, workflow, reflection.

Received: july 21th 2009
Accepted: november 15th 2010

[1] Ingeniero de Sistemas. M.Sc., en Informática, Universidad Industrial de Santander, Colombia. Candidato a Ph.D., en Ingeniería de Sistemas y Computación, Universidad Nacional de Colombia, Bogotá, Colombia. Docente de Planta Tiempo Completo Categoría Titular, Universidad del Cauca, Colombia. Investigador del Grupo de I+D en Tecnologías de la Información (GTI), Universidad del Cauca, Colombia. ccobos@unicauca.edu.co.

[2] Ingeniero de Sistemas, Universidad del Cauca, Colombia. Programador, Informática y Gestión S.A., Colombia. Auxiliar de investigación del Grupo de I+D en Tecnologías de la Información, Universidad del Cauca, Colombia. jzunigaparedes@unicauca.edu.co.

[3] Ingeniero de Sistemas, Universidad del Cauca, Colombia. Programador, Solsoft S.A., Colombia. Auxiliar de investigación del Grupo de I+D en Tecnologías de la Información, Universidad del Cauca, Colombia. jguarin@unicauca.edu.co.

[4] Ingeniera de Sistemas. M.Sc., en Ingeniería de Sistemas, Universidad Nacional de Colombia, Colombia. M.Sc., in Electrical and Computer Engineering, University of Memphis, EEUU. Ph.D., in Computer Science and Computer Engineering, University of Louisville, EEUU. Docente de Planta Tiempo Completo Categoría Asistente, Universidad Nacional de Colombia sede Bogotá, Colombia. Investigadora del Laboratorio de Investigación en Sistemas Inteligentes (LISI), Universidad Nacional de Colombia sede Bogotá, Colombia. eleonguz@unal.edu.co.

[5] Ingeniera de Sistemas. M.Sc., en Informática, Universidad Industrial de Santander, Colombia. Estudiante de Ph.D., En Ingeniería de Sistemas y Computación, Universidad Nacional de Colombia sede Bogotá, Colombia. Docente de Planta Tiempo Completo Categoría Titular, Universidad del Cauca, Colombia. Investigadora del GTI, Universidad del Cauca, Colombia. mmendoza@unicauca.edu.co.

[6] Systems Engineer. M.Sc in Computer Science, Universidad Industrial de Santander, Colombia. Ph.D., candidate in Computer and Systems Engineering, Universidad Nacional de Colombia, Bogotá, Colombia. Plant Teachers Full Time Category Holder, Universidad del Cauca, Colombia. Researcher ID Group on Information Technology (GIT), Universidad del Cauca, Colombia. ccobos@unicauca.edu.co.

[7] Systems Engineer, Universidad del Cauca, Colombia Programmer, Informática y Gestión S.A., Colombia. Research Assistant Group ID in Information Technology, Universidad del Cauca, Colombia. jzunigaparedes@unicauca.edu.co.

[8] Systems Engineer, Universidad del Cauca, Colombia. Programmer, Solsoft S.A., Colombia. Research Assistant Group ID in Information Technology, Universidad del Cauca, Colombia. jguarin@unicauca.edu.co.

[9] Systems Engineer. M.Sc., in Systems Engineering, Universidad Nacional de Colombia, Colombia. M.Sc., in Electrical and Computer Engineering, University of Memphis, EEUU. Ph.D., in Computer Science and Computer Engineering, University of Louisville, EEUU. Plant Teachers Full Time Category Assistant, Universidad Nacional de Colombia, Bogotá, Colombia. Laboratory researcher in Intelligent Systems Research (LISI), Universidad Nacional de Colombia, Bogotá, Colombia. eleonguz@unal.edu.co.

[10] Systems Engineer. M.Sc., in Computer Science, Universidad Industrial de Santander, Colombia. Ph.D., student in Engineering Systems and Computing, Universidad Nacional de Colombia sede Bogotá, Colombia. Plant Teachers Full Time Category Holder, Universidad del Cauca, Colombia GTI Researcher, Universidad del Cauca, Colombia. mmendoza@unicauca.edu.co.

Appendix L

Title	A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes
Journal	Information Processing & Management
Class	ISI Journal (rated category A2 by PUBLINDEX-COLCIENCIAS)
Send date	March 20, 2012
Status	Published (May 2013)
Note	Volume 49. Issue 3. pp. 607-625. ISSN: 1657-4583.

Information Processing and Management 49 (2013) 607–625

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes

Carlos Cobos [a,b,*], Orlando Rodriguez [a,c], Jarvein Rivera [a], John Betancourt [a], Martha Mendoza [a,b], Elizabeth León [d], Enrique Herrera-Viedma [e]

[a] Information Technology Research Group (GTI), Universidad del Cauca, Sector Tulcán Office 422 FIET, Popayán, Colombia
[b] Computer Science Department, Electronic and Telecommunications Engineering Faculty, Universidad del Cauca, Colombia
[c] Mathematics Department, Faculty of Exact and Natural Sciences, Universidad del Cauca, Colombia
[d] Systems and Industrial Engineering Department, Engineering Faculty, Universidad Nacional de Colombia, Colombia
[e] Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain

ARTICLE INFO

Article history:
Received 20 March 2012
Received in revised form 29 November 2012
Accepted 5 December 2012
Available online 20 January 2013

Keywords:
Recommender systems
Pedagogical patterns
Singular value decomposition
Cosine similarity
Collaborative filtering
Resnick prediction formula

ABSTRACT

To carry out effective teaching/learning processes, lecturers in a variety of educational institutions frequently need support. They therefore resort to advice from more experienced lecturers, to formal training processes such as specializations, master or doctoral degrees, or to self-training. High costs in time and money are invariably involved in the processes of formal training, while self-training and advice each bring their own specific risks (e.g. of following new trends that are not fully evaluated or the risk of applying techniques that are inappropriate in specific contexts).This paper presents a system that allows lecturers to define their best teaching strategies for use in the context of a specific class. The context is defined by: the specific characteristics of the subject being treated, the specific objectives that are expected to be achieved in the classroom session, the profile of the students on the course, the dominant characteristics of the teacher, and the classroom environment for each session, among others. The system presented is the Recommendation System of Pedagogical Patterns (RSPP). To construct the RSPP, an ontology representing the pedagogical patterns and their interaction with the fundamentals of the educational process was defined. A web information system was also defined to record information on courses, students, lecturers, etc.; an option based on a unified hybrid model (for content and collaborative filtering) of recommendations for pedagogical patterns was further added to the system. RSPP features a minable view, a tabular structure that summarizes and organizes the information registered in the rest of the system as well as facilitating the task of recommendation. The data recorded in the minable view is taken to a latent space, where noise is reduced and the essence of the information contained in the structure is distilled. This process makes use of Singular Value Decomposition (SVD), commonly used by information retrieval and recommendation systems. Satisfactory results both in the accuracy of the recommendations and in the use of the general application open the door for further research and expand the role of recommender systems in educational teacher support processes.

© 2012 Elsevier Ltd. All rights reserved.

* Corresponding author at: Information Technology Research Group (GTI), Universidad del Cauca, Sector Tulcán Office 422 FIET, Popayán, Colombia. Tel.: +57 2 8366524; fax: +57 2 8209810.
E-mail address: coboscarlos@gmail.com (C. Cobos).

0306-4573/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ipm.2012.12.002

Appendix M

Title	Clustering of Web Search Results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion
Event	2013 IFSA-NAFIPS Joint Congress (International Fuzzy Systems Association)
Ranking	"B" by CORE (Computing Research and Education Association of Australasia) in 2010
Send date	January 28, 2013
Status	Published (June 24-28, 2013)
Note	Edmonton, Canada. pp. ISBN:

Clustering of Web Search Results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion

Carlos Cobos
Computer Science Department
Universidad del Cauca
Popayán, Colombia
ccobos@unicauca.edu.co

Martha Mendoza
Computer Science Department
Universidad del Cauca
Popayán, Colombia
mmendoza@unicauca.edu.co

Elizabeth León
Systems and Industrial Department
Universidad Nacional de Colombia
Bogotá, Colombia
eleonguz@unal.edu.co

Milos Manic
Department of Computer Science
University of Idaho at Idaho Falls
Idaho Falls, U.S.A.
misko@uidaho.edu

Enrique Herrera-Viedma
Department of Computer Science
and Artificial Intelligence
University of Granada
Granada, Spain
viedma@decsai.ugr.es

***Abstract*—The clustering of web search has become a very interesting research area among academic and scientific communities involved in information retrieval. Clustering of web search result systems, also called Web Clustering Engines, seek to increase the coverage of documents presented for the user to review, while reducing the time spent reviewing them. Several algorithms for web document clustering already exist, but results show there is room for more to be done. This paper introduces a new description-centric algorithm for clustering of web results called IFCWR. IFCWR initially selects a maximum estimated number of clusters using Forgy's strategy, then it iteratively merges clusters until results cannot be improved. Every merge operation implies the execution of Fuzzy C-Means for clustering results of web search and the calculus of Bayesian Information Criterion for automatically evaluating the best solution and number of clusters. IFCWR was compared against other established web document clustering algorithms, among them: Suffix Tree Clustering and Lingo. Comparison was executed on AMBIENT and MORESQUE datasets, using precision, recall, f-measure, SSL_k and other metrics. Results show a considerable improvement in clustering quality and performance.**

Keywords-web document clustering; fuzzy c-means; bayesian information criterion

I. INTRODUCTION

In recent years, clustering of web search results -or web document clustering- has become a very interesting research area among academic and scientific communities involved in information retrieval (IR) and web search [2]. Web document clustering systems seek to increase the coverage (amount) of documents presented for the user to review, while reducing the time spent reviewing them [3]. In IR, these web document clustering systems are called web clustering engines and the main exponents in the field are Carrot^2 (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, originally named as Vivisimo and then as Clusty), iBoogie (www.iboogie.com), and KeySRC (http://keysrc.fub.it) [4]. Such systems usually consist of four main components: search results acquisition, preprocessing of input, ***cluster construction and labeling***, and visualization of resulting clusters [2].

To obtain good results in web document clustering the algorithms must meet the following specific requirements [2, 5]: Automatically define the number of clusters to be created; generate relevant clusters for the user and assign the documents to appropriate clusters; define labels or names for the clusters that are easily understood by users; handle overlapping clusters (this means that documents can belong to multiple clusters); handle short input data descriptions (document snippets); reduce the high-dimension that is presented in the management of document collections; handle the processing time (the algorithm must be able to work with snippets and not only with the full text of the document); and handle the noise that is very common in the collection of documents. Several algorithms for web document clustering already exist, but results show there is still much to be done. There are three types of algorithms [2]: data-centric, description-aware and description-centric. Each of these builds ***clusters*** of documents and most of them assign a label to each group.

All of these algorithms report quality of clustering values, represented by low values of F-measure, i.e. between only 0.5 and 0.58 for AMBIENT and MORESQUE datasets, when the goal is 1.0 and their cluster labels can be improved. This is the main motivation of the present work, in which a new algorithm

This work was supported by the University of Cauca, the National University of Colombia (Bogotá), and the Spanish Ministry of Public Works and Transport.

Appendix N

Title	Extractive single-document summarization based on genetic operators and guided local search
Journal	Expert Systems with Applications
Class	ISI Journal (rated category A1 by PUBLINDEX-COLCIENCIAS)
Send date	October 18, 2013
Status	In Press, Accepted Manuscript (December 28, 2013)
Note	Volume -. Issue -. pp. -. ISSN: 0957-4174.

Expert Systems with Applications xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Extractive single-document summarization based on genetic operators and guided local search

Martha Mendoza [a,b,*], Susana Bonilla [a], Clara Noguera [a], Carlos Cobos [a,b], Elizabeth León [c]

[a] Information Technology Research Group (GTI), Universidad del Cauca, Sector Tulcán Office 422, Popayán, Colombia
[b] Computer Science Department, Electronic and Telecommunications Engineering Faculty, Universidad del Cauca, Colombia
[c] Data Mining Research Group (MIDAS), Engineering Faculty, Universidad Nacional de Colombia, Bogotá, Colombia

ARTICLE INFO

Keywords:
Extractive summarization
Single document
Memetic algorithm
Guided local search

ABSTRACT

Due to the exponential growth of textual information available on the Web, end users need to be able to access information in summary form – and without losing the most important information in the document when generating the summaries. Automatic generation of extractive summaries from a single document has traditionally been given the task of extracting the most relevant sentences from the original document. The methods employed generally allocate a score to each sentence in the document, taking into account certain features. The most relevant sentences are then selected, according to the score obtained for each sentence. These features include the position of the sentence in the document, its similarity to the title, the sentence length, and the frequency of the terms in the sentence. However, it has still not been possible to achieve a quality of summary that matches that performed by humans and therefore methods continue to be brought forward that aim to improve on the results. This paper addresses the generation of extractive summaries from a single document as a binary optimization problem where the quality (fitness) of the solutions is based on the weighting of individual statistical features of each sentence – such as position, sentence length and the relationship of the summary to the title, combined with group features of similarity between candidate sentences in the summary and the original document, and among the candidate sentences of the summary. This paper proposes a method of extractive single-document summarization based on genetic operators and guided local search, called MA-SingleDocSum. A memetic algorithm is used to integrate the own-population-based search of evolutionary algorithms with a guided local search strategy. The proposed method was compared with the state of the art methods UnifiedRank, DE, FEOM, NetSum, CRF, QCS, SVM, and Manifold Ranking, using ROUGE measures on the datasets DUC2001 and DUC2002. The results showed that MA-SingleDocSum outperforms the state of the art methods.

© 2014 Published by Elsevier Ltd.

1. Introduction

Due to the exponential growth of textual information available on the Web and the access to information by the users through new portable devices, it is necessary that the end user can access the information in summary form and without losing the most important aspects presented therein. Some of the application areas of the generation of extractive summaries from a single document are the summaries of web pages presented on the search engines (Porselvi & Gunasundari, 2013); the assignation of the labels to groups generated in the web document clustering (Carpineto, Osinski, Romano, & Weiss, 2009); and in the E-learning context is used to select the most important information from a text (Kumaresh & Ramakrishnan, 2012). The automatic generation of text summaries has been tasked with addressing this problem for many years, seeking to obtain short texts that present the most relevant ideas in a document (Lloret & Palomar, 2012; Nenkova & McKeown, 2012; Spärck Jones, 2007). To achieve this, several methods have been developed that summarize one or multiple documents, with the aim that the user select and review in the shortest time those documents that really meet their information needs.

Different taxonomies for the summaries exist (Ježek & Steinberger, 2008; Lloret & Palomar, 2012; Nenkova & McKeown, 2012), based on the way the summary is generated, the target audience of the summary, the number of documents to be summarized, and so on.

According to the way in which it is generated, the summary may represent either an extraction or an abstraction (Ježek & Steinberger, 2008; Lloret & Palomar, 2012; Nenkova & McKeown, 2012).

* Corresponding author at: Information Technology Research Group (GTI), Universidad del Cauca, Sector Tulcán Office 422, Popayán, Colombia. Tel.: +57 28366524; fax: +57 28209810.
E-mail addresses: mmendoza@unicauca.edu.co, mendoza.martha.eliana@gmail.com (M. Mendoza).

0957-4174/$ - see front matter © 2014 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.eswa.2013.12.042

Please cite this article in press as: Mendoza, M., et al. Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications* (2014), http://dx.doi.org/10.1016/j.eswa.2013.12.042

UNCORRECTED PROOF