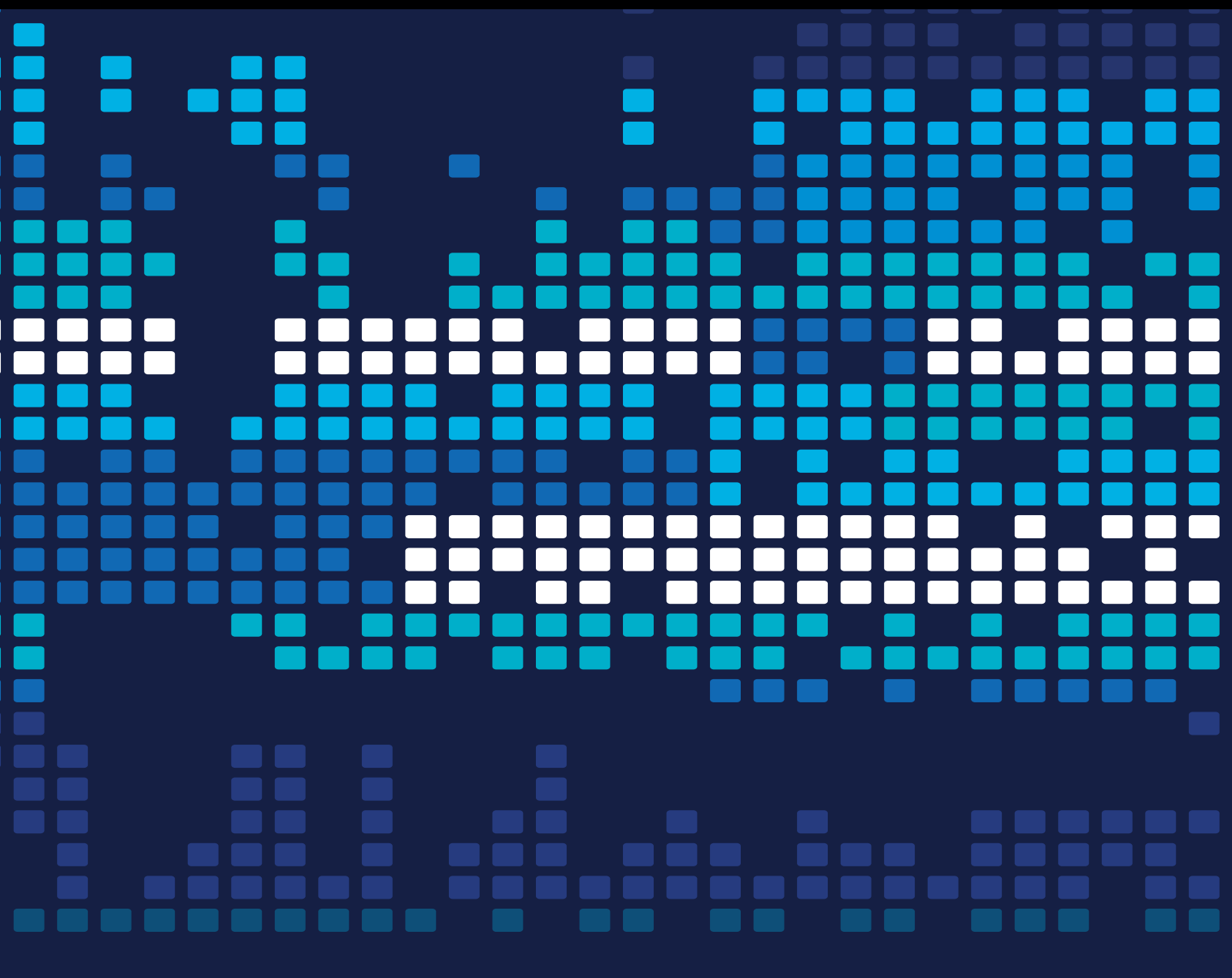
Scientific Programming

Scientific Programming Techniques and Algorithms for Data-Intensive Engineering Environments

Lead Guest Editor: Giner Alor-Hernandez
Guest Editors: Jezreel Mejia-Miranda and José María Álvarez-Rodríguez



Scientific Programming Techniques and Algorithms for Data-Intensive Engineering Environments

Scientific Programming

Scientific Programming Techniques and Algorithms for Data-Intensive Engineering Environments

Lead Guest Editor: Giner Alor-Hernández
Guest Editors: Jezreel Mejia-Miranda
and José María Álvarez-Rodríguez

Editorial Board

Contents

Hindawi
Scientific Programming
Volume 2018, Article ID 1351239, 3 pages
https://doi.org/10.1155/2018/1351239

Editorial

Scientific Programming Techniques and Algorithms for Data-Intensive Engineering Environments

Giner Alor-Hernández,[1] **Jezreel Mejía-Miranda**,[2] **and José María Álvarez-Rodríguez**[3]

[1]*Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Orizaba, VER, Mexico*
[2]*Research Center in Mathematics (CIMAT, A. C.), Unit Zacatecas, Zacatecas, ZAC, Mexico*
[3]*Department of Engineering and Computer Science, Universidad Carlos III de Madrid, Madrid, Spain*

Correspondence should be addressed to Giner Alor-Hernández; galor@itorizaba.edu.mx

Received 18 February 2018; Accepted 19 February 2018; Published 5 November 2018

1. Introduction

In recent years, the development, advancement, and use of Information and Communications Technology (ICT) have had a major impact on the operation, structure, and strategy of organizations around the world. Today, it is unthinkable to conceive an organization without the use of ICT, because it allows for the reduction of communication costs and operation while increasing flexibility, interactivity, performance, and productivity. As a result, digital technology fueled by ICT and ICT is currently embedded in any task, activity, and process that is done in any organization or even our daily life activities. This new digital age also implies that science, engineering, and business environments need to reshape their strategies and underlying technology to become a key player of the Industrial Revolution 4.0.

Engineering methods such as requirements engineering, systems modeling, complex network analysis, or simulation are currently applied to support the development of critical systems and decision-making processes in operational environments. As an example, cyberphysical systems featured by mechanical, electrical, and software components are a major challenge for the industry in which new and integrated science and engineering techniques are required to develop and operate these systems in a collaborative data-intensive environment.

Both the development processes and the operational environments of complex systems need the application of scientific and engineering methods to fulfill the management of new multidisciplinary, data-intensive, and software-centric environments. Programming paradigms such as functional, symbolic, logic, linear, or reactive programming in conjunction with development platforms are considered a cornerstone for the proper development of intelligent and federated programming platforms to support continuous and collaborative engineering.

More specifically, the availability of huge amounts of data that are continuously generated by persons, tools, sensors, and any other smart connected device requires new architectures to address the challenge of solving complex problems such as pattern identification, process optimization, discovery of interactions, knowledge inference, execution of large simulations, or machine cooperation. This situation implies the rethinking and application of innovative scientific programming techniques for numerical, scientific, and engineering computation on top of well-defined hardware and software architectures to support the proper development and operation of complex systems.

In this context, the evolution and extension of engineering methods through scientific programming techniques in data-intensive environments are expected to take advantage of innovative algorithms implemented using different programming paradigms and execution platforms. The conjunction of scientific programming techniques and engineering techniques will support and enhance existing development and production environments to provide high quality, economical, reliable, and efficient data-centric software products and services. This advance in the field of scientific

programming methods will become a key enabler for the next wave of software systems and engineering.

Therefore, the main objective of this special issue was to collect and consolidate innovative and high quality research contributions regarding scientific programing techniques and algorithms applied to the enhancement and improvement of engineering methods to develop real and sustainable data-intensive science and engineering environments.

2. Papers

This special issue aims to provide insights into the recent advances in the aforementioned topics by soliciting original scientific contributions in the form of theoretical foundations, models, experimental research, surveys, and case studies for scientific programing techniques and algorithms in data-intensive environments. This special issue just contains one type of contribution: regular research papers. These works have been edited according to the norms and guidelines of the journal. Several call for papers were distributed among the main mailing lists of the field for researchers to submit their works to this issue. We received a total of 20 submissions which were subject to a rigorous review process to ensure their clarity, authenticity, and relevance to this special issue. At least three reviewers were assigned to every work to proceed with the peer review process. Seven papers were finally accepted for their publication after corrections requested by reviewers and editors were addressed. The seven regular research papers introduce new and interesting results in the form of theoretical and experimental research and case studies under new perspectives on scientific programming techniques and algorithms for data-intensive engineering environments.

One of the special issue's research papers is entitled "Design and Solution of a Surrogate Model for Portfolio Optimization Based on Project Ranking," where E. Fernandez et al. propose a knowledge-based decision support system for the Portfolio Selection Problem on a Set of Ordered Projects. The results show that the reduction of the dimensionality supports the decision-maker in choosing the best portfolio.

In another contribution, entitled "Sentiment Analysis in Spanish for Improvement of Products and Services: A Deep Learning Approach," M. A. Paredes-Valverde et al. propose a deep-learning-based approach that allows companies and organizations to detect opportunities for improving the quality of their products or services through sentiment analysis. The approach is based on Convolutional Neural Network (CNN) and word2vec. To determine the effectiveness of the approach for classifying tweets, some experiments were conducted with different sizes of a Twitter corpus composed of 100000 tweets. Results showed a precision of 88.7%, a recall of 88.7%, and an F-measure of 88.7% considering the complete dataset.

In a further paper, entitled *Scalable* "Parallel Distributed Coprocessor System for Graph Searching Problems with Massive Data," Y. Sun at al. propose a scalable and novel field programmable gate array-based heterogeneous multicore system for scientific programming. Authors designed considerable parallelism and relatively low clock frequencies to achieve high performance and customized memory architecture to deal with irregular memory access pattern.

In an additional contribution, entitled "Analysis of Medical Opinions about the Nonrealization of Autopsies in a Mexican Hospital Using Association Rules and Bayesian Networks," E. R. Delgado identifies the factors influencing the reduction of autopsies in a hospital in Veracruz. The study is based on the application of data mining techniques such as association rules and Bayesian networks in datasets created from the opinions of physicians. For the exploration and extraction of the knowledge, some algorithms like Apriori, FPGrowth, PredictiveApriori, Tertius, J48, Naive Bayes, MultilayerPerceptron, and BayesNet were analyzed. To generate mining models and present the new knowledge in natural language, a web-based application was developed. The results have been validated by specialists in the field of pathology.

The fifth paper, entitled "Applying Softcomputing for Copper Recovery in Leaching Process," C. Leiva et al. present a predictive modelling contrasting; a linear model, a quadratic model, a cubic model, and a model based on the use of an artificial neural network (ANN) for copper recovery in leaching process. The ANN was constructed with 9 input variables, 6 hidden layers, and a neuron in the output layer corresponding to copper leaching prediction. The validation of the models was performed with real information and these results were used by a mining company in northern Chile to improve copper mining processes.

In a further contribution, entitled "Semantic Annotation of Unstructured Documents Using Concepts Similarity," F. Pech et al. propose a semantic annotation strategy for unstructured documents as part of a semantic search engine. Ontologies are used to determine the context of the entities specified in the query. The strategy to extracting the context is focused on concepts similarity. Each relevant term of the document is associated with an instance in the ontology. The similarity between each of the explicit relationships is measured through the combination of two types of associations: the association between each pair of concepts and the calculation of the weight of the relationships.

Finally, a paper entitled "A Heterogeneous System Based on Latent Semantic Analysis Using GPU and Multi-CPU" is by G. A. Leon-Paredes et al. They introduce a heterogeneous Latent Semantic Analysis (hLSA) system, which has been developed using general-purpose computing on graphics processing units (GPGPU), which can solve large problems faster through the thousands of concurrent threads on the multiple-core multiprocessors of GPUs and multi-CPU architectures which offer a shared memory programming model in a multithreaded environment. The results of the experiments show that the acceleration reached by an hLSA system for large matrices with one hundred and fifty thousand million values is around eight times faster than the standard LSA version with an accuracy of 88% and a recall of 100%. The performance gain is achieved by using heterogeneous system architectures for matrix computation and text processing.

3. Conclusions

As can be seen, all accepted papers are aligned with the scope of the special issue, and all of them provide quite interesting research techniques, models, and studies directly applied to the area of scientific programming techniques and algorithms for data-intensive engineering Environments.

Acknowledgments

The preparation of this special collection has been partially supported by the Tecnologico Nacional de Mexico (TECNM), National Council of Science and Technology (CONACYT), and the Public Education Secretary (SEP) through PRODEP. It has also been supported by the Research Agreement between the RTVE (the Spanish Radio and Television Corporation) and the UC3M to boost research in the fields of big data, linked data, complex network analysis, and natural language. Last but not least, we would also like to express our gratitude to the reviewers who kindly accepted to contribute in the evaluation of papers at all stages of the editing process. We equally and especially wish to thank the editorial board for the opportunity to edit this special issue and for providing valuable comments to improve the selection of research works.

Giner Alor-Hernández
Jezreel Mejía-Miranda
José María Álvarez-Rodríguez

Hindawi
Scientific Programming
Volume 2018, Article ID 8467413, 21 pages
https://doi.org/10.1155/2018/8467413

Review Article

Survey of Scientific Programming Techniques for the Management of Data-Intensive Engineering Environments

Jose María Álvarez-Rodríguez,[1] **Giner Alor-Hernández**,[2] **and Jezreel Mejía-Miranda**[3]

[1]*Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Madrid, Spain*
[2]*Division of Research and Postgraduate Studies, Tecnológico Nacional de México/I.T., Orizaba, Mexico*
[3]*Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico*

Correspondence should be addressed to Jose María Álvarez-Rodríguez; josemaria.alvarez@uc3m.es

Received 22 February 2018; Accepted 3 October 2018; Published 30 October 2018

Academic Editor: Giuseppe Scanniello

The present paper introduces and reviews existing technology and research works in the field of scientific programming methods and techniques in data-intensive engineering environments. More specifically, this survey aims to collect those relevant approaches that have faced the challenge of delivering more advanced and intelligent methods taking advantage of the existing large datasets. Although existing tools and techniques have demonstrated their ability to manage complex engineering processes for the development and operation of safety-critical systems, there is an emerging need to know how existing computational science methods will behave to manage large amounts of data. That is why, authors review both existing open issues in the context of engineering with special focus on scientific programming techniques and hybrid approaches. 1193 journal papers have been found as the representative in these areas screening 935 to finally make a full review of 122. Afterwards, a comprehensive mapping between techniques and engineering and nonengineering domains has been conducted to classify and perform a meta-analysis of the current state of the art. As the main result of this work, a set of 10 challenges for future data-intensive engineering environments have been outlined.

1. Introduction

Digital technology fueled by software is currently embedded in any task, activity, and process that are done in any organization or even in our daily life activities. The Digital Age has come to stay meaning that any industry or business need to reshape strategies (operational, social, and economic) to become part of what is known as "the 4th Industrial Revolution" or "Industry 4.0" [1]. The first step towards the smart automation and data exchange in industrial environments relies on the application of new techniques to create new technology-driven business opportunities. However, technology is not completely the focus but people: consumers, workers, and partners. The change in the corporate culture through technology is considered a cornerstone to empower people and to drive the change and disruption in a domain turning a traditional organization into a leading digital entity. In this context, it is possible to find several consultancy reports, such as the "2016 Accenture Technology vision report" that outlines five trends to shape this new environment. Organizations shall focus on the creation of data-driven solutions powered by artificial intelligence ("Intelligent Automation") equipping people ("Liquid Workforce") with the required skills to build new execution platforms ("Platform Economy"), boosting disruption ("Predictable Disruption"), and trustworthy digital ecosystems ("Digital Trust").

In the frame of engineering processes and methods, the initiatives of "Industrial Internet" or "Industrial Data Spaces" among others are trying to define the new methodologies and good practices to bring the digital age to the industry and engineering. In this light, a set of technologies such as Security, Big Data, Mobility, Natural Language Processing, Deep Learning, Internet of X (things, people, tools, everything, etc.), User Interfaces, 3D Printing, Virtual Reality, or Cloud Computing are aimed at changing both the

development and production environments of complex products and services. The notion of the cyberphysical system [2] (CPS) is currently gaining momentum to name those engineering systems-combining technologies coming from different disciplines such as mechanical, electrical, and software engineering. CPSs represent the next generation of interconnected complex critical systems in sectors such as automotive, aerospace, railway, or medical devices in which the combination of different engineering areas are governed by software. The increasing complexity in the development of such systems also implies unprecedented levels of interaction between models and formalisms from the inception of the system to the production, distribution, support, and decommissioning stages.

In order to tackle the needs of new engineering environments, collaborative engineering [3], "*concept of optimizing engineering processes with objectives for better product quality, shorter lead time, more competitive cost, and higher customer satisfaction*" [4], represents a shifting paradigm from islands of domain knowledge to an interconnected knowledge graph of services [5], engineering processes, and people. Methods such as requirements engineering, modelling, and simulation or cosimulation to support processes such as analysis, design, traceability, verification, validation, or certification demand now integration and interoperability in the development toolchain to automatically exchange data, information, and knowledge. That is why, last times have seen the emergence of model-based systems engineering (MBSE) [6] as a complete methodology to address the challenge of unifying the techniques, methods, and tools to support the whole specification process of a system. In the context of the well-known Vee life cycle model, it means that there is "formalized application of modelling" (http://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf?sfvrsn=0) to support the left-hand side of this system life cycle implying that any process, task, or activity will generate different system artifacts but all of them represented as a model. In this manner, the current practice in the Systems Engineering discipline is improved through the creation of a continuous and collaborative engineering environment easing the interaction and communication between both tools and people.

However, there is much more at stake than the connection of tools and people, the increasing complexity of products and services also requires the improvement and extension of existing techniques. In this context, scientific programming techniques such as computational modelling and simulation, numerical and nonnumerical algorithms, or linear programming to name a few have been largely used and studied [7, 8] and applied to solve complex problems in disciplines such as natural sciences, engineering, or social sciences. Furthermore, the exponential increase of data has created a new complete environment in which computational scientists and practitioners must concentrate [8] not only in the formulation of hypotheses, creation of models, and execution of experiments but also in the complexity of techniques and combination of tools [9] to deal with large amounts of data.

Considering the current digital transformation in the industry, the need of producing complex CPS through the collaboration between different engineering domains and the increasing amounts of data, this survey looks for summarizing the last existing studies applying scientific programming techniques in the context of data-intensive engineering environments.

2. Background

The development of a complex system such as CPS requires the involvement of hundreds of engineers working with different tools and generating thousands of system artifacts such as requirements specifications, test cases, or models (logical, physical, simulation, etc.). These large amounts of data must then be integrated together to give support to those engineering processes that require a holistic view of a system such as traceability management or verification. In this frame, data management techniques are becoming a cornerstone to the proper exploitation of the underlying data and for the successful development of the system.

Last years have also seen a large body of work in the field of Big Data covering from the creation of tools and architectural frameworks to its application to different domains [10–12] such as social network analysis [13], bioinformatics [14], earth science [15], e-government [16], e-health [17, 18], or e-tourism [19]. More specifically, in the context of engineering and industry, some works have been reported [20] by large companies such as Teradata in particular domains such as maintenance of aircraft engines. All the work around tries to provide a response for data-centric environments in which it is necessary to deal with the well-known V's of a data ecosystem: variety, volume, velocity, and veracity. Big Data technology has been successfully applied to deal with the large amounts of data that are usually created during simulation processes of complex critical systems such as CPS. Moreover, last times have seen the emergence of a new notion "digital twin." A digital twin is defined as a "*digital replica of physical assets, processes and systems*" that looks for replying, in a digital environment, the same working conditions than a physical environment. This approach is being used to improve the verification and validation stages of CPS such as smart cars. For instance, autonomous cars need to pass certification processes in which manufacturers must demonstrate the proper behavior of the car under certain circumstances and consider the new artificial intelligence capabilities. Since it is not completely possible to create a complete digital environment for these large simulations, there is currently a trend to design a kind of validation loop in which real data are used to feed synthetic data. From a data management perspective, this approach implies the need of providing a data life cycle management process to represent, store, access, and enrich data. After several simulations, data gathered from car sensors under real conditions can be over hundreds of petabytes. Similar approaches are used to validate aircraft engines or wind turbines.

Thus, data storage systems and processing frameworks have been developed [21] demonstrating the viability of a technology that can now be considered mature. NoSQL data storage systems [22] based on different representation mechanisms such as key-value stores, documents, distributed

files, wide column stores, graphs, object databases, and tuple stores such as Apache Hive, MongoDB, ArangoDB, Apache Cassandra, Neo4J, Redis, Virtuoso, or CouchDB can be found to handle large volumes of evolving data. In the case of processing frameworks, technology offering capabilities to process data under different models (stream, batch, or event) such as the Apache projects: Hadoop (and its ecosystem of technology), Storm, Samza, Spark, or Flink can also be found. Furthermore, most of them usually offer us not just a distributed computational model to tackle the well-known CAP theorem [23] but a set of libraries [24] for implementing applications on top of existing machine learning techniques [25] (e.g., Mlib in Spark or FlinkML in Flink) or large-scale graph processing [26] (e.g., GraphX in Spark).

This plethora of tools and technology has also generated the development of technology and tools for the management of Big Data infrastructures and resources such as Apache Messos or YARN and the emergence of companies and commercial tools such as Cloudera, MapR, Teradata, or HortonWorks (apart from the toolsets offered by the big software players such as Amazon, Google, IBM, Oracle, or SAP). Moreover, Big Data technology has found in the cloud computing area a good travelling companion [27, 28] since the cloud provides a powerful and massive scale technology for complex computation decreasing the cost of hardware and software maintenance. In this sense, the current challenges rely on the automatic deployment of Big Data infrastructures under different topologies and technologies keeping nonfunctional aspects such as scalability, availability, security, regulatory and legal issues, or data quality as main drivers for research and innovation.

Once Big Data and cloud computing technology have been briefly summarized, it is important to highlight what is being considered one of the next big things in this area: the combination of high-performance computing (HPC) and Big Data technology [29]. The growing interest in this area [30] looks for joining the efforts to shift the paradigm of large and complex computations to a kind of Big Data analysis problem. In this way, new computational and programming models taking advantage of new data storage systems and extensions in programming languages such as R, Fortran, or Python are becoming critical to address the objective of performing time-consuming tasks (computational complexity) over large amounts of data.

That is why, the main aim of this review is to examine the existing works in the field of Big Data and scientific programming in the engineering discipline. A better understanding of this new environment may potentially allow us to address existing challenges in the new computational models to be designed.

3. Review Protocol

As it has been previously introduced, a good number of systematic reviews can be found in the field of Big Data [18, 27] and specific domains such as the health sector [17, 18, 31] the same manner, scientific programming techniques have been widely studied and reviewed [7, 8]. However, the application of Big Data technology and scientific programming techniques to the engineering domain has not been fully reviewed, and it is not easy to draw a comprehensive picture of the current state of the art apart from works in specific domains such as climate [32] or astronomy [20]. That is why, this work aims at providing an objective, contemporary, and high-quality review of the existing works following the formal procedures for conducting systematic reviews [31, 33]. The first step relies then on the definition of the systematic review protocol (SRL) as follows:

(1) *Research Question*. The main objective of this work is to provide a response to the following research question:

Which are the last techniques, methods, algorithms, architectures, and infrastructures in scientific programming/computing for the management, exploitation, inference, and reasoning of data in engineering environments?

To formulate a researchable and significant question, the PICO model is used to specify the different elements of the query to be formulated; see Table 1 outlining the description of each PICO element.

(2) *Search String*. According to the PICO model and the application of natural language processing techniques to add new terms to the query, the next search string has been formulated; see Table 2.

(3) *Bibliographic Database Selection*. In this case, common and large bibliographic databases in the field of computer science have been selected as in other previous survey works [34]:

(a) ACM Digital Library (ACM): this library comprises the most comprehensive collection of full-text articles in the fields of computing and information technology. The ACM Digital Library consists of 54 journals, 8 niche technology magazines, 37 special interest group newsletters, +275 conference proceedings volumes (per year), and over 2,237,215 records in the Guide to Computing Literature Index.

(b) IEEE Xplore Digital (IEEE): the IEEE Xplore® digital library provides "*access to the cutting-edge journals, conference proceedings, standards, and online educational courses that define technology today.*" "*The content in IEEE Xplore comprises 195 + journals, 800+ conferences, 6,200+ technical standards, approximately 2,400 books, and 425+ educational courses. Approximately 20,000 new documents are added to IEEE Xplore each month.*"

(c) ScienceDirect (Elsevier): ScienceDirect "*is a leading full-text scientific database offering science, medical, and technical (STM) journal articles and book chapters from more than 3,800 peer-reviewed journals and over 37,000 books. There are currently more than 11 million articles/chapters, a content base that is growing at a rate of almost 0.5 million additions per year.*"

Table 1: The PICO model applied to formulate the research question.

PICO element	Description
Population	Engineering environments and methods
Intervention	Scientific programming/computing techniques, algorithms, methods, architectures, and infrastructures to deal with data-intensive engineering
Comparison	Performance measures, benchmarks, and datasets
Outcomes	New and collaborative techniques, algorithms, architectures, infrastructures, domains, and case studies

Table 2: Search string including operators.

(scientific AND (programming OR computing) AND (model OR technique OR method OR algorithms OR architecture OR infrastructure) AND (data OR "big data") AND engineering)

(d) SpringerLink (Springer): ScienceDirect is a leading full-text scientific database offering science, medical, and technical (STM) journal articles and book chapters. It comprises 12,258,260 resources consisting of a 52% of articles, 34% of chapters, 8% of conference papers, and 4% reference work entries.

Moreover, some aggregation services such as Google Scholar and DBLP have also been checked with the aim of getting those results that can be found on other sources such as webpages or nonscientific articles. However, these aggregation services can also include works that have been published under different formats such as technical reports, conference, and journal papers, so it is important to carefully select the complete and most up-to-date version of the works to avoid duplicates. That is why, during the selection procedure, works were selected removing potential duplicates to provide a whole and unique picture of the research landscape.

(4) *Inclusion Criteria*. This review collects studies that have been published in English as a journal paper. English has been selected as the target language since relevant journals in these topics mainly include works to get the attraction of researchers and practitioners at a worldwide scale. Although many conferences and workshops in the field of data science and engineering have emerged during the last times generating a quite good number of papers, this review looks for works with strong foundations as those available as journal papers. In the case of the timeframe, a range of 5 years (2012–2017) has been defined to include relevant and up-to-date papers. Furthermore, a special attention has been paid to remove duplicate studies or extensions that can appear in several publications keeping in that case the most complete and up-to-date version of the work.

In terms of the contents, studies to be included in the review shall contain information about the scientific programming methods, Big Data technologies, and datasets (if any) that have been used. Since a huge amount of works (thousands) can be found in the field of Big Data technologies, only those papers related to scientific programming techniques shall be selected. However, in some cases, such as the use of hybrid approaches between scientific programming and artificial intelligence implies the need of extending the scope to explore other possibilities that may have impact in the current state of the art of scientific programming for data-intensive engineering environments.

In the same manner, novel techniques applied to other domains such as earth sciences, biology, meteorology, or social sciences may be included to check whether such works represent progress regarding the current techniques.

As a result of the application of the inclusion criteria, Figure 1 shows the number of papers published per year and database (Figure 2). According to the trendline, there is increasing interest in the creation of know-how around these techniques, methods and technologies having in the last 5 years an increment of a 56% of the number of published papers just in journals what indicates a rising demand of new techniques to take advantage of all the technology and approaches that are currently available. In the case of the databases, Springer is becoming a key player promoting the research and innovation in the data science and engineering areas, Figures 3 and 4 also shows the distribution of papers per year and database. However, the type of search engine in each database may have impact in the number of articles that are returned as the result explaining the big difference between Springer and the rest of publishers.

(5) *Selection Procedure*. Those papers fulfilling the inclusion criteria are summarized in a spreadsheet including the main facts of each work. The proposed approach seeks for ensuring that papers suitable for review must first pass a quality check. More specifically, the identifier, title, abstract, keywords, conclusions, methods, technology, measures, scope, and domain are extracted to create a table of metainformation that serve us to present the information to two different experts to decide whether the work is finally selected for review or not. To do so, a quantitative value is assigned to each entry being that: 1: applicable; 0.5: unknown; 0: not applicable. This approach allows us to accomplish with the two-fold objective of having a qualitative and quantitative selection procedure and follow the guidelines established in the PRISMA model [35].

As a result, Figure 5 depicts the number of works that have been identified after searching in the database (1193), the number of works that have been screened successfully (935) and excluded (258), the number of works selected for quality assessment (322) and, finally, the number of works that have been included in the review (114).

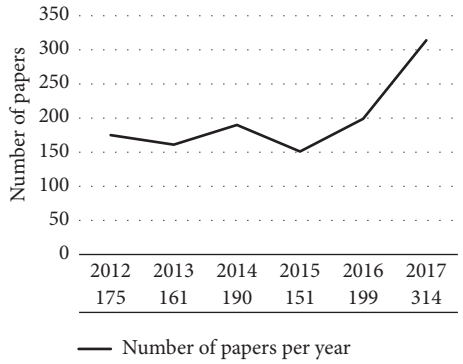


Figure 1: Distribution of the number of papers per year.

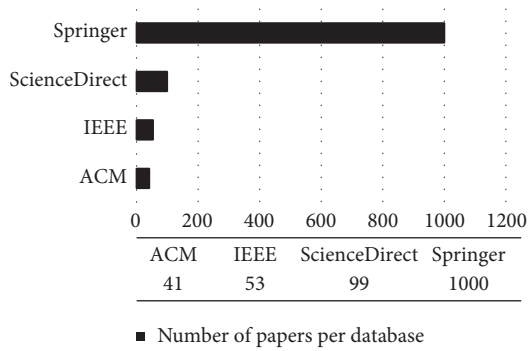


Figure 2: Distribution of the number of papers per database.

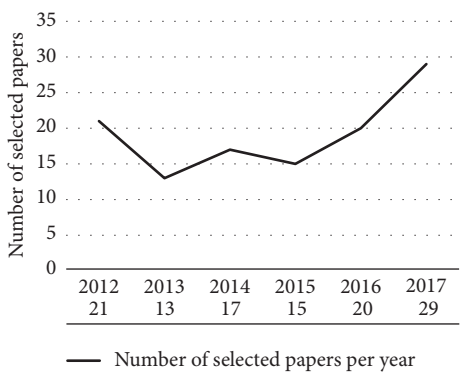


Figure 3: Distribution of the number of selected papers per year.

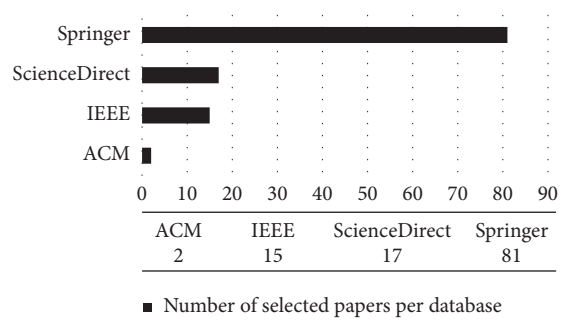


Figure 4: Distribution of the number of selected papers per database.

(6) *Evaluation Procedure and Data Extraction Strategy*. In this step, the selected papers are evaluated according to a Likert-scale 1–6 being 1 applicable but not fully representative for the scientific programming community and 6 applicable and fully representative. In this way, it is possible to create a sort of the selected papers to finally select those which are beyond a value of 4. In this manner, only papers beyond the average are then evaluated. To provide a proper classification and mapping of the works, the topics covered by the works are organized in broader themes. This classification looks for organizing the works providing a comprehensive view of the existing state of the art. In case of a work that can be applied to different domains, it is classified as a general domain work.

(7) *Synthesis of Data and Analysis of Results*. The synthesis of findings in different studies is hard to draw since many factors can affect a data-intensive environment. In this work, after passing the qualitative and quantitative evaluation, we have realized that the best approach is to group the different works according to themes relevant to a specific field. In this manner, we have selected, following a typical Big Data architecture [36], the next dimensions: infrastructure, software technique/method, and application domain. All data generated during this review are publicly available in the following repository: https://github.com/catedra-rtve-uc3m/smart-public/tree/master/papers/data-scientific-programming-review-2018.

4. Analysis of Results

In the field of data-intensive environments, it is necessary to separate the different responsibilities and aspects of both hardware and software components. To do so, as it has been previously introduced, the NIST (National Institute of Standards and Technology) has published the "Big Data Interoperability Framework [36]" allowing us to perfectly define and sort responsibilities and aspects of functional blocks in a Big Data environment.

In this work, we take as a reference this architectural framework simplifying and grouping works in three big themes: hardware infrastructures, software components being that techniques, methods, algorithms, and libraries and applications.

In the first case, hardware resources are becoming critical to offer high-performance computing (HPC) capabilities to data-intensive environments [29]. Last times have seen the growing interest to take advantage of new hardware infrastructures and techniques to optimize the execution of time-consuming applications. In this sense, the use of GPUs (graphical processing units) is a cornerstone to accelerate data-intensive applications being a critical component for performing complex tasks keeping a balance between cost and performance. More specifically, GPUs and CUDA (GPU and a programming language) are being currently used to train machine learning algorithms decreasing the training time from weeks to days or even hours. The potential use of

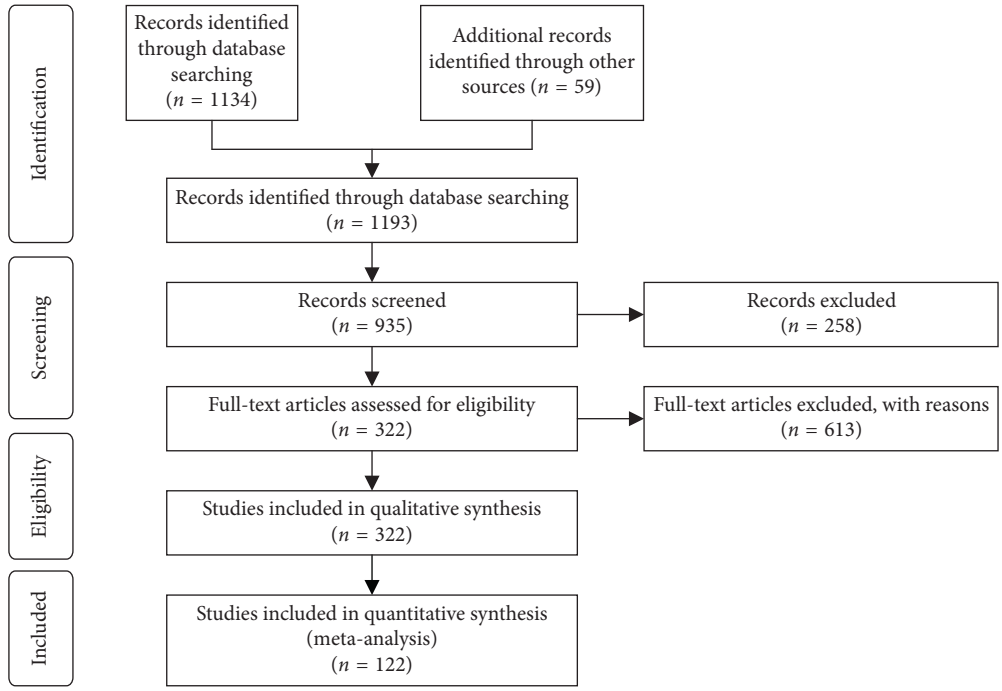


Figure 5: PRISMA flow diagram of the systematic review.

many processing cores enables algorithms to parallelize data and processing in a real multithread environment taking advantage of a high computing power, a large memory bandwidth and, in general, a very low power consumption. However, the main drawback of this type of architecture relies on the difficulties to design and code algorithms due to the expressivity of the language and the need of handling interdependencies in a highly parallelized environment.

FPGAs (field-programmable gate arrays) are other type of integrated circuit and programming language (hardware description language, a kind of block-based language) to run jobs repeatedly avoiding unnecessary processing cycles. It perfectly fits to problems in which repetitive tasks must be done several times such as pattern matching. However, the on-the-fly configuration of a new algorithm is not flexible as in GPUs or CPUs, and more advanced programmer toolkits are still missing.

Multiprocessor architectures are hardware infrastructures based on the use of several CPUs in the same computer system (commonly a super computer). According to the well-known Flyinn's taxonomy, multiprocessors-based systems lay on different configurations depending on how data are input and processed. Shared memory and message passing are also typical synchronization mechanisms of these computer systems that usually present symmetry in the type of CPUs and a set of primitives to easily process data, e.g., vectors.

Finally, grid computing represents a set of geographically distributed computer resources that are combined to reach a common goal. Several configurations and processing schemes can be found in each node for processing different jobs and workloads. To manage a grid computing system, a general-purpose middleware is used to synchronize both data sharing and job processing what is considered critical in a data-intensive environment where a data management strategy is required to organize, store, and share data items across the grid computing system.

As a final remark, parallelization and distribution are again general approaches [12] for improving performance while executing time-consuming tasks. This situation is also applicable to data-intensive environments where the combination of large amounts of data and complex calculations require new strategies and configurations to avoid the bottlenecks produced by the need of synchronizing data.

Secondly, techniques, methods, and algorithms for scientific computation are generally complex implying the need of optimization techniques to avoid recomputation. When large amounts of data meet this type of techniques, this situation is becoming critical since complexity is increased due to the processing of large inputs and intermediate results (e.g., cosimulation in engineering). That is why, it is possible to find combination of different techniques trying to tackle the problem of implementing a technique while keeping a reasonable balance between time and cost.

In this sense, after screening existing works, a plethora of techniques and methods was found trying to provide innovative ways of solving existing problems. With the aim of grouping and sorting them adequately, a first classification was made according to the 2012 ACM Computing Classification System at different levels of granularity. To do so, techniques and methods were grouped under general themes trying to make a comprehensive and clear distinction between the type of problem and the technique and/or method. In some cases, it was not completely possible to make such clear distinction since the same

technique can be applied to solve a huge set of problems (e.g., deep learning).

(i) Artificial intelligence (AI): nowadays, artificial intelligence techniques are spread to solve a huge variety of problems from space observation to autonomous driving. Robotics, medicine, and many other domains are being affected by AI in which more computational power and a change of the people mindset is still required. This category technique tries to equip machines with intelligence to be able to perceive its environment and take actions reaching specific goals. Some techniques such as deep learning, fuzzy logic, gene programming, neural networks, or planning methods fall in this category. Although AI techniques can directly be applied to solve particular problems, in the context of this review, they are usually combined with other techniques to optimize the resolution of a more complex problem.

(ii) Computation architecture: as it has been previously introduced, a hardware setting may have a critical impact in terms of time and costs. Instead of lowering the configuration to a hardware level, software-defined architectures and methods are used to provide a virtual infrastructure that practitioners can easily use and configure. Infrastructure customization, scheduling techniques, and high-level design of workflows can be found as abstract methods to manage complex hardware settings.

(iii) Computation model: a computational model can be defined as the formal model that is applied to solve a complex system. In this case, the computational models that have been selected refer to the different strategies to manage, synchronize, and process data and tasks. To do so, event calculus, message passing interface (MPI), parallel programming, query distribution, and stream processing are the main models that have been found in the review.

(iv) Computational science: complex problems may require the collaboration of different disciplines to provide innovative solutions. Computational science, and more specifically its application to the engineering domain, is the core of this review. The use of models, simulations, and many other techniques such as Euler models or statistical methods are widely spread to understand and model the behavior of complex systems such as those that can be found in engineering. However, it seems that the combination of these techniques under different hardware settings and software models is being a trend that must be systematically observed and evaluated.

(v) Graph theory: last times have seen an increasing use of graph theory to model problems in domains such as social network analysis, telecommunications, or biology. Any problem that can be modeled as a network is a good candidate to apply graph theory techniques understanding how the network is built and to infer new relationships through the exploitation of the underlying knowledge coded in the different nodes, relationships, and layers. Multilayered and multimode networks are a common logical representation for relational data that can be used to perform tasks of structural analysis [37] or relational learning [38]. It is also well known that techniques and methods in this domain [39] are usually complex and time-consuming. That is why, new data-intensive environments may require taking advantage of techniques such as complex network analysis or automata, but they will also require a good number of computational resources.

(vi) Engineering: building on the notions presented in computational science, the different disciplines in engineering make intensive use of formal models, simulations, and numerical and nonnumerical methods for building complex systems. In this case, methods such as finite elements and simulation have been identified as critical techniques already available in the literature as candidates to make use of data-intensive environments.

(vii) Machine learning: it is considered a brand of artificial intelligence and defined as a set of techniques to equip machines with intelligence in changing environments. Basically, a model is built through a training method that it is then evaluated in the testing phase. Large amounts of data directly impact the performance of both phases, so it is possible to find libraries that can work stand-alone (e.g., Weka, Python, and Scikit) or in a distributed environment (e.g., Spark Mlib). Machine learning techniques based on unsupervised and supervised learning methods are commonly applied to solve multiple problems in classification, computer vision, data mining, natural language processing and text mining, sentiment analysis, opinion mining, speech systems, pattern recognition, or question answering to name just a few. Once a model is created using a concrete technique, it can easily be used to perform predictive and prescriptive analysis processes. Potential applications of the techniques falling in this category range from social network, medicine, or biology to logistics or manufacturing. In this review, Bayesian machine learning, data mining, information fusion, pattern recognition, predictive models, support vector machines, and regression models have been found as representatives and popular types of problems and techniques in machine learning for engineering in combination with other scientific programming mechanisms.

(viii) Mathematics and applied mathematics: the use of strong theoretical foundations is critical to build complex systems such as those in engineering.

Mathematics (in this review, mainly Algebra) and, more specifically, applied mathematics offer use of a set of formal methods in science, engineering, computer science, and industry to solve practical problems. Gradient descent optimization, integer linear programming, linear algebra/solvers, linear and nonlinear programming, matrix calculation, numerical methods, and symbolic execution represent a set of common types of problems and techniques widely studied and applied to the scientific programming area.

(ix) Programming techniques: programming as a practice represents the main driver of implementation of a system after the analysis and design processes. Different programming techniques can be found to optimize the development of algorithms and take advantage of the different programming language primitives. In this sense, constraint programming, cube computation, dynamic programming, domain specific languages (DSL), and stochastic programming have been found as representative programming techniques in a scientific environment to take the most data and perform tasks such as analysis or data quality checking.

Once the hardware settings and the main methods, techniques, and algorithms at a software level have been presented, a set of tentative application domains can be outlined. In this light, the review has found two main directions: (1) engineering domains such as Aerospace, Automotive, Civil engineering, Cyberphysical systems, Feature engineering, Industrial applications, Iron mining, Manufacturing or Nuclear domain, and (2) other domains such as Earth science, Geometry, Image analysis, Internet of things, Medicine, Scientific research, Social Sciences, or Spatial modelling.

All these domains share some common characteristics and needs strong mathematical foundations to govern the different systems, processing of large amounts of data, application of methods, techniques, and models such as simulation and, in general, criticality (they are considered safety-critical or life-critical systems; a failure can imply death or serious injury to people and severe damage to equipment or environment). That is why, the emerging use of innovative hardware architectures and exploitation of data must be controlled by strict requirements that make scientific programming techniques even more important than in any other application domain.

4.1. Hardware Infrastructure for Data-Intensive Engineering Environments. The previous section has introduced the main hardware settings for data-intensive environments. In Table 3, a mapping between the hardware architectures and the different software-based techniques is presented.

The use of GPUs (and CUDA) [41–50, 54–56, 58, 60] is a widely accepted hardware setting for providing a hardware infrastructure to process large amounts of data regardless of the type of software technique. In this manner, the review has found 18 out of 24 (75%) works in this field. GPUs clearly represent a major step to optimize the execution of complex tasks iterating over data. Here, the possibility of reducing the number of repetitions in terms of data processing becomes relevant and can be critical for some time-restricted domains in which it is necessary to make decisions in near real-time. More specifically, GPUs are mainly used to solve linear algebra problems [54–56] or to perform large-scale matrix calculations [54–56, 58]. These are common problems in the field of engineering.

Other alternatives for large-scale computational infrastructures seem to be the traditional environments for parallel (FPGA and multiprocessor systems) and distributed computing. In the first case, the works that the review has found in the field of FPGAs [53] are not very representative (just 1 out of 24) while multiprocessor architecture works [40, 51, 57, 59] are still relevant (4 out of 24). Although FPGAs represent a powerful alternative to GPUs, the lack of (1) abstraction to code programs and (2) of a kind of hot-deployment method makes this alternative not very attractive for data science and engineering.

In the case of multiprocessor architectures, they represent the traditional powerful and expensive data processing centers for high-performance computation that are available in large research institutions. As it has been introduced in the first section, the research efforts [29] to merge Big Data and HPC are becoming popular to take advantage of existing large-scale computational infrastructures. Moreover, multiprocessor architectures also provide high-level APIs (application programming interfaces) that allow researchers and practitioners to easily configure, deploy, and run computing-intensive tasks. Finally, grid computing approaches [52] that have been widely spread for the deployment of cost-effective Big Data frameworks are not yet a popular option for scientific computation of large amounts of data.

4.2. Software Methods and Techniques for Data-Intensive Engineering Environments. To present the meta-analysis of the works regarding the typology of techniques and domains, Table 4 groups the different works making a mapping between the type of technique and its application domain. According to this table, AI techniques are the most prominent methods, representing the 21.43% of the reviewed works, to tackle problems in different sectors what is completely aligned to new perspectives open by the Industry 4.0 and digitalization processes.

Mathematics and applied mathematics methods still represent a 16.07% of the existing works bringing formal foundations for the computation of large amounts of data applying techniques such as integer linear programming, linear algebra, or symbolic execution. This situation makes sense since engineering domains are based on modelling physical systems using mathematics; so the existence and generation of more data only reinforces the need of improving the performance of existing methods to deal with more and greater amounts of data. This situation also affects computation architectures, trying to improve infrastructure

Table 3: Mapping between methods/models/techniques and hardware domains.

		GPU	FPGA	Multiprocessor	Grid computing
Artificial intelligence	*Deep learning* *Fuzzy logic* *Gene programming* *General techniques* *Neural networks* *Planning*				
Computational architecture	*Infrastructure* *Scheduling techniques* *Workflow*			[40]	
Computation model	*Event calculus* *MPI* *Parallel programming* *Query distribution* *Stream processing*	[41, 42] [43]			
Computational science	*Euler models* *Scientific computation* *Statistical methods*	([44], p. 2)			
Graph theory	*Automata* *Complex network analysis*				
Engineering	*Finite elements* *Simulation*	[45] [46, 47]			
Machine learning	*Bayesian machine learning* *Data mining* *Information fusion* *Pattern recognition* *Predictive models* *Support vector machines* *Regression model*	[48] [49, 50]	[53]	[51]	[52]
Mathematics and applied mathematics	*Gradient descent search* *Integer linear programming* *Linear algebra/solvers* *Linear programming* *Matrix calculation* *Nonlinear programming* *Numerical methods* *Symbolic execution*	[54–56] [54–56, 58]		[57] [59]	
Programming techniques	*Constraint programming* *Cube computation* *Dynamic programming* *DSL* *Stochastic programming*	[60]			

Table 4: Mapping between methods/models/techniques and domains including aggregated percentages.

	General application to engineering	Engineering	Nonengineering	Aggregated (%)
Artificial intelligence	[62, 63] (1.79%)	[64–73] (8.93%)	[74–85] (10.71%)	21.43%
Computational architecture	[86–88] (2.68%)	[89–95] (6.25%)	[90–95] (5.36%)	14.29%
Computation model	[96–98] (3.57%)	[99–101] (2.68%)	[102, 103] (1.79%)	8.04%
Computational science	[104] (0.89%)	[85, 105, 106] (2.68%)	[107] (0.89%)	4.46%
Graph theory	[37, 108–117] (9.82%)		[118, 119] (1.79%)	11.61%
Engineering methods	[96, 120–124] (5.36%)			5.36%
Machine learning	[125, 126] (1.79%)	[127] (0.89%)	[128–130] (2.68%)	5.36%
Mathematics and applied mathematics	[62, 88, 131–136] (7.14%)	[70, 137–141] (5.36%)	[141–144] (3.57%)	16.07%
Programming techniques	[74, 114, 122, 145–147] (5.36%)	[69, 148] (1.79%)	[149] (0.89%)	8.04%
General software application	[89, 115, 150–153] (5.36%)			5.36%
Aggregated (%)	43.75%	28.57%	27.68%	100/100%

settings via software-defined systems. Works in this area represent around 14.29% of the selected articles and, in general, they look for easing the creation of large software-defined infrastructures and APIs that can be used by the previous areas of AI and Mathematics and Applied Mathematics.

Graph techniques (11.61%), computational models (8.04%), and programming techniques (8.04%) represent the midclass methods in this classification. It is especially relevant to highlight the number of works that have emerged in the field of complex network analysis. Historically, large graph processing was a very time- and resource-consuming task that was preventing the broad use of these techniques. Currently and since new hardware infrastructures and APIs are available for scientific computation of large graphs, these techniques have gained momentum and they are being applied to different domains such as social network analysis, telecommunications, or biology.

Finally, the last section of the classification includes very specific works in the areas of Machine learning (5.36%), Engineering methods (5.36%), and Computational science (4.46%) that could be classified in other broader areas, but they represent concrete and representative works for data-intensive engineering environments.

On the other hand, selected works can be aligned to their scope in the different domains. In this sense, it has been found that techniques that can be applied to engineering (but not directly designed for engineering) represent a 43.75% what means that existing papers are reporting works to offer general-purpose solutions instead of solving specific problems. This is also a trend in the digitalization [61] in which one of the main cornerstones is the delivery of platforms ("Platform Economy") that can be extended and enable people to build new things on top of a baseline technology. In the case of Engineering, the selected works represent a 28.57% focusing on specific engineering problems that are now facing some new data landscape where existing techniques must be reshaped to offer new possibilities and more integrated and smart engineering methods. Finally, the review has also included a 27.68% of works in other domains to demonstrate that scientific computation and data are also an open issue in the other science-oriented domains. The two major conclusions of this meta-analysis are as follows:

(1) Engineering is not being indifferent to data-driven innovations. Assuming that more data will imply more knowledge, engineering methods are trying to find new opportunities in a data world to become more optimized and accurate.

(2) Hybrid approaches mixing existing techniques such as computational science and mathematics (deterministic) and AI techniques (probabilistic) are an emerging research area to optimize the use and exploitation of data.

In both cases, hardware- and software-defined infrastructures will play a key role to provide a physical or virtual environment to run complex tasks consuming and generating large amounts of data (e.g., simulations).

Once a whole picture of the techniques and methods in use for engineering has been depicted, it is necessary to identify and align existing research to specific engineering domains. To do so, Table 5 shows a mapping between the different methods and techniques and its application to the engineering domains.

Here, the development and operation of safety-critical systems implies an implicit need of managing data, information and knowledge that is generated by the different engineering teams. In this light, it is necessary to emphasize the research advances in Aerospace [105, 137, 138, 150] and Automotive [64, 65, 72, 89, 139] engineering. In both cases, the "one-size fits all" is again not true. Although formal methods are completely necessary to model this kind of complex systems, it is also necessary to combine different techniques that can shift the current practice in systems engineering. Existing engineering processes such as requirements engineering, modelling, simulation, co-simulation, traceability, or verification and validation are now completely impacted by an interconnected data-driven environment in which interoperability, collaboration, and continuous integration of system components are completely necessary.

Civil engineering [66, 67, 151], Industrial applications [148], Iron Mining [70], Manufacturing [73] Nuclear domain [143] works are also relevant in terms of using a huge up-to-date variety of techniques to exploit data that are continuously being generated by tools, applications, sensors, and people.

Finally, other rising sectors completely fueled by data such as CPS [71] or Feature Engineering [127] represent the update of classical engineering disciplines in aerospace, automotive, or robotics. As a final remark, general purpose engineering techniques have been also reported [68, 69, 99–101, 106, 140, 152, 154], making use of data, scientific programming techniques, and hybrid approaches.

On the other hand, as it has been previously introduced, the present work also looks for reviewing the impact of new techniques and methods in other nonengineering domains (Table 6). Earth science [78, 79, 84, 102, 149] is a common domain in which large amounts of data are generated by satellites and other data sources that must be processed to provide services such as environmental management and control, urban planning, and so on.

Scientific programming techniques are becoming even more relevant to Geometry [77, 80, 118, 130] paving the way to solve complex problems in areas such as civil engineering, physics, or architecture. In the case of image analysis [75, 76, 81, 144], it is possible to find techniques that are now gaining popularity for equipping smart cars with autonomous driving capabilities, identifying people, assisting physicians in medical diagnosis and surgery, or drone field monitoring with object recognition capabilities. More specifically, Medicine [74, 115, 128, 129] is taking advantage of scientific programming techniques and data for improving its processes of health monitoring and prevention. Social sciences [83] or Spatial modelling [94, 103] are also offering new capabilities such as social network analysis or simulation through the application of existing scientific

Table 5: Mapping between methods/models/techniques and engineering domains.

		General application to engineering	Aerospace	Automotive	Civil engineering	Cyberphysical-systems	Feature engineering	Engineering methods	Industrial applications	Iron mining
General software application			[150]	[89]	[151]			[152]		
Artificial intelligence	*Deep learning*									
	Fuzzy logic									
	General techniques			[64, 65]	[66, 67]					
	Gene Programming							[68]		
	Neural networks	[62, 63]						[69]		[70]
	Reasoning techniques					[71]				
	Planning			[72]						
Computational architecture	*Infrastructure*	[86–88]								
	Scheduling techniques			[89]						
	Workflow									
Computation model	*Event calculus*	[96]						[99–101]		
	MPI	[97]								
	Parallel programming									
	Query distribution	[87, 98]								
	Stream processing									
Computational science	*Euler models*									
	Scientific computation	[104]								
	Statistical methods		[105]					[106, 154]		
Graph theory	*Automata*	[124]								
	Graph/complex network analysis	[37, 108–117]								
Engineering methods	*Finite elements*	[155]								
	Simulation	[96, 120–124]								
Machine learning	*Bayesian machine learning*	[125, 126]								
	Data mining									
	Information fusion									
	Pattern recognition									
	Predictive models									
	Support vector machines									
	Regression model						[127]			
Mathematics and applied mathematics	*Gradient descent search*									
	Integer linear programming	[88]								
	Linear algebra/solvers	[131, 132]	[137]	[139]						
	Linear programming	[136]								
	Matrix calculation	[133]								
	Nonlinear programming	[62]	[138]							[70]
	Numerical methods							[140]		
	Symbolic execution	[134, 135]								
Programming techniques	*Constraint programming*	[74, 145–147]								
	Cube computation									
	Dynamic programming	[114]								
	Domain specific languages	[153]								
	Stochastic programming	[122]						[69]	[148]	

Table 6: Mapping between methods/models/techniques and other data-intensive domains.

		Earth science	Geometry	Image analysis	Internet of things	Manufacturing	Medicine	Nuclear domain	Scientific research	Social sciences	Spatial modelling
General software application							[115]				
Artificial intelligence	*Deep learning*			[75, 76]							
	Fuzzy logic										
	General techniques		[77]								
	Gene Programming	[78]				[73]					
	Neural networks	[79]	[80]	[81]					[82]	[83]	
	Reasoning techniques										
	Planning	[84]	[85]				[74]				
Computational architecture	*Infrastructure*								[90–93]		[94]
	Scheduling techniques										
	Workflow								[95]		
Computation model	*Event calculus*										
	MPI										
	Parallel programming	[102]									
	Query distribution										
	Stream processing										[103]
Computational science	*Euler models*										
	Scientific computation								[107]		
	Statistical methods										
Graph theory	*Automata*										
	Graph/complex network analysis		[118]							[119]	
Engineering methods	*Finite elements*										
	Simulation										
Machine learning	*Bayesian machine learning*						[128]				
	Data mining										
	Information fusion						[129]				
	Pattern recognition										
	Predictive models		[130]								
	Support vector machines										
	Regression model										
Mathematics and applied mathematics	*Gradient descent search*			[144]							
	Integer linear programming										
	Linear algebra/solvers								[141]		
	Linear programming				[142]						
	Matrix calculation							[143]			
	Nonlinear programming										
	Numerical methods										
	Symbolic execution										
Programming techniques	*Constraint programming*										
	Cube computation										
	Dynamic programming										
	Domain specific languages	[149]									
	Stochastic programming										

programming techniques over large datasets. Finally, this review has also found a good number of works in the field of scientific research [82, 90–93, 95, 107, 141] providing foundations for analysis and exploitation of data for other domains.

4.3. Additional Remarks about Benchmarking and Datasets. As it has been reviewed in the previous sections, a good number of hardware configurations and software frameworks to deal with large amounts of data in different domains can be found. However, it is also relevant to briefly introduce the notion of benchmarking for large-scale data systems as a method to evaluate this huge variety of tools and methods. In general, benchmarking comprises three main stages [156]: workload generation, input data or ingestion, and calculation of performance metrics. Benchmarking has been also widely studied [156, 157] in the field of Big Data systems and the main players in the industry [158] such as TPC ("Transaction Processing Performance Council"), SPEC ("the Standard Performance Evaluation Corporation") or CLDS ("Center for Large-scale Data System Research") have published different types of benchmarks that usually fall into three categories: microbenchmarks, functional benchmarks, or genre-specific benchmarks. The Yahoo! Cloud Serving Benchmark (YCSB) Framework, the AMP Lab Big Data Benchmark, BigBench [158], BigDataBench [159], BigFrame [160], CloudRank-D [161], or GridMix are some of the benchmarks that have been evaluated in [157] apart from others [157, 162] directly designed for Hadoop-based infrastructures such as HiBench, MRBench, MapReduceBenchmarkSuite (MRBS), Pavlo's Benchmark, or PigMix. Benchmarking is therefore a key process to ensure the capabilities of the different hardware settings and software frameworks in comparison with others.

Finally, it is also convenient to point out the possibility of accessing large datasets (open, free, and commercial) for computational and research purposes. These large datasets are usually managed by public institutions such as the European Data Portal (https://www.europeandataportal.eu/) (European Union), the Data.gov initiative (https://catalog.data.gov/dataset) (U.S. General Services Administration), research centers such as "Institute for Data Intensive Engineering and Science" (http://idies.jhu.edu/research/) (John Hopkins University), or large research projects such as Big Data Europe (https://www.big-data-europe.eu/). However, one of the best options to search and find high-quality datasets (license, provider, size, domain, etc.) is the use of some aggregating service such as the Amazon AWS Public Dataset Program, Google Public Data Explorer, or the most recent Google Dataset Search (https://toolbox.google.com/datasetsearch) (September 2018) that indexes any dataset published under a schema.org description. Kaggle (https://www.kaggle.com/datasets) and services organizing data-based competitions are also a good alternative to find complete and high-quality datasets. As a final type of data source, public APIs such as those from large social networks (e.g., Twitter, Linkedin, or Facebook) or specific sites such as GeoNames can also be used to access a good amount of data under some restrictions (depending on the API terms of service).

5. Answer to the Research Question and Future Challenges

The main objective of this systematic review was to provide a comprehensive and clear answer to the following question:

Which are the last techniques, methods, algorithms, architectures, infrastructures in scientific programming/computing for the management, exploitation, inference, and reasoning of data in engineering environments?

According to the results provided in the previous section and the meta-analysis, it is possible to define the different techniques, methods, algorithms, architectures, and infrastructures that are currently in use for scientific programming in data-intensive engineering environments. More specifically, in terms of hardware infrastructures, systems based on GPUs are now the main type of hardware setting to run complex and time-consuming tasks. Other alternatives such as FPGAs, multiprocessor architectures, or grid computing are being used for solving specific problems. However, the main drawback of FPGAs lies on the need of higher levels of abstraction to ease the development and deployment of complex problems. In the case of multiprocessor architectures, they are becoming popular since there is an effort to merge Big Data and HPC, but there is not yet so much works reporting relevant works in this area. Finally, grid computing represents a good option for Big Data problems where distribution is a key factor. However, when complex computational tasks must be executed, it does not seem to be a good option due to the need of synchronizing processes and data. In terms of architectures, there are also a good number of works looking for creating software-defined infrastructures easing the configuration and management of computational resources. In this sense, research works in the field of workflow management represent a trend to manage both the data life cycle and the execution of complex tasks.

On the other hand, scientific programming techniques and computational science methods such as integer linear programming, linear algebra/solvers, linear programming, matrix calculation, nonlinear programming, numerical methods or symbolic execution are being challenged by a complete new data environment in which large datasets of information are available. However, all these techniques have strong theoretical foundations that will be necessary to tackle problems in engineering. Moreover, programming techniques such as constraint and *stochastic* programming seem to be a good option to implement existing formal models in engineering.

This review has also identified that graph-based techniques, mainly complex network analysis, are becoming popular since a good number of problems can be represented as a set of nodes and relationships that can then be analyzed using graph theory. Considering that graph analysis techniques are based on matrix calculations, the possibility of having high-performance scientific methods

and infrastructures to execute such operations is being a key enabler for this type of analysis.

A relevant outcome of this review comes with the identification of AI and machine learning techniques as a counterpart of traditional scientific programming techniques. In the era of the 4th industrial revolution, the possibility of equipping not just machines but existing software techniques with intelligence is becoming a reality. These techniques are used to prepare the input of existing methods, to optimize the creation of computational models, and to learn and provide feedback on existing scientific programming techniques depending on the output. Although not every process, task, or engineering method is expected to include AI techniques, the review outlines the current trend of merging deterministic and probabilistic approaches. However, the use of AI is still under discussion since the impact of AI in engineering processes such as certification or assurance of safety-critical systems is completely open, e.g., autonomous driving.

Finally, the main engineering domains affected by huge amounts of data are Aerospace and Automotive. In these safety-critical sectors, engineering claims for innovative methods to enable collaboration between processes, people, and tools. The development and operation of a complex system cannot be anymore a set of orchestrated tasks via documents but a data-driven choreography in which each party can easily exchange, understand, and exploit data and information generated by others.

5.1. Future Challenges. Data-driven systems are already a reality ready to use. Big Data technologies and tools are enough mature and continuously improved and extended to cover all the stages of the data life cycle management. Capabilities for data ingestion, cleaning, representation, storage, processing models, and visualization are also available as stand-alone applications or as a part of a Big Data suite. Many use cases have successfully applied Big Data technology to solve problems in different sectors such as marketing, social network analysis, medicine, finances, and so on. However, a paradigm shift requires some efforts [163, 164] in the following topics:

(1) A reference and standardized architecture is necessary to have a separation of concerns between the different functional blocks. The standardization work in [36] represents a first step towards the harmonization of Big Data platforms.

(2) A clear definition of interfaces to exchange data between data acquisition processes, data storage techniques, data analysis methods, and data visualization models is completely necessary.

(3) A set of storage technologies to represent information depending on its nature. A data platform must support different types of logical representations such as documents, key/values, graphs, or tables.

(4) A set of mechanisms to transport (and transform) data between different functional blocks, for instance, between the data storage and analysis layers.

(5) An interface to provide data analysis as a service. Since problems to be solved and data may have different nature and objectives, it is necessary to avoid a kind of vendor lock-in problem and support a huge variety of technologies to be able to run diverse data analysis processes. Thus, it is possible to take the most of the existing libraries and technologies, e.g., libraries in R, Python, or Matlab. Here, it is also important to remark again that "no one size fits all" and hybrid approaches for analytical processes may be considered for the next generation of Big Data platforms.

(6) A set of processing mechanisms that can minimize the consumption of resources (in-memory, parallel, and distributed) maximizing the possibilities of processing data under different paradigms (event, stream, and batch) and analyzing data with different methods and techniques (AI, machine learning, scientific programming techniques, or complex network analysis) are also required.

(7) A management console to monitor the status of the platform and the possibility of creating data-oriented workflows (like the traditional methodologies in business process management but focusing on data). Data-intensive environments must take advantage of last technologies in cloud computing and enable users and practitioners to easily develop and deploy more complex techniques for the different phases and activities of the data life cycle.

(8) A catalogue of available hardware and software resources. A Big Data platform must offer capabilities to manage computational resources, tools, techniques, methods, or services. Thus, the operator can configure and build their own data-driven system combining existing resources.

(9) A set of nonfunctional requirements. Assuming that scalability can be easily reached through the use of cloud computation environments, interoperability, flexibility, and extensibility represents major nonfunctional characteristics that future platforms must include.

(10) A visualization layer or, at least, a method to ingest data in existing visualization platforms must be provided to easily summarize and extract meaning of huge amounts of data.

Apart from these general-purpose directions for the future of data-intensive environments, the engineering domain must also reshape the current methods to develop complex systems. Engineering is not anymore an isolated activity of experts in some discipline (software, mechanics, electronics, telecommunications, etc.) that produces a set of work products that serve us to specify and build a system.

Products and services are becoming complex and that complexity also implies the need of designing new ways of doing engineering. Tools, applications, and people

(engineers in this case) must collaborate and improve the current practice in systems engineering processes through the reuse of existing data, information, and knowledge. To do so, data-intensive engineering environments must be equipped with the proper tools and methods to ease processes such as analysis, design, verification and validation, certification, or assurance. In this light, scientific programming techniques must also be enriched with new and existing algorithms coming from the AI area. Hybrid approaches of techniques to solve complex problems represent the natural evolution of scientific programming techniques to take advantage of AI models and existing infrastructure.

5.2. Data-Intensive Engineering Environments and Scientific Programming. The development of complex engineering systems is challenging the current engineering discipline. From the development life cycle to the operation and retirement of an engineering product, an engineering product is now a combination of hardware and software that must be designed considering different engineering disciplines such as software engineering, mechanics, telecommunications, electronics, and so on. Furthermore, all technical processes, perfectly describe in standards such as the ISO/IEC/IEEE 15288:2015 "Systems and software engineering—System life cycle processes," are now more interrelated than never. A technical process implemented through an engineering method requires data and information that has been generated in previous development stages enriching the current engineering practice.

From a technical perspective, interoperability is becoming a cornerstone to enable collaborative engineering environments and to provide a holistic view of the system under development. Once data and information can be integrated together, new analytical services can be implemented to check the impact of a change, to discover traces, or to ensure the consistency of the system. However, it is necessary to provide common and standardized data models and access protocols to ensure that it is possible to build an integrated view of the system such as a repository. In this sense, the Open Services for Lifecycle Collaboration (OSLC) initiative applying the principles of Linked Data and REST and the Model-Based Systems Engineering approach are two of the main approaches to create a uniform engineering environment relying on existing standards. Approaches such as the OSLC Knowledge Management (KM) specification and repository [165] or the SysML 2.0 standard are defined under the assumption of providing standardized APIs (application programming interfaces) to access any kind of system artifact (in the case of OSLC KM) or model (in the case of SysML 2.0) meaning that the exchange of system artifacts is not anymore a single and isolated file but a kind of service. In general, engineering environments are already fueled by interconnected data shifting the paradigm of data silos to a kind of industrial knowledge graph. The impact of having an interconnected engineering environment relies on the possibility of increasing the time to release an engineering product or service meeting the evolving needs of customers.

In the operational environment, complex engineering systems are an example of Internet of Things (IoT) products in which thousands of sensors are continuously producing data that must be processed with different goals: prediction of failures, making decisions, etc. This environment represents a perfect match for designing and developing analytical services. Examples of data challenges for specific disciplines can be found in the railway sector [166], aerospace [167], or civil engineering [168] where Big Data technologies are mainly used to exploit data and information generated during the operation of the system.

However, in both cases, it is possible to find some common barriers to implement a data management strategy: data privacy, security, and ownership. For instance, in the automotive industry, it would be nice that the data used to validate the behavior of an autonomous car were shared among car manufacturers to ensure that all cars accomplish with a minimum level of compatibility and to ease the activity of certification bodies. Nevertheless, it seems also clear that this will not happened since it can represent a competitive advantage in a market-oriented economy. That is why, it is completely necessary to design policies at a political level that can ensure a proper development of new engineering products and services where manufacturers must focus on providing better user experiences under a common baseline of data.

Finally, it is necessary to remark again that scientific programming techniques have been widely used to solve complex problems in different domains under several hardware settings. The rising of Big Data technologies has created a new data-intensive environment in which scientific programming techniques are still relevant but facing a major challenge: How to adapt existing techniques to deal with large amounts of data?

In this context, scientific programming techniques can be adapted at hardware or software (platform, technique, or application) levels. For instance, it is possible to find again examples of GPUs architectures [169] to improve the practice in parallel programming for scientific programming. Cloud-based infrastructures [170, 171] and platforms for analytics [172] are another field of study. In the case of foundations of scientific programming, the work in [173] reviewing the main foundations of scientific programming techniques and the use of pattern matching techniques in large graphs [174] are examples of improvements and works in the scope of software techniques. Finally, applications in coal mining [175], recommendation engines for car sharing services [176], health risk prediction [177], text classification [178], or information security [179] are domains in which data are continuously being generated representing good candidates to apply scientific programming techniques.

6. Conclusions

Data are currently fueling any task, activity, or process in most industries. A Big Data ecosystem of infrastructures, technologies, and tools is already available and ready to tackle complex problems. Scientific programming techniques are being disrupted for this *mare magnum* of

technology. Complexity is not just the main driver to compute large and complex tasks. Large amounts of data are now used as input of complex algorithms, techniques, and methods to generate again huge amounts of more data items (e.g., simulation processes). These data-intensive environments strongly affect the current engineering discipline and methods that must be reshaped to take advantage of more and smarter data and techniques to improve both the development and operation of complex and safety-critical systems. That is why, the provision of new engineering methods through the exploitation of existing resources (infrastructure) and combination of well-known techniques will enable the industry to build complex systems faster and safer. However, there is also an implicit need to properly manage and harmonize all the aspects concerning a data-driven engineering environment. New integration platforms (and standardized architectural frameworks) must be designed to cover all stages of the data life cycle encouraging people to run large-scale experiments and improve the current practice in systems engineering.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The current work has been partially supported by the Research Agreement between the RTVE (the Spanish Radio and Television Corporation) and the UC3M to boost research in the field of Big Data, Linked Data, Complex Network Analysis, and Natural Language. It has also received the support of the Tecnologico Nacional de Mexico (TECNM), National Council of Science and Technology (CONACYT), and the Public Education Secretary (SEP) through PRODEP.

References

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] N. Jazdi, "Cyber physical systems in the context of Industry 4.0," in *Proceedings of 2014 IEEE International Conference on Automation Quality and Testing, Robotics*, pp. 1–4, IEEE, Cluj-Napoca, Romania, May 2014.

[3] M. Borsato and M. Peruzzini, "Collaborative engineering," in *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, J. Stjepandić, N. Wognum, and W. J. C. Verhagen, Eds., pp. 165–196, Springer International Publishing, Cham, Switzerland, 2015.

[4] W. Shen, Q. Hao, and W. Li, "Computer supported collaborative design: retrospective and perspective," *Computers in Industry*, vol. 59, no. 9, pp. 855–862, 2008.

[5] J. Lee, H.-A. Kao, and S. Yang, "Service innovation and smart analytics for industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.

[6] INCOSE, *Systems Engineering Vision 2020, INCOSE*, Technical INCOSE-TP-2004-004-02, 2004.

[7] G. Wilson, D. A. Aruliah, C. Titus Brown et al., "Best practices for scientific computing," *PLoS Biology*, vol. 12, no. 1, article e1001745, 2014.

[8] P. Prabhu, T. B. Jablin, A. Raman et al., "A survey of the practice of computational science," in *State of the Practice Reports on-SC 11*, p. 19, ACM, New York, NY, USA, 2011.

[9] J. E. Hannay, C. MacLeod, J. Singer, H. P. Langtangen, D. Pfahl, and G. Wilson, "How do scientists develop and use scientific software?," in *Proceedings of 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, pp. 1–8, IEEE Computer Society, Montréal, Canada, 2009.

[10] N. Khan, I. Yaqoob, I. A. Targio Hashem et al., "Big Data: survey, technologies, opportunities, and challenges," *Scientific World Journal*, vol. 2014, pp. 1–18, 2014.

[11] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya, "Big Data computing and clouds: trends and future directions," *Journal of Parallel and Distributed Computing*, vol. 79, pp. 3–15, 2015.

[12] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: a survey on Big Data," *Information Sciences*, vol. 275, pp. 314–347, 2014.

[13] S. Gole and B. Tidke, "A survey of big data in social media using data mining techniques," in *Proceedings of 2015 International Conference on Advanced Computing and Communication Systems*, pp. 1–6, Coimbatore, India, 2015.

[14] C. S. Greene, J. Tan, M. Ung, J. H. Moore, and C. Cheng, "Big data bioinformatics," *Journal of Cellular Physiology*, vol. 229, no. 12, pp. 1896–1900, 2014.

[15] G. Boulton, "The challenges of a big data earth," *Big Earth Data*, vol. 2, no. 1, pp. 1–7, 2018.

[16] G.-H. Kim, S. Trimi, and J.-H. Chung, "Big-data applications in the government sector," *Communications of the ACM*, vol. 57, no. 3, pp. 78–85, 2014.

[17] C. S. Kruse, R. Goswamy, Y. Raval, and S. Marawi, "Adoption factors of the electronic health record: a systematic review," *JMIR Medical Informatics*, vol. 4, no. 4, p. e38, 2016.

[18] S. Hamrioui, I. de la Torre Díez, B. Garcia-Zapirain, K. Saleem, and J. J. P. C. Rodrigues, "A systematic review of security mechanisms for big data in health and new alternatives for hospitals," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–6, 2017.

[19] R. Kitchin, "The real-time city? Big data and smart urbanism," *GeoJournal*, vol. 79, no. 1, pp. 1–14, 2014.

[20] S. Yin and O. Kaynak, "Big data for modern industry: challenges and trends [point of view]," *Proceedings of the IEEE*, vol. 103, no. 2, pp. 143–146, 2015.

[21] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1–5, 2013.

[22] F. Gessert, W. Wingerath, S. Friedrich, and N. Ritter, "NoSQL database systems: a survey and decision guidance," *Computer Science-Research and Development*, vol. 32, no. 3–4, pp. 353–365, 2017.

[23] S. Gilbert and N. Lynch, "Perspectives on the CAP theorem," *Computer*, vol. 45, no. 2, pp. 30–36, 2012.

[24] C.-H. Chen, C.-L. Hsu, and K.-Y. Tsai, "Survey on open source frameworks for big data analytics," in *Proceedings of Third International Conference on Electronics and Software Science*, , Takamatsu, Japan, 2017.

[25] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[26] A. Ching, S. Edunov, M. Kabiljo, D. Logothetis, and S. Muthukrishnan, "One trillion edges: graph processing at Facebook-scale," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1804–1815, 2015.

[27] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of 'big data' on cloud computing: review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[28] D. Agrawal, S. Das, and A. El Abbadi, "Big data and cloud computing: current state and future opportunities," in *Proceedings of the 14th International Conference on Extending Database Technology*, pp. 530–533, 2011.

[29] V. Holmes and M. Newall, "HPC and the BIG Data challenge," *Safety and Reliability*, vol. 36, no. 3, pp. 213–224, 2016.

[30] I. Corporation, *Big Data Meets High Performance Computing*, Intel HPC, 2014, https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/big-data-meets-high-performance-computing-white-paper.pdf.

[31] B. Kitchenham, *Procedures for Performing Systematic Reviews*, Vol. 33, Keele University, Keele, UK, 2004.

[32] S. Bhattacharyya and D. Ivanova, "Scientific Computing and Big Data Analytics: Application in Climate Science," in *Distributed Computing in Big Data Analytics*, S. Mazumder, R. Singh Bhadoria, and G. C. Deka, Eds., pp. 95–106, Springer International Publishing, Cham, Switzerland, 2017.

[33] D. Budgen and P. Brereton, "Performing systematic literature reviews in software engineering," in *Proceedings of the 28th International Conference on Software Engineering*, pp. 1051-1052, Shanghai, China, 2006.

[34] J. M. Álvarez-Rodríguez, J. E. Labra-Gayo, and P. O. de Pablos, "New trends on e-Procurement applying semantic technologies: Current status and future challenges," *Computers in Industry*, vol. 65, no. 5, pp. 800–820, 2014.

[35] L. Shamseer et al., "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015: elaboration and explanation," *BMJ*, vol. 349, p. g7647, 2015.

[36] NIST Big Data Public Working Group Technology Roadmap Subgroup, *NIST Big Data Interoperability Framework: Volume 7*, Standards Roadmap, National Institute of Standards and Technology, NIST SP 1500-7, 2015.

[37] Z. Halim, M. Waqas, A. R. Baig, and A. Rashid, "Efficient clustering of large uncertain graphs using neighborhood information," *International Journal of Approximate Reasoning*, vol. 90, pp. 274–291, 2017.

[38] T. Kajdanowicz, R. Michalski, K. Musial, and P. Kazienko, "Learning in unlabeled networks – An active learning and inference approach," *AI Communications*, vol. 29, no. 1, pp. 123–148, 2015.

[39] P. Kazienko, R. Alhajj, and J. Srivastava, "Computational aspects of social network analysis," *Scientific Programming*, vol. 2015, pp. 1-2, 2015.

[40] E. I. M. Zayid and M. F. Akay, "Predicting the performance measures of a message-passing multiprocessor architecture using artificial neural networks," *Neural Computing and Applications*, vol. 23, no. 7–8, pp. 2481–2491, 2013.

[41] Y. Ma, L. Chen, P. Liu, and K. Lu, "Parallel programing templates for remote sensing image processing on GPU architectures: design and implementation," *Computing*, vol. 98, no. 1-2, pp. 7–33, 2016.

[42] T. S. Sliwinski and S.-L. Kang, "Applying parallel computing techniques to analyze terabyte atmospheric boundary layer model outputs," *Big Data Research*, vol. 7, pp. 31–41, 2017.

[43] M. M. Rathore, H. Son, A. Ahmad, A. Paul, and G. Jeon, "Real-time big data stream processing using GPU with spark over Hadoop ecosystem," *International Journal of Parallel Programming*, vol. 46, no. 3, pp. 630–646, 2017.

[44] W. Xue, C. Yang, H. Fu et al., "Ultra-scalable CPU-MIC acceleration of mesoscale atmospheric modeling on Tianhe-2," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2382–2393, 2015.

[45] Y. Aksari and H. Artuner, "Forward and back substitution algorithms on GPU: a case study on modified incomplete Cholesky Preconditioner for three-dimensional finite difference method," *Journal of Supercomputing*, vol. 62, no. 1, pp. 550–572, 2012.

[46] A. Neic, M. Liebmann, E. Hoetzl et al., "Accelerating Cardiac Bidomain Simulations Using Graphics Processing Units," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 8, pp. 2281–2290, 2012.

[47] E. Thompson, N. Clem, D. A. Peter, J. Bryan, B. I. Peterson, and D. Holbrook, "Parallel cuda implementation of conflict detection for application to airspace deconfliction," *Journal of Supercomputing*, vol. 71, no. 10, pp. 3787–3810, 2015.

[48] L. Jian, C. Wang, Y. Liu, S. Liang, W. Yi, and Y. Shi, "Parallel data mining techniques on Graphics Processing Unit with Compute Unified Device Architecture (CUDA)," *Journal of Supercomputing*, vol. 64, no. 3, pp. 942–967, 2013.

[49] W. Zhou, Z. Cai, B. Lian, J. Wang, and J. Ma, "Protein database search of hybrid alignment algorithm based on GPU parallel acceleration," *Journal of Supercomputing*, vol. 73, no. 10, pp. 4517–4534, 2017.

[50] P. Leite, J. M. Teixeira, T. Farias, B. Reis, V. Teichrieb, and J. Kelner, "Nearest neighbor searches on the GPU: a massively parallel approach for dynamic point clouds," *International Journal of Parallel Programming*, vol. 40, no. 3, pp. 313–330, 2012.

[51] P. Liu, A. Hemani, K. Paul, C. Weis, M. Jung, and N. Wehn, "3D-stacked many-core architecture for biological sequence analysis problems," *International Journal of Parallel Programming*, vol. 45, no. 6, pp. 1420–1460, 2017.

[52] T. A. Wassenaar, M. van Dijk, N. Loureiro-Ferreira et al., "WeNMR: structural biology on the grid," *Journal of Grid Computing*, vol. 10, no. 4, pp. 743–767, 2012.

[53] S. Venkateshan, A. Patel, and K. Varghese, "Hybrid working set algorithm for SVM learning with a kernel coprocessor on FPGA," *IIEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 10, pp. 2221–2232, 2015.

[54] J. K. Debnath, A. M. Gole, and W.-K. Fung, "Graphics-processing-unit-based acceleration of electromagnetic transients simulation," *IEEE Transactions on Power Delivery*, vol. 31, no. 5, pp. 2036–2044, 2016.

[55] A. Munir, F. Koushanfar, A. Gordon-Ross, and S. Ranka, "High-performance optimizations on tiled many-core embedded systems: a matrix multiplication case study," *Journal of Supercomputing*, vol. 66, no. 1, pp. 431–487, 2013.

[56] L. Ziane Khodja, R. Couturier, A. Giersch, and J. M. Bahi, "Parallel sparse linear solver with GMRES method using minimization techniques of communications for GPU clusters," *Journal of Supercomputing*, vol. 69, no. 1, pp. 200–224, 2014.

[57] H. Ltaief, P. Luszczek, and J. Dongarra, "Profiling high performance dense linear algebra algorithms on multicore architectures for power and energy efficiency," *Computer Science-Research and Development*, vol. 27, no. 4, pp. 277–287, 2012.

[58] M. A. Al-Mouhamed and A. H. Khan, "SpMV and BiCG-Stab optimization for a class of hepta-diagonal-sparse matrices on GPU," *Journal of Supercomputing*, vol. 73, no. 9, pp. 3761–3795, 2017.

[59] S. Li, C. Hu, J. Zhang, and Y. Zhang, "Automatic tuning of sparse matrix-vector multiplication on multicore clusters," *Science China Information Sciences*, vol. 58, no. 9, pp. 1–14, 2015.

[60] G. Zhou and H. Chen, "Parallel cube computation on modern CPUs and GPUs," *Journal of Supercomputing*, vol. 61, no. 3, pp. 394–417, 2012.

[61] A. Digital, "People first: the primacy of people in a digital age," *Accenture Technology Vision*, vol. 2016, 2016.

[62] A. Mansoori, S. Effati, and M. Eshaghnezhad, "An efficient recurrent neural network model for solving fuzzy non-linear programming problems," *Applied Intelligence*, vol. 46, no. 2, pp. 308–327, 2017.

[63] T. Muhammad and Z. Halim, "Employing artificial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique," *Applied Soft Computing*, vol. 49, pp. 365–384, 2016.

[64] Z. Halim, R. Kalsoom, S. Bashir, and G. Abbas, "Artificial intelligence techniques for driving safety and vehicle crash prediction," *Artificial Intelligence Review*, vol. 46, no. 3, pp. 351–387, 2016.

[65] J. Quadflieg, M. Preuss, and G. Rudolph, "Driving as a human: a track learning based adaptable architecture for a car racing controller," *Genetic Programming and Evolvable Machines*, vol. 15, no. 4, pp. 433–476, 2014.

[66] S. Terzi and S. Serin, "Planning maintenance works on pavements through ant colony optimization," *Neural Computing and Applications*, vol. 25, no. 1, pp. 143–153, 2014.

[67] A. Anton and A. Aldea, "Pumping stations: solving algorithm with inverse functions," *Procedia Engineering*, vol. 70, pp. 67–74, 2014.

[68] D. G. Savakar and A. Kannur, "A practical aspect of identification and classifying of Guns based on gunshot wound patterns using gene expression programming," *Pattern Recognition and Image Analysis*, vol. 26, no. 2, pp. 442–449, 2016.

[69] S. Aras and I. D. Kocakoç, "A new model selection strategy in time series forecasting with artificial neural networks: IHTS," *Neurocomputing*, vol. 174, pp. 974–987, 2016.

[70] Y. He, S. Gao, N. Liao, and H. Liu, "A nonlinear goal-programming-based DE and ANN approach to grade optimization in iron mining," *Neural Computing and Applications*, vol. 27, no. 7, pp. 2065–2081, 2016.

[71] L. Petnga and M. Austin, "Ontologies of time and time-based reasoning for MBSE of cyber-physical systems," *Procedia Computer Science*, vol. 16, pp. 403–412, 2013.

[72] G. Hongbo, X. Guotao, Z. Xinyu, and C. Bo, "Autonomous parking control for intelligent vehicles based on a novel algorithm," *Journal of China Universities of Posts and Telecommunications*, vol. 24, no. 4, pp. 51–56, 2017.

[73] S. Bingöl and H. Y. Kılıçgedik, "Application of gene expression programming in hot metal forming for intelligent manufacturing," *Neural Computing and Applications*, vol. 30, no. 3, pp. 937–945, 2016.

[74] L. Dioşan and A. Andreica, "Multi-objective breast cancer classification by using multi-expression programming," *Applied Intelligence*, vol. 43, no. 3, pp. 499–511, 2015.

[75] K. Charalampous and A. Gasteratos, "On-line deep learning method for action recognition," *Pattern Analysis and Applications*, vol. 19, no. 2, pp. 337–354, 2016.

[76] G. Ososkov and P. Goncharov, "Shallow and deep learning for image classification," *Optical Memory and Neural Networks*, vol. 26, no. 4, pp. 221–248, 2017.

[77] H. Karami, S. Karimi, H. Bonakdari, and S. Shamshirband, "Predicting discharge coefficient of triangular labyrinth weir using extreme learning machine, artificial neural network and genetic programming," *Neural Computing and Applications*, vol. 29, no. 11, pp. 983–989, 2016.

[78] V. Havlíček, M. Hanel, P. Máca, M. Kuráž, and P. Pech, "Incorporating basic hydrological concepts into genetic programming for rainfall-runoff forecasting," *Computing*, vol. 95, no. S1, pp. 363–380, 2013.

[79] A. Alexandridis, E. Chondrodima, E. Efthimiou, G. Papadakis, F. Vallianatos, and D. Triantis, "Large earthquake occurrence estimation based on radial basis function neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 9, pp. 5443–5453, 2014.

[80] J. Li, C. Li, Z. Wu, and J. Huang, "A feedback neural network for solving convex quadratic bi-level programming problems," *Neural Computing and Applications*, vol. 25, no. 3–4, pp. 603–611, 2014.

[81] M. Mulas and P. Massobrio, "NeuVision: A novel simulation environment to model spontaneous and stimulus-evoked activity of large-scale neuronal networks," *Neurocomputing*, vol. 122, pp. 441–457, 2013.

[82] K. J. Turner and P. S. Lambert, "Workflows for quantitative data analysis in the social sciences," *International Journal on Software Tools for Technology Transfer*, vol. 17, no. 3, pp. 321–338, 2015.

[83] S. Chakraborty, A. Cortesi, and N. Chaki, "A uniform representation of multi-variant data in intensive-query databases," *Innovations in Systems and Software Engineering*, vol. 12, no. 3, pp. 163–176, 2016.

[84] S. Multsch, D. Grabowski, J. Lüdering et al., "A practical planning software program for desalination in agriculture -SPARE:WATERopt," *Desalination*, vol. 404, pp. 121–131, 2017.

[85] K. Zafar, R. Baig, N. Bukhari, and Z. Halim, "Route planning and optimization of route using simulated ant agent system," *Journal of Circuits, Systems and Computers*, vol. 20, no. 03, pp. 457–478, 2011.

[86] A. Alexandrov, R. Bergmann, S. Ewen et al., "The Stratosphere platform for big data analytics," *VLDB Journal*, vol. 23, no. 6, pp. 939–964, 2014.

[87] F. Nadeem, D. Alghazzawi, A. Mashat, K. Fakeeh, A. Almalaise, and H. Hagras, "Modeling and predicting execution time of scientific workflows in the Grid using radial basis function neural network," *Cluster Computing*, vol. 20, no. 3, pp. 2805–2819, 2017.

[88] S. J. van Zelst, B. F. van Dongen, W. M. P. van der Aalst, and H. M. W. Verbeek, "Discovering workflow nets using integer linear programming," *Computing*, vol. 100, no. 5, pp. 529–556, 2017.

[89] G. Li, D. Wu, J. Hu, Y. Li, M. S. Hossain, and A. Ghoneim, "HELOS: heterogeneous load scheduling for electric vehicle-integrated microgrids," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5785–5796, 2017.

[90] D. de Oliveira, K. A. C. S. Ocaña, F. Baião, and M. Mattoso, "A provenance-based adaptive scheduling heuristic for parallel scientific workflows in clouds," *Journal of Grid Computing*, vol. 10, no. 3, pp. 521–552, 2012.

[91] I. K. Musa, S. D. Walker, A. M. Owen, and A. P. Harrison, "Self-service infrastructure container for data intensive application," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 3, no. 1, p. 5, 2014.

[92] M. Atkinson, C. S. Liew, M. Galea et al., "Data-intensive architecture for scientific knowledge discovery," *Distributed and Parallel Databases*, vol. 30, no. 5–6, pp. 307–324, 2012.

[93] M. S. Mayernik, J. C. Wallis, and C. L. Borgman, "Unearthing the infrastructure: humans and sensors in field-based scientific research," *Computer Supported Cooperative Work (CSCW)*, vol. 22, no. 1, pp. 65–101, 2013.

[94] J. Zhang, J. Yan, Y. Ma, D. Xu, P. Li, and W. Jie, "Infrastructures and services for remote sensing data production management across multiple satellite data centers," *Cluster Computing*, vol. 19, no. 3, pp. 1243–1260, 2016.

[95] Q. Wu, M. Zhu, Y. Gu et al., "A distributed workflow management system with case study of real-life scientific applications on grids," *Journal of Grid Computing*, vol. 10, no. 3, pp. 367–393, 2012.

[96] A. Pellegrini, S. Peluso, F. Quaglia, and R. Vitali, "Transparent speculative parallelization of discrete event simulation applications using global variables," *International Journal of Parallel Programming*, vol. 44, no. 6, pp. 1200–1247, 2016.

[97] C. Lively, X. Wu, V. Taylor et al., "Power-aware predictive models of hybrid (MPI/OpenMP) scientific applications on multicore systems," *Computer Science-Research and Development*, vol. 27, no. 4, pp. 245–253, 2012.

[98] Y. Eom, J. Kim, and B. Nam, "Multi-dimensional multiple query scheduling with distributed semantic caching framework," *Cluster Computing*, vol. 18, no. 3, pp. 1141–1156, 2015.

[99] M. Z. A. Bhuiyan, J. Wu, G. Wang, T. Wang, and M. M. Hassan, "e-Sampling: event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 12, no. 1, pp. 1–29, 2017.

[100] Z. Milosevic, W. Chen, A. Berry, and F. A. Rabhi, "An open architecture for event-based analytics," *International Journal of Data Science and Analytics*, vol. 2, no. 1–2, pp. 13–27, 2016.

[101] S.-W. Lee, "Evidence-driven decision support in critical infrastructure management through enhanced domain knowledge modeling," *Multimedia Tools and Applications*, vol. 71, no. 1, pp. 309–330, 2014.

[102] R. Jacob, J. Krishna, X. Xu et al., "ParNCL and ParGAL: data-parallel tools for postprocessing of large-scale earth science data," *Procedia Computer Science*, vol. 18, pp. 1245–1254, 2013.

[103] D. Amagata and T. Hara, "A general framework for MaxRS and MaxCRS monitoring in spatial data streams," *ACM Transactions on Spatial Algorithms and Systems*, vol. 3, no. 1, pp. 1–34, 2017.

[104] J. Weinbub, K. Rupp, and S. Selberherr, "ViennaX: a parallel plugin execution framework for scientific computing," *Engineering with Computers*, vol. 30, no. 4, pp. 651–668, 2014.

[105] G. Zuo, G. Guan, and R. Wang, "Numerical modeling and optimization of vacuum membrane distillation module for low-cost water production," *Desalination*, vol. 339, pp. 1–9, 2014.

[106] M. D. G. Garcia-Hernandez, J. Ruiz-Pinales, E. Onaindia et al., "New prioritized value iteration for Markov decision processes," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 157–167, 2012.

[107] G. Simonin, C. Artigues, E. Hebrard, and P. Lopez, "Scheduling scientific experiments for comet exploration," *Constraints*, vol. 20, no. 1, pp. 77–99, 2015.

[108] W. Horn, M. Kumar, J. Jann et al., "Graph programming interface (GPI): a linear algebra programming model for large scale graph computations," *International Journal of Parallel Programming*, vol. 46, no. 2, pp. 412–440, 2017.

[109] S. Lai, G. Lai, F. Lu, G. Shen, J. Jin, and X. Lin, "A BSP model graph processing system on many cores," *Cluster Computing*, vol. 20, no. 2, pp. 1359–1377, 2017.

[110] V. Boulos, S. Huet, V. Fristot, L. Salvo, and D. Houzet, "Efficient implementation of data flow graphs on multi-gpu clusters," *Journal of Real-Time Image Processing*, vol. 9, no. 1, pp. 217–232, 2014.

[111] J. Dümmler, R. Kunis, and G. Rünger, "SEParAT: scheduling support environment for parallel application task graphs," *Cluster Computing*, vol. 15, no. 3, pp. 223–238, 2012.

[112] Q. D. Pham, Y. Deville, and P. Van Hentenryck, "LS(Graph): a constraint-based local search for constraint optimization on trees and paths," *Constraints*, vol. 17, no. 4, pp. 357–408, 2012.

[113] W. Ju, J. Li, W. Yu, and R. Zhang, "iGraph: an incremental data processing system for dynamic graph," *Frontiers of Computer Science*, vol. 10, no. 3, pp. 462–476, 2016.

[114] O. Çeliktutan, C. Wolf, B. Sankur, and E. Lombardi, "Fast exact hyper-graph matching with dynamic programming for spatio-temporal data," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 1–21, 2015.

[115] J.-H. Jung, J. Lee, K. R. Hoffmann, T. Dorazio, and E. B. Pitman, "A rapid interpolation method of finding vascular CFD solutions with spectral collocation methods," *Journal of Computational Science*, vol. 4, no. 1–2, pp. 101–110, 2013.

[116] J. Xu, J. Han, K. Xiong, and F. Nie, "Robust and sparse fuzzy K-means clustering," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2224–2230, New York, NY, USA, 2016.

[117] F. Nie, C. Ding, D. Luo, and H. Huang, "Improved MinMax cut graph clustering with nonnegative relaxation," in *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, pp. 451–466, Berlin, Heidelberg, 2010.

[118] H. Motallebi and S. Parsa, "Data locality optimization of interference graphs based on polyhedral computations," *Journal of Supercomputing*, vol. 61, no. 3, pp. 935–965, 2012.

[119] Z. Halim, M. Atif, A. Rashid, and C. A. Edwin, "Profiling players using real-world datasets: clustering the data and correlating the results with the big-five personality traits," *IEEE Transactions on Affective Computing*, p. 1, 2017.

[120] A. Weiß and D. Karastoyanova, "Enabling coupled multi-scale, multi-field experiments through choreographies of data-driven scientific simulations," *Computing*, vol. 98, no. 4, pp. 439–467, 2016.

[121] S. P. Muszala, G. Alaghband, J. Hack, and D. Connors, "Natural Load Indices (NLI) for scientific simulation," *Journal of Supercomputing*, vol. 59, no. 1, pp. 392–413, 2012.

[122] R. R. Upadhyay and O. A. Ezekoye, "libMoM : a library for stochastic simulations in engineering using statistical moments," *Engineering with Computers*, vol. 28, no. 1, pp. 83–94, 2012.

[123] B. Ben Youssef, "A parallel cellular automata algorithm for the deterministic simulation of 3-D multicellular tissue growth," *Cluster Computing*, vol. 18, no. 4, pp. 1561–1579, 2015.

[124] H. Mohamed and S. Marchand-Maillet, "Distributed media indexing based on MPI and MapReduce," *Multimedia Tools and Applications*, vol. 69, no. 2, pp. 513–537, 2014.

[125] W. Wang and G. Zeng, "Bayesian Cognitive Model in scheduling algorithm for data intensive computing," *Journal of Grid Computing*, vol. 10, no. 1, pp. 173–184, 2012.

[126] C. Ling, T. Hamada, J. Gao, G. Zhao, D. Sun, and W. Shi, "MrBayes tgMC ^{3} ++: a high performance and resource-efficient GPU-oriented phylogenetic analysis method," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 5, pp. 845–854, 2016.

[127] P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, and J. L. Ayala, "Enhancing regression models for complex systems using evolutionary techniques for feature engineering," *Journal of Grid Computing*, vol. 13, no. 3, pp. 409–423, 2015.

[128] G. Zhang, H. Pu, W. He, F. Liu, J. Luo, and J. Bai, "Bayesian framework based direct reconstruction of fluorescence parametric images," *IEEE Transactions on Medical Imaging*, vol. 34, no. 6, pp. 1378–1391, 2015.

[129] L. Peng, B. Liao, W. Zhu, Z. Li, and K. Li, "Predicting drug–target interactions with multi-information fusion," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 2, pp. 561–572, 2017.

[130] E. Park, J. Cavazos, L.-N. Pouchet, C. Bastoul, A. Cohen, and P. Sadayappan, "Predictive modeling in a polyhedral optimization space," *International Journal of Parallel Programming*, vol. 41, no. 5, pp. 704–750, 2013.

[131] M. Kreutzer, J. Thies, M. Röhrig-Zöllner et al., "GHOST: building blocks for high performance sparse linear algebra on heterogeneous systems," *International Journal of Parallel Programming*, vol. 45, no. 5, pp. 1046–1072, 2017.

[132] A. B. Manic, A. P. Smull, F.-H. Rouet, X. S. Li, and B. M. Notaros, "Efficient scalable parallel higher order direct MoM-SIE method with hierarchically semiseparable structures for 3-D scattering," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 5, pp. 2467–2478, 2017.

[133] K. V. Ryabinin and S. I. Chuprina, "A unified approach to adapt scientific visualization systems to third-party solvers," *Programming and Computer Software*, vol. 42, no. 6, pp. 347–355, 2016.

[134] X. Deng, J. Lee, and Robby, "Efficient and formal generalized symbolic execution," *Automated Software Engineering*, vol. 19, no. 3, pp. 233–301, 2012.

[135] Y. Chen, D. Chen, S. U. Khan, J. Huang, and C. Xie, "Solving symbolic regression problems with uniform design-aided gene expression programming," *Journal of Supercomputing*, vol. 66, no. 3, pp. 1553–1575, 2013.

[136] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, "Automation and combination of linear-programming based stabilization techniques in column generation," *INFORMS Journal on Computing*, vol. 30, no. 2, pp. 339–360, 2018.

[137] P. Chobeau, G. Guillaume, J. Picaut, D. Ecotière, and G. Dutilleux, "A Transmission Line Matrix model for sound propagation in arrays of cylinders normal to an impedance plane," *Journal of Sound and Vibration*, vol. 389, pp. 454–467, 2017.

[138] S. Deng, T. Meng, and Z. Jin, "Nonlinear programming control using differential aerodynamic drag for CubeSat formation flying," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 7, pp. 867–881, 2017.

[139] M. Frego, E. Bertolazzi, F. Biral, D. Fontanelli, and L. Palopoli, "Semi-analytical minimum time solutions with velocity constraints for trajectory following of vehicles," *Automatica*, vol. 86, pp. 18–28, 2017.

[140] S. Dey, T. Mukhopadhyay, A. Spickenheuer, S. Adhikari, and G. Heinrich, "Bottom up surrogate based approach for stochastic frequency response analysis of laminated composite plates," *Composite Structures*, vol. 140, pp. 712–727, 2016.

[141] H. Astsatryan, V. Sahakyan, Y. Shoukouryan et al., "On the easy use of scientific computing services for large scale linear algebra and parallel decision making with the P-grade portal," *Journal of Grid Computing*, vol. 11, no. 2, pp. 239–248, 2013.

[142] A. Umbarkar, V. Subramanian, and A. Doboli, "Linear programming-based optimization for robust data modeling in a distributed sensing platform," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 10, pp. 1531–1544, 2014.

[143] H. M. Aktulga, M. Afibuzzaman, S. Williams et al., "A high performance block eigensolver for nuclear configuration interaction calculations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 6, pp. 1550–1563, 2017.

[144] F. Kizel, M. Shoshany, N. S. Netanyahu, G. Even-Tzur, and J. A. Benediktsson, "A stepwise analytical projected gradient descent search for hyperspectral unmixing and its code vectorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 4925–4943, 2017.

[145] A. Demiriz, N. Bagherzadeh, and A. Alhussein, "Using constraint programming for the design of network-on-chip architectures," *Computing*, vol. 97, no. 6, pp. 579–592, 2015.

[146] Á. Horváth and D. Varró, "Dynamic constraint satisfaction problems over models," *Software & Systems Modeling*, vol. 11, no. 3, pp. 385–408, 2012.

[147] W.-M. Ching and D. Zheng, "Automatic parallelization of array-oriented programs for a multi-core machine," *International Journal of Parallel Programming*, vol. 40, no. 5, pp. 514–531, 2012.

[148] D. Carvalho, L. R. de Souza, R. G. Barbastefano, and F. M. G. França, "Stochastic product-mix: a grid computing industrial application," *Journal of Grid Computing*, vol. 13, no. 2, pp. 293–304, 2015.

[149] A. N. Johanson and W. Hasselbring, "Effectiveness and efficiency of a domain-specific language for high-performance marine ecosystem simulation: a controlled experiment," *Empirical Software Engineering*, vol. 22, no. 4, pp. 2206–2236, 2017.

[150] A. Aguilera, R. Grunzke, D. Habich et al., "Advancing a gateway infrastructure for wind turbine data analysis," *Journal of Grid Computing*, vol. 14, no. 4, pp. 499–514, 2016.

[151] L. Linzer, L. Mhamdi, and T. Schumacher, "Application of a moment tensor inversion code developed for mining-induced seismicity to fracture monitoring of civil engineering materials," *Journal of Applied Geophysics*, vol. 112, pp. 256–267, 2015.

[152] S. Zhu, R. Gao, and Z. Li, "Stereo matching algorithm with guided filter and modified dynamic programming," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 199–216, 2017.

[153] T. Muhammad, Z. Halim, and M. A. Khan, "Visualizing trace of Java collection APIs by dynamic bytecode instrumentation," *Journal of Visual Languages & Computing*, vol. 43, pp. 14–29, 2017.

[154] A. Kern, C. Schelthoff, and M. Mathieu, "Probability of lightning strikes to air-terminations of structures using the electro-geometrical model theory and the statistics of lightning current parameters," *Atmospheric Research*, vol. 117, pp. 2–11, 2012.

[155] H. Hakula and T. Tuominen, "Mathematica implementation of the high order finite element method applied to eigenproblems," *Computing*, vol. 95, no. S1, pp. 277–301, 2013.

[156] R. Han, L. K. John, and J. Zhan, "Benchmarking big data systems: a review," *IEEE Transactions on Services Computing*, vol. 11, no. 3, pp. 580–597, 2018.

[157] T. Ivanov, "Big data benchmark compendium," in *Performance Evaluation and Benchmarking: Traditional to Big Data to Internet of Things*, pp. 135–155, Cham, Switzerland, 2016.

[158] C. Baru, M. Bhandarkar, R. Nambiar, M. Poess, and T. Rabl, "Setting the direction for big data benchmark standards," in *Selected Topics in Performance Evaluation and Benchmarking*, pp. 197–208, Berlin, Heidelberg, 2013.

[159] L. Wang, "BigDataBench: a big data benchmark suite from Internet services," in *Proceedings of 2014 IEEE 20th International Symposium on High Performance Computer Architecture*, pp. 488–499, HPCA, Orlando, FL, USA, 2014.

[160] M. Kunjir, P. Kalmegh, and S. Babu, "Thoth: towards managing a multi-system cluster," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1689–1692, 2014.

[161] C. Luo, J. Zhan, Z. Jia et al., "Cloudrank-D: benchmarking and ranking cloud computing systems for data processing applications," *Frontiers of Computer Science*, vol. 6, no. 4, pp. 347–362, 2012.

[162] A. Pavlo, E. Paulson, A. Rasin et al., "A comparison of approaches to large-scale data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pp. 165–178, 2009.

[163] L. Cao, "Data science: nature and pitfalls," *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 66–75, 2016.

[164] L. Cao, "Data science: challenges and directions," *Communications of the ACM*, vol. 60, no. 8, pp. 59–68, 2017.

[165] J. M. Alvarez-Rodríguez, J. Llorens, M. Alejandres, and J. M. Fuentes, "OSLC-KM: a knowledge management specification for OSLC-based resources," *INCOSE International Symposium*, vol. 25, no. 1, pp. 16–34, 2015.

[166] N. Attoh-Okine, "Big data challenges in railway engineering," in *Proceedings of IEEE International Conference on Big Data*, pp. 7–9, 2014.

[167] A. M. Chandramohan, D. Mylaraswamy, B. Xu, and P. Dietrich, "Big data infrastructure for aviation data analytics," in *Proceedings of IEEE International Conference on Cloud Computing in Emerging Markets*, pp. 1–6, 2014.

[168] O. Kapliński, N. Košeleva, and G. Ropaitė, "Big data in civil engineering: a state-of-the-art survey," *Engineering Structures and Technologies*, vol. 8, no. 4, pp. 165–175, 2016.

[169] L. Xu, A. Song, and W. Zhang, "Scalable parallel algorithm of multiple-relaxation-time lattice boltzmann method with large eddy simulation on multi-GPUs," *Scientific Programming*, vol. 2018, pp. 1–12, 2018.

[170] S. Ullah, M. D. Awan, and M. Sikander Hayat Khiyal, "Big data in cloud computing: a resource management perspective," *Scientific Programming*, vol. 2018, pp. 1–17, 2018.

[171] I. Kholod, I. Petukhov, and A. Shorov, "Cloud for distributed data analysis based on the actor model," *Scientific Programming*, vol. 2016, pp. 1–11, 2016.

[172] B. R. Chang, Y.-D. Lee, and P.-H. Liao, "Development of multiple big data analytics platforms with rapid response," *Scientific Programming*, vol. 2017, pp. 1–13, 2017.

[173] W. Zhao, L. Gao, and A. Liu, "Programming foundations for scientific big data analytics," *Scientific Programming*, vol. 2018, pp. 1-2, 2018.

[174] L. Zhang and J. Gao, "Incremental graph pattern matching algorithm for big graph data," *Scientific Programming*, vol. 2018, pp. 1–8, 2018.

[175] X. Xia, Z. Chen, and W. Wei, "Research on monitoring and prewarning system of accident in the coal mine based on big data," *Scientific Programming*, vol. 2018, pp. 1–10, 2018.

[176] Z. Liu, Y. Jia, and X. Zhu, "Deployment strategy for car-sharing depots by clustering urban traffic big data based on affinity propagation," *Scientific Programming*, vol. 2018, pp. 1–9, 2018.

[177] H. Zhong and J. Xiao, "Enhancing health risk prediction with deep learning on big data and revised fusion node paradigm," *Scientific Programming*, vol. 2017, pp. 1–18, 2017.

[178] W. Aziguli, Y. Zhang, Y. Xie et al., "A robust text classifier based on denoising deep neural network in the analysis of big data," *Scientific Programming*, vol. 2017, pp. 1–10, 2017.

[179] A. AlShawi, "Applying data mining techniques to improve information security in the cloud: a single cache system approach," *Scientific Programming*, vol. 2016, pp. 1–5, 2016.

Hindawi
Scientific Programming
Volume 2018, Article ID 4304017, 21 pages
https://doi.org/10.1155/2018/4304017

Research Article

Analysis of Medical Opinions about the Nonrealization of Autopsies in a Mexican Hospital Using Association Rules and Bayesian Networks

Elayne Rubio Delgado,[1] Lisbeth Rodríguez-Mazahua,[1] José Antonio Palet Guzmán,[2] Jair Cervantes,[3] José Luis Sánchez Cervantes,[4] Silvestre Gustavo Peláez-Camarena,[1] and Asdrúbal López-Chau[5]

[1]*División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Orizaba, Orizaba, VER, Mexico*
[2]*Hospital Regional de Río Blanco (HRRB), Río Blanco, VER, Mexico*
[3]*Universidad Autónoma del Estado de México, Centro Universitario UAEM Texcoco, Texcoco, MEX, Mexico*
[4]*CONACYT-Instituto Tecnológico de Orizaba, Orizaba, VER, Mexico*
[5]*Universidad Autónoma del Estado de México, Centro Universitario UAEM Zumpango, Zumpango, MEX, Mexico*

Correspondence should be addressed to Lisbeth Rodríguez-Mazahua; lisbethr08@gmail.com

Received 6 May 2017; Revised 28 September 2017; Accepted 9 January 2018; Published 13 February 2018

Academic Editor: José María Álvarez-Rodríguez

This research identifies the factors influencing the reduction of autopsies in a hospital of Veracruz. The study is based on the application of data mining techniques such as association rules and Bayesian networks in data sets obtained from opinions of physicians. We analyzed, for the exploration and extraction of the knowledge, algorithms like Apriori, FPGrowth, PredictiveApriori, Tertius, J48, NaiveBayes, MultilayerPerceptron, and BayesNet, all of them provided by the API of WEKA. To generate mining models and present the new knowledge in natural language, we also developed a web application. The results presented in this study are those obtained from the best-evaluated algorithms, which have been validated by specialists in the field of pathology.

1. Introduction

Autopsy [1] is a very important practice for medicine. It is the only study that allows identifying the true cause of death of the deceased, studying the evolution of diseases, determining the effectiveness of traditional treatments, discovering new diseases, and much more. However, in a hospital of Veracruz State, this method is practically in disuse, which have motivated the pathology department to investigate the probable causes. Figure 1 shows how autopsy studies have declined in the hospital. In the years 2012 and 2016, none of them were performed. For this reason, in [2, 3], the Apriori algorithm was applied to several data sets obtained by the application of an instrument (survey) to doctors of the hospital, with the aim of finding interesting association rules to identify the main causes for the nonrealization of autopsies in the hospital. In this paper, we present a more complete analysis because we apply four association rule mining algorithms, that is, Apriori, FPGrowth, PredictiveApriori, and Tertius, as well as Bayesian network learning to the data sets in order to identify the factors which influence the reduction of autopsies in the hospital. Also, we evaluated different classification algorithms such as J48, NaiveBayes, MultilayerPerceptron, and Sequential Minimal Optimization (SMO) to determine which is the best to be applied in the classification of open questions of the survey.

The purpose of this study is providing a tool to the department of pathology that makes the analysis of the above-described situation easier and allows discarding or testing hypotheses about the origin of the problem, so that physicians

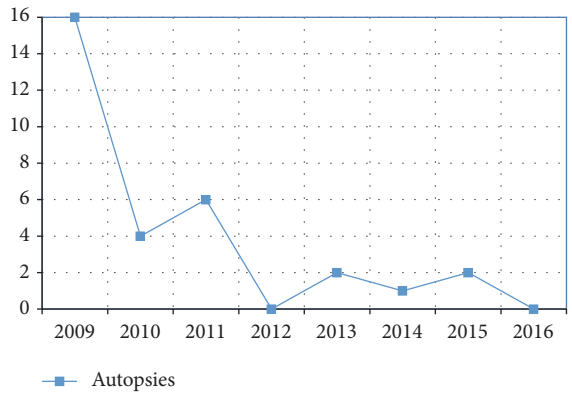


Figure 1: Behavior of autopsies in a hospital in Veracruz.

can base their solution proposals on intelligent, statistical, and probabilistic methods. Therefore, we developed a web application, which is capable of generating mining models, interpreting them, and returning the results in a natural language that pathologists can understand.

Prior to the development of the web application, we reviewed several research-related papers, which are described in Section 2. The proposals of solutions and results of these studies demonstrate the usefulness of data mining (DM) to solve problems such as the one presented, so we considered relevant for its solution to identify the factors influencing the reduction of autopsies, through the application of machine learning algorithms for DM tasks.

As a social research technique, we used the survey, a tool that allowed us to collect the opinion of physicians regarding the importance, requesting, and realization of autopsies in the hospital. For some years, autopsies have been subject of interest of some researchers as is evidenced in [4]. This demonstrates that the problem of the study is not trivial and that it is also affecting areas beyond this Mexican hospital, so it is necessary to do research like this one to help continuing the use of this practice in order to maintain quality in medicine.

The disuse of autopsies is an evident fact in many parts of the world, as mentioned in [5]. The paper stated that at health centers in Nigeria there are currently no autopsies and that the most frightening thing is that nothing is done about it. The author, in addition to exposing the real benefits of this practice and mentioning possible reasons that have generated this situation, cited a set of works that report data that make the decrease of autopsies in other places like United Kingdom (UK) evident, where currently autopsies are only performed to 10% of the deceased. In addition, the study ruled out the possibility that autopsy could be replaced by high technology techniques because no one has so far managed to mitigate errors in clinical diagnoses.

Another work that showed that autopsies are declining in many countries of the world is [6]. The authors analyzed the Cancer Registries of Zurich between 1980 and 2010 to see how the number of autopsies in cancer patients has changed. The investigation showed that the autopsy rate decreased from 60% to 7% in the analyzed period.

The rates of autopsies of different places at UK that were presented in [7] are alarming. This research analyzed data from autopsies conducted in 2013 and found that England has an average autopsy rate of 0.51%, Scotland of 2.13%, Wales of 0.65%, and Northern Ireland of 0.46%. As can be seen, most of the values are below 1%, which is why the authors proposed to carry out investigations that analyze the impact of this for patient safety, health, and medical education.

In [8], it was argued that the situation in which the autopsy is going through is serious. It refers to the alarming decline of the autopsy practice, explaining that in the fourth part of the UK National Health Service autopsies are no longer performed and that, among other areas, in Europe and the United States the number of autopsies has also reduced considerably. The main cause of this is that doctors are not requesting them. Therefore, the authors intend to raise awareness of the importance of this medical study in the area of health, as well as among politicians and the general public, all this with the goal of resuming autopsies like a routine procedure.

The research [9] showed the results of a study conducted at Punjab Medical College, Faisalabad, where a group of medical students was surveyed during the 2011 and 2012 academic years. The survey recorded a group of emotional reactions to this practice, but in general all the students recommended its use. This work points out the importance of emphasizing the maintenance of autopsies in medical education, because without this learning resource, future doctors will present problems when explaining procedures they have never seen.

For a better explanation of the objective of study, the rest of the article is structured as follows. Section 2 shows some of the works related to the objective of study of this research, that is, data mining techniques applied to the medical field. The method that was followed to obtain the results of this research is described in Section 3. The association rules and Bayesian networks obtained are presented in Section 4. Finally, the conclusions are given in Section 5.

2. Related Works

We can classify the related works into two groups according to the type of DM techniques applied to solve problems in the medical area. We first discuss the articles that are focused on the use of classification techniques, or a combination between several techniques, followed by the works done exclusively under association rules.

2.1. Classification Techniques. With the motivation of supporting the development of the health sector in smart cities, several advances and trends of data mining in this area were described in [10]. According to the authors, Neural Networks and decision trees are the data mining techniques most used in the predictive analysis.

In [11], support vector machines (SVMs) were used to classify features associated with the effects of Human Immunodeficiency Virus (HIV) on the brain during three different periods of the early clinical course: primary infection, 4–12 months postinfection (pi), and >12–24 months pi.

Moreover, the goal of [12] was to demonstrate the use of artificial intelligence-based methods such as Bayesian networks to open up opportunities for creation of new knowledge in management of chronic conditions. The research found links between asthma and renal failure, which demonstrated the usefulness of this method to discern clinically relevant and not always evident correlations.

Also, in [13] it was determined that the best predictor of hypertriglyceridemia, based on traditional indicators derived from anthropometric measurements, may differ according to gender and age. To identify suitable predictors among the group of measures, authors employed two widely used machine learning algorithms to solve classification problems: logistic regression and Naïve Bayes (NB) algorithm.

In order to achieve highly accurate, concise, and interpretable classification rules that facilitate the diagnosis of type 2 diabetes mellitus and medical decision-making, the combined use of the Recursive-Rule eXtraction (Re-RX) and J48graft algorithms was proposed in [14]. Also, using this combination, a new extraction algorithm was developed, which the authors recommended to be tested in other data sets to validate its accuracy.

Different schemas for the identification of class labels for a given data set were compared in [15], to show how the proposal of an Improved Global Feature Selection Scheme (IGFSS) is more efficient than the classic ones. Also, the author described the use of algorithms commonly employed for text classification as NB and SVM, in order to demonstrate the effectiveness of his proposal.

With the goal of predicting the causes of deaths related to the World Health Organization standard classification of diseases, in [16] automatic learning techniques were applied in forensic text reports. In turn, the authors performed a comparison of feature extraction approaches, feature set representation approaches, and text classifiers such as SVM, Random Forest (RF), and NB for the classification of forensic autopsy reports. The data set used was the result of 400 forensic autopsy reports from a hospital in Kuala Lumpur, Malaysia, comprising eight of the most common causes of death. The results of the decision models for SVM exceeded those of RF and NB.

Also the results of [17] were interesting because authors proposed a system of automatic classification (multiclass) to predict the causes of death from decision models of automatic classification of texts. The data analyzed were 2200 autopsy records for accidents at a Kuala Lumpur hospital. The researchers evaluated SVM, NB, K-nearest neighbor (KNN), decision tree (DT), and RF algorithms according to precision, recall, F-measure, and ROC (Receiver Operating Characteristic) area metrics, from the data mining tool WEKA (Waikato Environment for Knowledge Analysis). RF and J48 proved to be the best-evaluated decision models.

The development of efficient and robust schemes for classifying text is of great importance to business intelligence and other areas. For this reason, [18] performed an empirical analysis on statistical methods for the extraction of keywords using the ACM (Association for Computing Machinery) and Reuters-21578 document collections. Authors also described the predictive behavior of classification algorithms and joint learning methods when using keywords to represent scientific text documents, thus demonstrating that as the number of keywords increases, the predictive performance of the classifiers tends to increase too.

In many countries with poor medical care, most deaths occur in households. Unlike hospital deaths, home deaths do not have a standard to be validated as it is indicated by [19], where it is explained that for this reason previous studies have shown contradictory performance of automated methods compared to physician-based classification of causes of death (COD). Therefore, authors compared the NB, open-source Tariff Method (OTM), and InterVA-4 (a model for interpreting verbal autopsy data into COD) classifiers in three data sets comprising about 21000 records of child and adult deaths. The result of the NB classifier overcame the other classifiers, although it was evident that none of the current automated classifiers is capable of adequately performing the individual COD classification.

The problem addressed by [20] was that it is difficult for experts to determine the degree of disease either when they lack sufficient evidence for medical diagnosis or when they have too much evidence. For this reason, authors analyzed important research that deals with the application of automatic learning algorithms for data mining tasks aimed at supporting the diagnosis of heart disease, breast cancer, and diabetes. The purpose of this research was to identify the DM algorithms that can be used efficiently in the field of medical prediction. In that sense, it reaffirmed the importance of diagnosing these diseases in their early stages and consecrated the need for a new approach to reduce the rate of false alarms and increase the detection rate of the disease.

Lastly, a review of sources that describe the application of the different data mining techniques in the medical field is presented in [21] to identify useful classification and clustering approaches for the development of prediction systems. Also, the available data processing and classification tools are discussed and it is explained that, for pattern recognition, the choice of mining tasks depends on the characteristics of

the data. Therefore, authors indicated the use of grouping techniques when the data is not labeled and the classification for the opposite case. Their study highlighted the importance of accuracy in the diagnosis of life-threatening diseases, such as cancer and heart disease, and pointed out that it is a factor that requires a novel approach, which alleviates false alarms and improves the diagnosis in the early stages of the diseases.

2.2. Association Rule Mining. Given the variety of traditional Korean medicine in what medicinal herbs are concerned, in [22] data mining association techniques are used to establish various ways of treating the same disease by addressing etiological factors. As a result of the analysis, representative herbs used specifically in each disease were identified.

In order to overcome the disadvantage of the large volume of rules derived from the application of data mining association algorithms to big medical databases, in [23] an ontology based on measures of great interest that favors the establishment of association rules hierarchies was proposed. Thus, this ontology knowledge mining approach is used to rank the semantically interesting rules. This method was applied to data of an ontology that responded to the mammographic domain.

With the goal of improving the quality of the healthcare service for the elderly, satisfying the medical needs of that social group, and making a better management of the medical resources involved, an intelligent medical replenishment system was designed in [24] that, based on fuzzy association rules mining and fuzzy logic, proved to be very effective.

Moreover, using data mining techniques too, [25] presented a medical diagnostic system for web applications, which helps to reduce expense and time of visiting doctors. Using association rules the system processes the information entered by users, analyzes symptoms and correlations of symptoms, and, based on that, is able to give a preliminary diagnosis.

In addition, the risk factors correlated to diabetes mellitus type 2 (DM2) and the way healthcare providers perform the management of this disease were identified in [26] applying association rules. The experiment was conducted using a database of patients with DM2 treated by a healthcare provider entity in Colombia. Moreover, in [27] risks factors and comorbid conditions associated with diabetes were identified through frequent item set mining, which was applied to a set of medical data. The research proposed a new approach based on the integration of improved association and classification techniques, which resulted in an algorithm with greater analytical and predictive power.

Also, a new data mining framework based on generalized association rules to discover multiple-level correlations among patient data was proposed in [28]. The framework discovered and analyzed correlations among prescribed examinations, drugs, medical treatments, and patient profiles at different abstraction levels. Also, rules were classified according to the involved data features (medical treatments or patient profiles) and then explored in a top-down fashion: from the small subset of high-level rules, a drill-down was performed to target more specific rules. Moreover, in [29] the Intensive Care Units risk prediction system, called ICU ARM-II (Association Rule Mining for Intensive Care Units), was presented. ICU ARM-II is based on a set of association rules that forms a flexible model for the prediction of personalized risk. This approach assumes a classification supported by association.

Data mining techniques can be used to improve decision-making in areas such as hospital management. In that sense, they can be very useful to replace the manual analysis of health insurance data. With the increase of people who have joined a plan, that task, going only to limited professional knowledge, has become increasingly difficult and impossible to perform efficiently. Therefore, [30] proposed a classification based on three criteria (precision, stability, and complexity) to allow a more efficient analysis of the volume of data, compared to a manual analysis. The data set used to test the effectiveness of this approach comprises tens of thousands of patients in a city and hundreds of thousands of medical reimbursement records during the 2010–2015 period. The results of the experiment performed from the medical data analyzed by the FPGrowth algorithm demonstrated that the proposed approach improves the decision model, so that the decision-making gains in flexibility and efficiency and surpasses the other schemes in terms of the precision for classification.

Based on machine learning algorithms for data mining, in [31] a study was performed about the characteristics of diseases caused by the mosquito, such as dengue-1, dengue-4, yellow fever, West Nile virus, and filariasis. Although for some of the above-mentioned diseases there is a cure, the authors assumed that as these mainly affect areas of great poverty such as the African continent and Western Asia, people cannot afford the indicated treatment. Therefore, the objective of this study was to find similar characteristics in the amino acid sequences, which allows creating a cure capable of healing the patient at one time. The results of the study showed that although there appear to be similar features between dengue virus, yellow fever virus, West Nile, and Brugia Malayi mitochondrion, the differences between these are stronger than their similarities. On the other hand, authors discovered that Leucine control might contribute to the development of a single effective cure for the cases of West Nile and Brugia Malayi mitochondrion.

Finally, because of the high cancer mortality, [32] investigated the sequences of various cytokines using algorithms such as Apriori, decision tree, and support vector machines (SVM). Cytokines play a central role in the immune system, so the study, if its goal is achieved, may contribute to others in finding new rules to determine whether a cytokine may have anticancer properties or not.

Table 1 shows a summary of these investigations.

As we can see in Table 1, the different studies mentioned before demonstrate the usefulness of data mining techniques for the solution of problems in the medical area, which is raised as a great object of study, with appropriate problems to be studied from this perspective. Nevertheless, to the best of our knowledge, there are not works which used association rule mining and Bayesian networks to analyze the decrease in the number of autopsies performed in a hospital; therefore

Table 1: Related works.

Work	DM techniques	DM tasks
[10]	Decision trees, Neural Networks	Classification
[11]	SVM	Classification
[12]	Bayesian networks	Classification
[13]	Logistic regression, NB	Classification
[14]	Re-RX, J48graft	Classification
[15]	NB, SVM	Classification
[16]	NB, SVM, RF	Classification
[17]	J48, RF, KNN, NB, SVM	Classification
[18]	NB, SVM, logistic regression, RF	Classification
[19]	NB, OTM, InterVA-4	Classification
[21]	Decision tree, Neural Networks	Classification
[22]	Association rules	Association
[23]	Apriori	Association
[24]	Fuzzy association rules Mining and fuzzy logic	Association
[25]	Formal Concept Analysis	Association
[27]	Apriori Split and Merge (SAM)	Association Classification
[28]	Association rules	Association
[29]	CBA (classification based on association)	Classification Association
[26]	Apriori, FPGrowth	Association
[30]	FPGrowth Decision trees	Association Classification
[31]	Apriori Decision trees	Association Classification
[32]	Apriori Decision trees, SVM	Association Classification

this determines the appropriateness, novelty, and interest of this research.

3. Methods

3.1. Collection and Preparation of Data. In order to carry out the study, it was necessary to collect data that record aspects about the opinion, attitudes, or behaviors of the physicians about the practice of autopsies, as well as the values, beliefs, or motifs that characterize them. To do this, one of the department's pathologists compiled a survey of 16 questions, divided into three open type and thirteen closed type, of which five include a section to specify other responses considered by the doctors. Table 2 shows a summary of the survey applied to the physicians and the number of categories generated by response.

The survey was answered by 86 physicians of the hospital. Their answers were processed, resulting in the following:

(i) 27 categories related to factors that the doctors considered negatives for the realization of autopsies and 26 categories for the positive factors were generated.

(ii) Nine motives for the family for autopsy rejection and eight possible reasons for the nonrealization of enough autopsies in the hospital were extracted.

(iii) Regarding the opinion of the physicians about the procedure to request an autopsy, 14 efficient methods and six options about the suitable staff to request an autopsy were considered.

(iv) The answers of general comments given by the doctors were reduced to 25 categories.

(v) The remaining questions kept the proposed options of the questionnaire, three possible answers for the area, and the grade of the doctor and five for each of the three questions related to the medical opinions about the discoveries found in autopsies.

A database was designed and implemented to store the information obtained from the surveys and ensure the persistence of these data, so that they could be used in subsequent analyses. This database was implemented using the PostgreSQL database management system. From the database records, two suitable representations (*binary-matrix* and *minable-view*) were created to apply the DM techniques. These structures were created by SQL functions, generated dynamically by the tables, and in this way, two different data sets were formed from the same data. In this paper, the *binary-matrix* table will be named as C and the *minable-view* table as D. Their characteristics are described in Table 3.

Table 2: Summary of the survey applied to the medical staff.

Aspects	Questions	Code	Type of question	Generated categories
Medical training	Area	*area*	Closed	3
	Grade	*grade*	Closed	3
	General medicine training center	*gral_med_inst*	Closed	47
	Medical specialty training center	*spec_ins*	Closed	47
Medical experience	Years of medical practice	*years_pract*	Closed	5
	Participation in autopsy cases	*cases*	Closed	5
Discoveries in autopsies	Cause discrepancy with the clinical diagnoses	*disc_findings*	Closed	5
	Originate in claim cases	*dem_findings*	Closed	5
	Originate in arbitration cases	*arb_findings*	Closed	5
Request of autopsies	Motives for autopsy acceptance	*aut_reason*	Open	26
	Motives for autopsy rejection	*reason_no_aut*	Open	27
	Motives for autopsy rejection by family	*fam_rejection*	Closed	9
	Motives for not enough autopsies performed in the hospital	*no_hosp*	Closed	8
Procedure to request an autopsy	Suitable staff to request an autopsy	*staff_request_aut*	Closed	6
	Efficient methods to request an autopsy	*method_request_aut*	Closed	14
General aspect	Comments	*com_sug_op*	Open	25

Table 3: Characteristics of C and D data sets.

Characteristics	C data set	D data set
Attributes	166	18
Objects	4	7859
Type of data	Nominal-binary-asymmetric	Nominal
Description	Binary-matrix ⟨answer, value⟩	Matrix represented by ⟨question, answer⟩
Missing values	Yes	No
Values out of range	No	No
Inconsistent values	No	No

Minable-View. The function constructs a matrix where rows mean combinations of answers for polls and columns represent the answers. The value of each column responds to the intersection that can be read as a pair ⟨question, answer⟩.

Binary-Matrix. The function constructs a binary-matrix, in which each row represents a respondent and the columns represent the answers. The value of each column responds to the intersection that can be read as a pair ⟨answer, value⟩, value being equal to "S" if that answer was answered and void otherwise.

Open answers provide research with a higher level of complexity because in these cases respondents can answer the question by writing a free idea. Because of this, the system has to perform an automatic categorization of the text, where it predicts or assigns a category to that response. To do this, we needed the classified data sets of each question to train the prediction model, so it was necessary that the experts established the possible categories and manually sorted the open answers of the recorded surveys (see Tables 4, 5, and 6). In this way, the data sets *"aut_reason"*, *"reason_no_aut"*, and *"com_sug_op"* were created with the answers to open questions. These ones answered for reasons for requesting autopsies, reasons for not requesting them, and comments (see characteristics in Table 7).

Table 4: Reasons, causes, and circumstances for requesting autopsies.

Reason
(a) Establish definitive diagnoses
(b) Subjects related to forensic medicine
(c) Pedagogical objectives
(d) Applicability of the study
(e) Lack of correlation between clinical and laboratory data
(f) Provide essential information to relatives in cases of infectious or congenital diseases
(g) Correct or incorrect application of the used treatment
(h) Health care problems
(i) Difficult diagnosis

Table 5: Reasons, causes, and circumstances for not requesting autopsies.

Reason
(a) That the service does not exist
(b) Fear of demand
(c) Negative of relatives
(d) Ignorance of the practice of autopsies
(e) Disinterest
(f) Known base disease
(g) Administrative formalities
(h) Lack of indication
(i) Religious
(j) Cultural
(k) Legal
(l) When the body is in decomposition
(m) Of the condition of the deceased
(n) Lack of human resources
(ñ) Lack of material resources
(o) Lack of financial resources
(p) When it is done without teaching purposes
(q) Social
(r) That the pathology service does not perform it
(s) Political
(t) Lack of doctor-patient communication
(u) Forgetfulness in the practice of autopsies

Table 6: Comments, opinions, and suggestions.

Comment
(a) No comment
(b) It should be routine
(c) Inadequate teaching of general pathology
(d) Usefulness of the survey
(e) Importance of autopsies
(f) Observation of the survey
(g) Reasons to request
(h) To strengthen the relationship between pathological anatomy and general surgery
(i) Clarification
(j) Participation of all services
(k) Acknowledgments
(l) Agreeing on a difficult clinical case among all
(m) Increase the number of autopsies
(n) Residents of pathology
(ñ) Compelling residents
(o) They have never performed autopsies on their patients
(p) Request consents
(q) Add pathology to the rotation of internal physician

The characteristics of the data sets were analyzed and it was determined that no transformation was necessary because they did not affect the performance of the algorithms that were contemplated to evaluate. So, we went directly to the DM phase.

3.2. Evaluation of Algorithms. According to the data and the objective of this paper, two DM tasks were considered to solve the problem. It was first thought to perform an association analysis to determine the relationships between the attributes and on the other hand to use the classification to recognize the relevant dependencies between attributes, according to probability and statistics, by using also Bayesian networks. Other data mining techniques for text classification were considered too. The comparison of the evaluation of the WEKA algorithms for each DM task considered in the research is presented below.

3.2.1. Association Algorithms

Apriori [33]. It is a classic algorithm for association rule mining. It generates rules through an incremental process that searches for frequent relationships between attributes bounded by a minimum confidence. The algorithm can be configured to run under certain criteria, such as upper and lower coverage limits, and to accept sets of items that meet the constraint, the minimum confidence, and order criteria to display the rules, as well as a parameter to indicate the specific amount of rules we want to show.

FPGrowth [34]. It is based on Apriori to perform the first exploration of the data, in which it identifies the sets of frequent items and their support, value that allows us to organize the sets in a descending way. The method proposes good selectivity and substantially reduces the cost of the search, given that it starts by looking for the shortest frequent patterns and then concatenating them with the less frequent ones (suffixes), and thus identifying the longest frequent patterns. It has been shown to be approximately one order of magnitude faster than the Apriori algorithm.

PredictiveApriori [35]. The algorithm achieves a favorable computational performance due to its dynamic pruning technique that uses the upper bound of all rules of the supersets of a given set of elements. In addition, through a backward bias of the rules, it manages to eliminate redundant ones that are derived from the more general ones. For this algorithm, it is necessary to specify the number of rules that are required.

Tertius [36]. It performs an optimal search based on finding the most confirmed hypotheses using a nonredundant refinement operator to eliminate duplicate results. The algorithm has a series of configuration parameters that allow its application to multiple domains.

The measures considered in the evaluation were as follows:

(i) Confirmation: this statistical measure indicates how interesting a rule can be.

Table 7: Characteristics of data sets for the classification of open-ended questions.

Data set	Number of attributes	Attribute/type	Class	Objects
aut_reason	2	*aut_reason*: nominal *text*: String	aut_reason	891
reason_no_aut	2	*reason_no_aut*: nominal *text*: String	reason_no_aut	698
com_sug_op	2	*com_sug_op*: nominal *text*: String	com_sug_op	472

Table 8: Application of the algorithms in the data sets.

Data set	Apriori	FPGrowth	PredictiveApriori	Tertius
C	√	√	√	√
D	√	×	×	√

(ii) Support: it represents the percentage of transactions from a transaction database that the given rule satisfies.

(iii) Confidence: it assesses the degree of certainty of the detected association.

(iv) Time: amount of milliseconds that takes the construction of a model.

In order to determine which algorithms could be applied to C and D data sets, an analysis was made based on the characteristics of the data sets, such as their attribute types and whether they contained missing, out of range, or inconsistent values. The results of such analysis are shown in Table 8.

The association algorithms were grouped, taking into account their configuration characteristics to compare each other. Therefore, Apriori and FPGrowth were first analyzed. For this, different support and confidence thresholds were established, since rules are considered interesting and strong if they satisfy both a minimum support threshold *(min_sup)* and a minimum confidence threshold *(min_conf)* [34]. Moreover, the number of rules generated, the execution time of the algorithm (in milliseconds), and the support and confidence averages for each case were recorded. On the other hand, to analyze PredictiveApriori and Tertius it was necessary to specify the number of rules to be generated by these algorithms. The execution time of the algorithm and the average of support and confidence were registered too. Finally, we compared the best-evaluated algorithms in each of these cases considering the following variables: number of rules generated, execution time, and average values of support.

The results of the evaluations of the algorithms for the C data set were recorded in Tables 9 and 10. Each evaluation was executed 100 times to estimate the average time for the construction of the models. Also, the average values of support and confidence were taken into account. Table 9 shows the comparison between Apriori and FPGrowth, as we can see the latter is computationally faster than the former. Moreover, although Apriori found more rules than FPGrowth in seven cases, the average confidence of the rules found by FPGrowth is greater in three cases and only lowest in one case, while the average support of its rules overcame the average support of the rules obtained by Apriori in four cases. Therefore, FPGrowth is better than Apriori for the C data set.

The comparison between PredictiveApriori and Tertius is presented in Table 10. Experiments demonstrate that Tertius obtained the same number of rules as PredictiveApriori in considerably lower time. Nevertheless, the average support of the rules found by the latter is greater than the average support of the rules of the former in three cases and lower only one time. Figures 2 and 3 show the comparison between the four algorithms with respect to support and time, respectively. Table 9 and Figure 3 show that FPGrowth is the fastest algorithm. In contrast, Table 10 and Figure 3 demonstrate that PredictiveApriori is the slowest. Also, the results indicate that the algorithms that generate rules with better support within the C data set are Apriori and FPGrowth. Thus, the best algorithm for the data set C is FPGrowth.

Table 11 shows the comparison results of the algorithm evaluations for the D data set. Also, each evaluation was performed 100 times to estimate the average time for the construction of the models and the number of rules and average support were taken into account. Apriori reports better results than Tertius because the time it takes to obtain the same number of rules is considerably shorter and the rules that it identifies have better support.

Although FPGrowth was the best algorithm for the C data set, it has the disadvantage that it cannot be applied to the D data set; therefore the Apriori algorithm is considered more appropriate for this work since it generates more rules than FPGrowth and can be used for the two data sets contemplated in this research, as illustrated in Figure 4.

3.2.2. Classification Algorithms. Bayesian networks were considered to analyze the data of the surveys, whereas J48, Neural Networks, NaiveBayes, and Sequential Minimal Optimization (SMO) were studied considering their application in the classification process for the open questions. We performed

Table 9: Test results for Apriori and FPGrowth for the C data set.

Algorithms	*min_conf*/*min_sup*	Rules	Time	Confidence	Support
Apriori	0.9/0.5	9	11	0.93	0.57
FPGrowth		9	**4**	0.93	0.57
Apriori	0.9/0.5	11	9	0.92	0.58
FPGrowth		11	**5**	0.92	0.58
Apriori	0.8/0.6	2	9	0.88	0.76
FPGrowth		2	**3**	0.88	0.76
Apriori	0.9/0.4	**25**	15	0.93	0.49
FPGrowth		24	**8**	0.93	0.49
Apriori	0.9/0.3	**87**	21	**0.95**	0.38
FPGrowth		78	**12**	0.94	**0.39**
Apriori	0.8/0.3	**167**	20	0.90	0.37
FPGrowth		140	**12**	**0.91**	**0.39**
Apriori	0.8/0.4	**48**	13	0.89	0.47
FPGrowth		47	**8**	**0.90**	0.47
Apriori	0.9/0.2	**568**	45	0.95	0.25
FPGrowth		528	**28**	**0.96**	**0.26**
Apriori	0.8/0.2	**985**	**45**	0.91	0.25
FPGrowth		961	26	0.91	0.25
Apriori	0.9/0.1	**16463**	285	0.97	0.12
FPGrowth		9349	**108**	0.97	**0.13**
Apriori	0.7/0.6	2	9	0.88	0.76
FPGrowth		2	**4**	0.88	0.76
Apriori	0.7/0.5	12	11	0.90	0.58
FPGrowth		12	**4**	0.90	0.58

Table 10: PredictiveApriori and Tertius test results for the C data set.

Algorithms	Rules	Time	Confidence	Support
PredictiveApriori	9	4050	0.78	**0.29**
Tertius	9	**97**	-	0.25
PredictiveApriori	11	4136	0.82	**0.28**
Tertius	11	**100**	-	0.26
PredictiveApriori	2	2794	1	0.33
Tertius	2	**83**	-	**0.49**
PredictiveApriori	2	2877	1	**0.33**
Tertius	2	**99**	-	0.22
PredictiveApriori	12	4109	0.83	0.27
Tertius	12	**97**	-	0.27

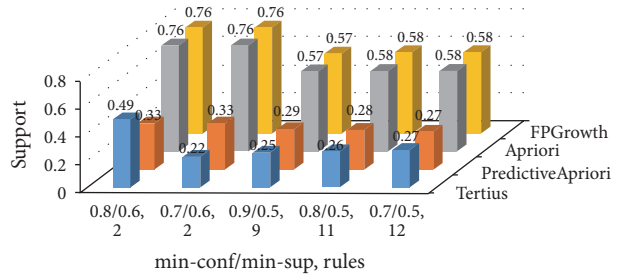


Figure 2: Comparison of Apriori, FPGrowth, PredictiveApriori, and Tertius algorithms in terms of support.

Table 11: Test results for Apriori and Tertius for the D data set.

Algorithms	Min Conf/Sup	Rules	Time	Support
Apriori	0.8/0.6	5	**112**	**0.65**
Tertius		5	9514	0.41
Apriori	0.7/0.6	6	**118**	**0.64**
Tertius		6	7467	0.42
Apriori	0.9/0.5	10	**149**	**0.56**
Tertius		10	9529	0.33
Apriori	0.8/0.5	28	**144**	**0.56**
Tertius		28	9700	0.42
A priori	0.7/0.5	36	**147**	**0.56**
Tertius		36	9471	0.42
Apriori	0.9/0.4	82	**198**	**0.45**
Tertius		82	10042	0.39

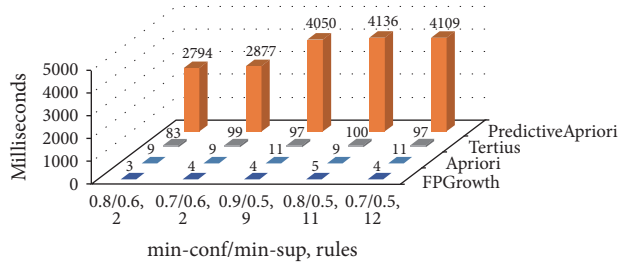


Figure 3: Comparison of Apriori, FPGrowth, PredictiveApriori, and Tertius algorithms in terms of time.

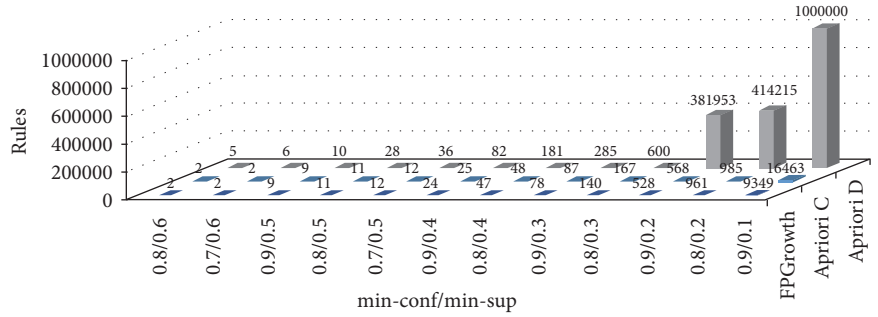


Figure 4: Comparison of Apriori algorithm in C and D data sets and FPGrowth for the C data set, with respect to number of rules.

10-fold cross-validation in all cases to avoid any problem of overfitting.

BayesNet. It determines the relations of dependence and probabilistic independence between all the variables of a data set, thus conforming the structure of the Bayesian network, represented by an acyclic graph where the nodes are the variables and the arcs are the probabilistic dependencies between the linked attributes [37, 38].

J48. It constructs a binary decision tree to model the classification process. This algorithm ignores the missing values or predicts them according to the known values of the attribute in the other registers [39, 40].

Neural Networks. These are mathematical procedures based on the exploitation of parallel local processing and the properties of distributed representation that imitate the structure of the nervous system and can be interpreted as the way to obtain knowledge from experience [41].

NaiveBayes. It is a probabilistic classifier that calculates the probabilities according to the combinations and frequencies of occurrence of the data in a given data set [39, 40].

SMO. It implements the algorithm to train SVMs and solve the problems of quadratic programming that these presuppose [42].

Table 12: Application of Bayesian networks in datasets C and D.

Data sets	Bayesian networks
C	×
D	√

The measures considered in the evaluation were as follows:

Accuracy. It is the percentage of test set tuples that are correctly classified by the classifier.

ROC Area. It refers to the area under the curve between true positives (y-axis) and false positives (x-axis); the result is better when it gets closer to one.

Kappa. It determines how good a classifier is according to the concordance of the results obtained by several classifiers of the same type. Values close to 1 affirm a good concordance, while values close to 0 show a concordance due exclusively to chance.

Time. It is the amount of milliseconds that takes the construction of a model.

The possibility of applying Bayesian network learning using various search algorithms in C and D sets was analyzed (see Table 12), as well as J48, Neural Networks, NaiveBayes, and SMO in *aut_reason*, *reason_no_aut*, and *com_sug_op* data sets (see Table 13).

To evaluate the Bayesian networks, the 18 attributes of D data set were considered as classes. Each test was executed 100 times to estimate the average time of construction of the network. Accuracy and ROC area values were considered too. Table 14 shows the results of the tests carried out by taking the grade of the respondent *(grade)* as a class. In the same way, the results of the 17 classes were recorded, which shows that the best results (see Table 18) were obtained with the TAN (Tree Augmented Naïve Bayes) search algorithm for 14 of the classes and HillClimber for the remaining 4. As we can see in Table 14, HillClimber for the class *grade* is the best algorithm because although TabuSearch, RepeatedHillClimber, and HillClimber present higher accuracy and ROC area than the other algorithms, HillClimber obtains the Bayesian network in lower time.

The evaluation of J48, Neural Networks (MultilayerPerceptron), NaiveBayes, and SMO algorithms, considered for text classification, is described in Tables 15, 16, and 17 for the *aut_reason*, *reason_no_aut*, and *com_sug_op* data sets, respectively. Each test was performed 100 times to estimate average values in terms of time and in addition other metrics such as accuracy, ROC area, and kappa were considered.

Table 18 presents the best cases for each algorithm analyzed according to the different data sets. This information can be very useful to guide the specialist about the parameters that must be generated by the models and thus obtain more accurate results. It should be noted that only the best configuration for the algorithms is attempted, but regardless of this, the specialists can configure them according to their interests.

3.2.3. Robustness Evaluation. Robustness is the ability of the classifier to make correct predictions given noisy data or data with missing values. This is typically assessed with a series of synthetic data sets representing increases in degrees of noisy and missing values [34]. To evaluate the robustness of the Bayesian network search algorithms, we create 18 data sets with 10% of missing values for each attribute, then we execute every algorithm considering each attribute as a class. Figures 5 and 6 show the evaluation results. TAN was the best algorithm because it had the greatest accuracy for 14 of the 18 classes. Also, it got the best ROC area for the 18 classes.

4. Results

The algorithms evaluated in Section 3.2 of this article were implemented in the web application, a tool proposed by the authors of this work to support the process of finding useful knowledge in the applied survey, generate the models, and submit the results to a thorough evaluation by the experts. These algorithms were configured with the parameters that gave rise to the best results during the evaluation stage.

The application allows physicians to answer the survey, which is the subject of study in this research. In addition, it guarantees the persistence of data and from these the data sets that the mining algorithms considered in this work must analyze are generated. The application, using the WEKA API, generates models according to the selected algorithms and returns the results. The format of the rules generated by the algorithms can be understood by experts in data mining but it is difficult to be understood by a common user, so it was decided to describe in the application each variable involved in the survey. From this, and using a pattern to define the explanation of a rule, it was possible to program a function that, given the rules of a model, returns the explanation of these in a natural language expression. In this way, specialists can perform evaluations without necessarily relying on data mining experts.

4.1. Association. We obtained the best 20 rules from each data set using Apriori, FPGrowth, PredictiveApriori, and Tertius algorithms. Then, the rules were evaluated by the expert. Figure 7 shows the rules generated by the application from the C data set using the Apriori algorithm.

4.2. BayesNet. Given the interest of the specialist to know the behavior of the data for the questions about the reasons why no autopsies are requested and the reasons why they are requested, two Bayesian networks were built taking these attributes as classes. The application shows the accuracy for each network and also represents them by nondirected acyclic graphs. This allows us to graphically appreciate the relationships between the nodes. The graph of the second network is shown in Figure 8.

Table 13: Application of the algorithms in the *aut_reason*, *reason_no_aut*, and *com_sug_op* data sets.

Data sets	J48	NaiveBayes	Neural networks	SMO
aut_reason	√	√	√	√
reason_no_aut	√	√	√	√
com_sug_op	√	√	√	√

Table 14: Results of Bayesian networks for *grade* class.

Class: grade	Accuracy	ROC area	Time
K2	0,995	0,999	59
TAN	0,999	0.999	55411
TabuSearch	0,999	1	21571
RepeatedHillClimber	0,999	1	117975
LAGDHillClimber	0,993	0,999	310
HillClimber	**0,999**	**1**	**10215**

Table 15: Test results for *aut_reason* class.

Class: aut_reason	Accuracy	ROC area	Kappa	Time
J48				
GainRatio	0,86	0,96	0,79	998
InfoGain	0,86	0,97	0,80	1273
NaiveBayes				
GainRatio	0,75	0,93	0,67	625
InfoGain	0,75	0,93	0,67	647
MultilayerPerceptron				
GainRatio	0,86	0,97	0,81	44611
InfoGain	**0,86**	**0,97**	**0,82**	56820
SMO				
GainRatio	0,83	0,91	0,75	1168
InfoGain	0,83	0,91	0,75	560

Table 16: Test results for *reason_no_aut* class.

Class: reason_no_aut	Accuracy	ROC area	Kappa	Time
J48				
GainRatio	0,78	0,97	0,74	1098
InfoGain	0,78	0,97	0,74	1180
NaiveBayes				
GainRatio	0,76	0,95	0,65	1279
InfoGain	0,76	0,95	0,65	1244
MultilayerPerceptron				
GainRatio	0,84	0,97	0,76	33674
InfoGain	**0,85**	0,96	**0,76**	**15904**
SMO				
GainRatio	0,81	0,94	0,73	995
InfoGain	0,81	0,94	0,73	696

4.3. Evaluation of the Results. To support the expert in the evaluation process, the system provides a natural language explanation of the mining models results, so that the pathologist understands them. In this way, the specialist can evaluate the results by subjectively analyzing, based on his experience and knowledge, the information extracted by the models.

The association rules analyzed by the specialist were the top 20 of each algorithm implemented in this research. For

Table 17: Test results for *com_sug_op* class.

Class: com_sug_op	Accuracy	ROC area	Kappa	Time
J48				
GainRatio	0,84	0,97	0,85	266
InfoGain	0,84	0,97	0,85	326
NaiveBayes				
GainRatio	0,84	0,97	0,76	332
InfoGain	0,84	0,97	0,76	315
MultilayerPerceptron				
GainRatio	0,90	0,97	0,88	14040
InfoGain	**0,92**	**0,98**	0,90	25796
SMO				
GainRatio	0,83	0,96	0,92	1473
InfoGain	0,84	0,96	0,83	1208

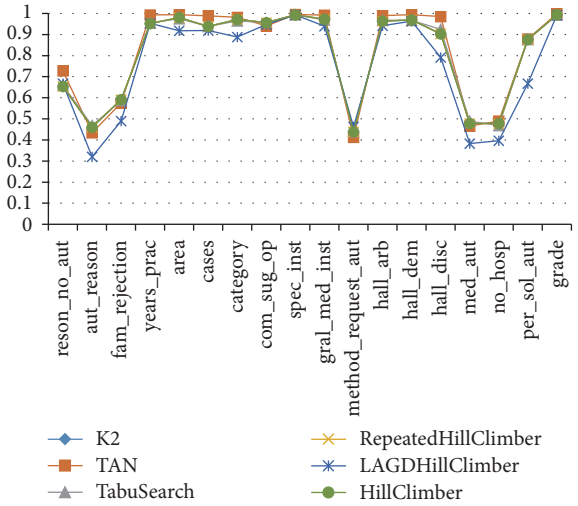


Figure 5: Comparison of Bayesian network search algorithms in data sets with missing values considering their accuracy.

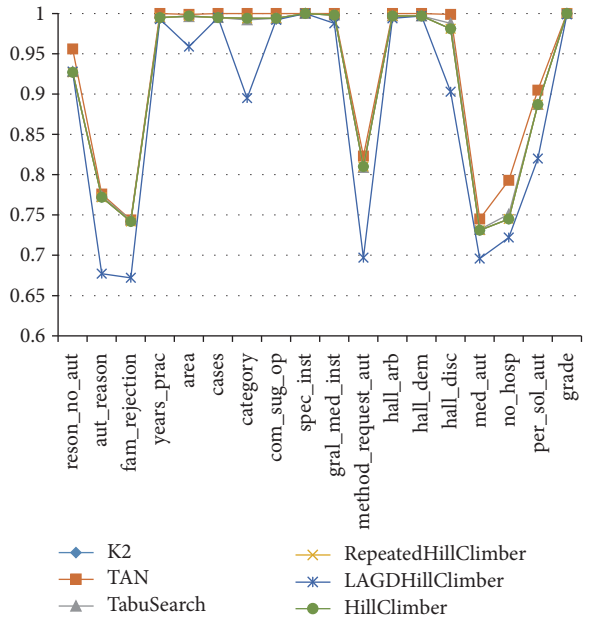


Figure 6: Comparison of Bayesian network search algorithms in data sets with missing values considering ROC area.

Table 18: Presentation of the best results of each algorithm evaluation.

Association analysis			
Data set	Algorithm	Parameters	
C	Apriori	Confidence = 0.9 Support = 0.4 Rules = 15	
	FPGrowth	Confidence = 0.9 Support = 0.4 Rules = 8	
	PredictiveApriori	Support = 0.5 Rules = 12	
	Tertius	Support = 0.5 Rules = 12	
D	Apriori	Confidence = 0.9 Support = 0.5 Rules = 10	
	Tertius	Support = 0.5 Rules = 10	
Classification analysis			
Data set	Algorithm	Class	Search algorithm
D	BayesNet	*reason_no_aut*	Tan
		aut_reason	HillClimber
		fam_rejection	Tan
		years_pract	Tan
		area	Tan
		cases	Tan
		category	Tan
		com_sug_op	Tan
		spec_inst	HillClimber
		gral_med_inst	Tan
		method_request_aut	HillClimber
		arb_findings	Tan
		dem_findings	Tan
		disc_findings	Tan
		physician_reason_aut	Tan
		no_hosp	Tan
		staff_request_aut	Tan
		grade	HillClimber
Data set	Algorithm	Attribute selection measures	
aut_reason	MultilayerPerceptron	InfoGain	
reason_no_aut	MultilayerPerceptron	InfoGain	
com_sug_op	MultilayerPerceptron	InfoGain	

example, the 20 association rules obtained by Apriori in the C data set have been shown in Figure 7.

For each node in a Bayesian network, the application shows a conditional probability table containing all probabilities of occurrences for its attributes. For this, the Bayesian networks provide much information to be analyzed, but this does not mean that everything is interesting. For this reason, the analysis of the two networks generated was limited to six survey questions, considering only the highest probability values. The questions are related to the years of practice of the physician, the number of cases in which the doctor has intervened, discrepant findings, demand findings, causes of autopsies rejection, reasons for not requesting autopsies in the hospital, and why the physician does not perform it, among others.

To illustrate this procedure more clearly, we describe the analysis process for the *no_hosp* question from the Bayesian network generated from the query *aut_reason* of Figure 8. From this network, we selected the node corresponding to the question of interest for this case *(no_hosp)*, which generated

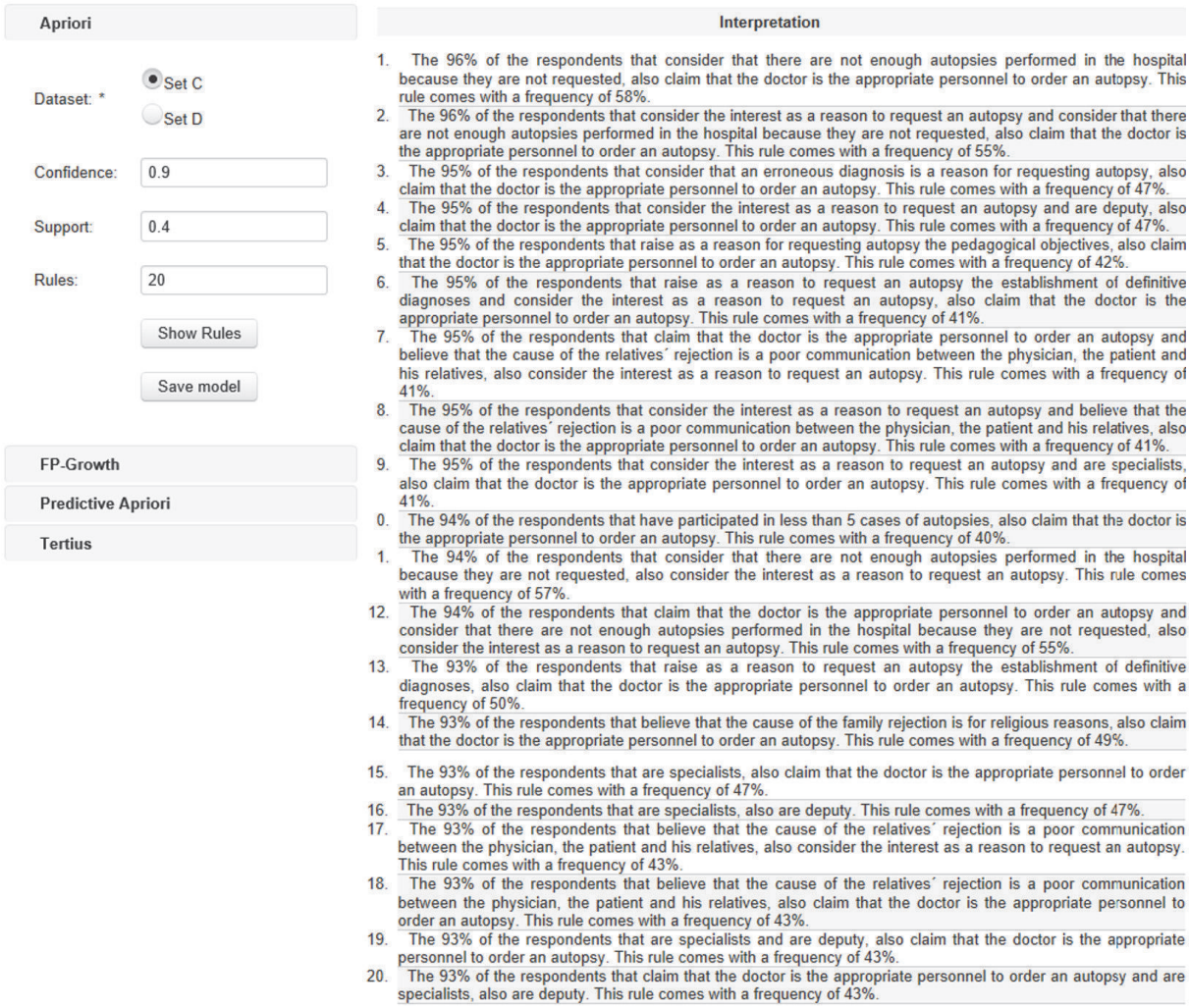


Figure 7: Apriori model for the C data set.

a total of 1875 conditional probabilities. The next step was to rule out the conditional probabilities less than 50% and this reduced to 14 the number of results (see Figure 9). In this particular case, the specialist ruled out the conditional probabilities 1, 2, 3, 4, and 11. With this procedure the remaining questions selected in each network were evaluated.

4.3.1. Association. The results comprised 120 rules, 20 per model, of which the specialist evaluated only 100. The decision of not including the rules of FPGrowth was made to avoid repetitions in the results, given that the majority of the rules are also obtained by Apriori.

After a thorough analysis of the rules, the conclusive results were as follows: from 100 rules, the expert approved 75. Eight rules were discarded in each model of Apriori, seven in PredictiveApriori and two in the model of Tertius with the D data set. The most accepted algorithm turned out to be Tertius with 90% of rules approved for the D data set and 100% for the C data set (see Table 19). In general, it can be concluded in the association analysis that the results had a 75% of approval.

4.3.2. BayesNet. It is complex to analyze the data generated by Bayesian models due to the large volume of probability relationships that have been extracted from these networks. This is why, for this research, the expert delimited the analysis to the results with a probability greater than 50% that relate the attributes: *years of practice*, *cases in which the physician has intervened, discrepant findings, demand findings*, *causes of autopsies rejection*, *reasons for not requesting autopsies in the hospital,* and *why the physician does not do it*. In this way, 352 probability relationships were evaluated. 168 were extracted from the network generated from the *aut_reason* class and 184 from the *reason_no_aut* class (see Table 20).

After a thorough analysis, the conclusive results were as follows: for a total of 352 conditional probabilities, 347 were approved by the expert. The specialist discarded one probability in the Bayesian network for *aut_reason* and four probabilities in the Bayesian network for *reason_no_aut*. In general, it is concluded that Bayesian networks had a 98.6% approval (see Table 21).

The networks also made establishing a situational diagnosis of autopsies in hospital possible, which is detailed in

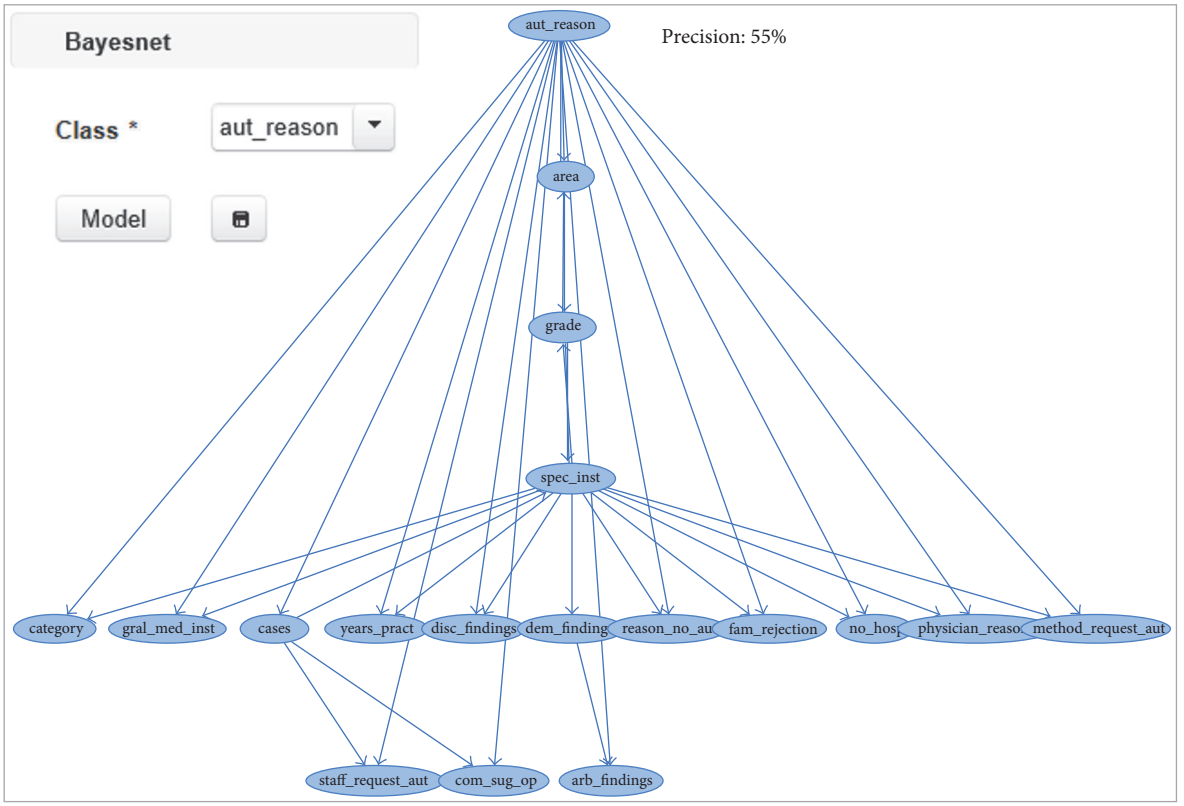


Figure 8: Bayesian network graph, considering motives for autopsy acceptance as class.

Table 19: Evaluation of association results.

Algorithm	Data set	Accepted	Discarded	Rules discarded	Acceptance
Apriori	C	12	8	9, 10, 15, 16, 17, 18, 19, and 20	60%
	D	12	8	8, 11, 14, 15, 17, 18, 19, and 20	60%
PredictiveApriori	C	13	7	6, 11, 12, 13, 14, 16, and 18	65%
Tertius	C	20	0		100%
	D	18	2	4 and 17	90%

Table 20: Results of Bayesian networks.

Bayes networks	Results
aut_reason	168
reason_no_aut	184
Total	*352*

Figures 10 and 11. These diagnoses are referred to the causes and reasons for requesting and not requesting autopsies, respectively.

5. Conclusions and Future Work

The prominent decrease in the number of autopsies in the hospitals around the world has raised questions about the motives for this phenomenon. The purpose of this work was to analyze the possible causes of the reduction of autopsies in the hospital system of "Servicios de Salud de Veracruz" by means of association rule mining and Bayesian networks from the data that belong to the medical opinions about such medical practice.

The analyzed data were collected through a survey that was applied to the doctors of the hospital. The survey was focused on the medical opinions about the causes or reasons of autopsies that were not performed, the study level of the specialists, their years of experience, and the cases of autopsies they have been involved in, among others.

The use of association rule mining techniques and Bayes networks allowed us to perform a descriptive analysis of the problematic situation and find the correlations between the categorical attributes of the data set, which formed the information obtained from the medical staff, all this

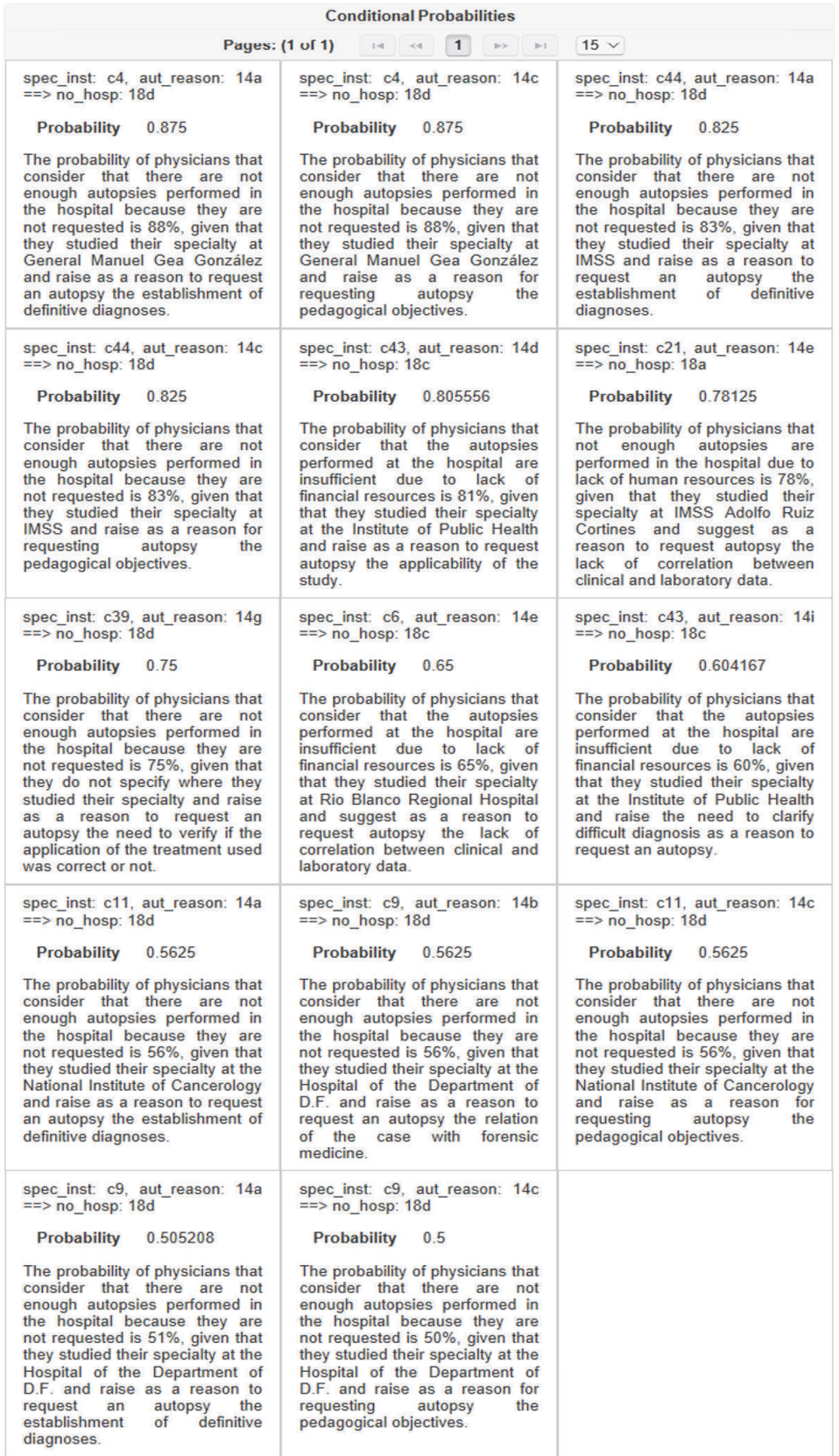


Figure 9: Results of relations for the node: *reasons for not requesting autopsies*.

Table 21: Evaluation of the results of Bayesian networks.

Results	Accepted	Discarded	Acceptance
aut_reason	167	1	99.4%
reason_no_aut	180	4	97.8%
Total	347	5	98.6%

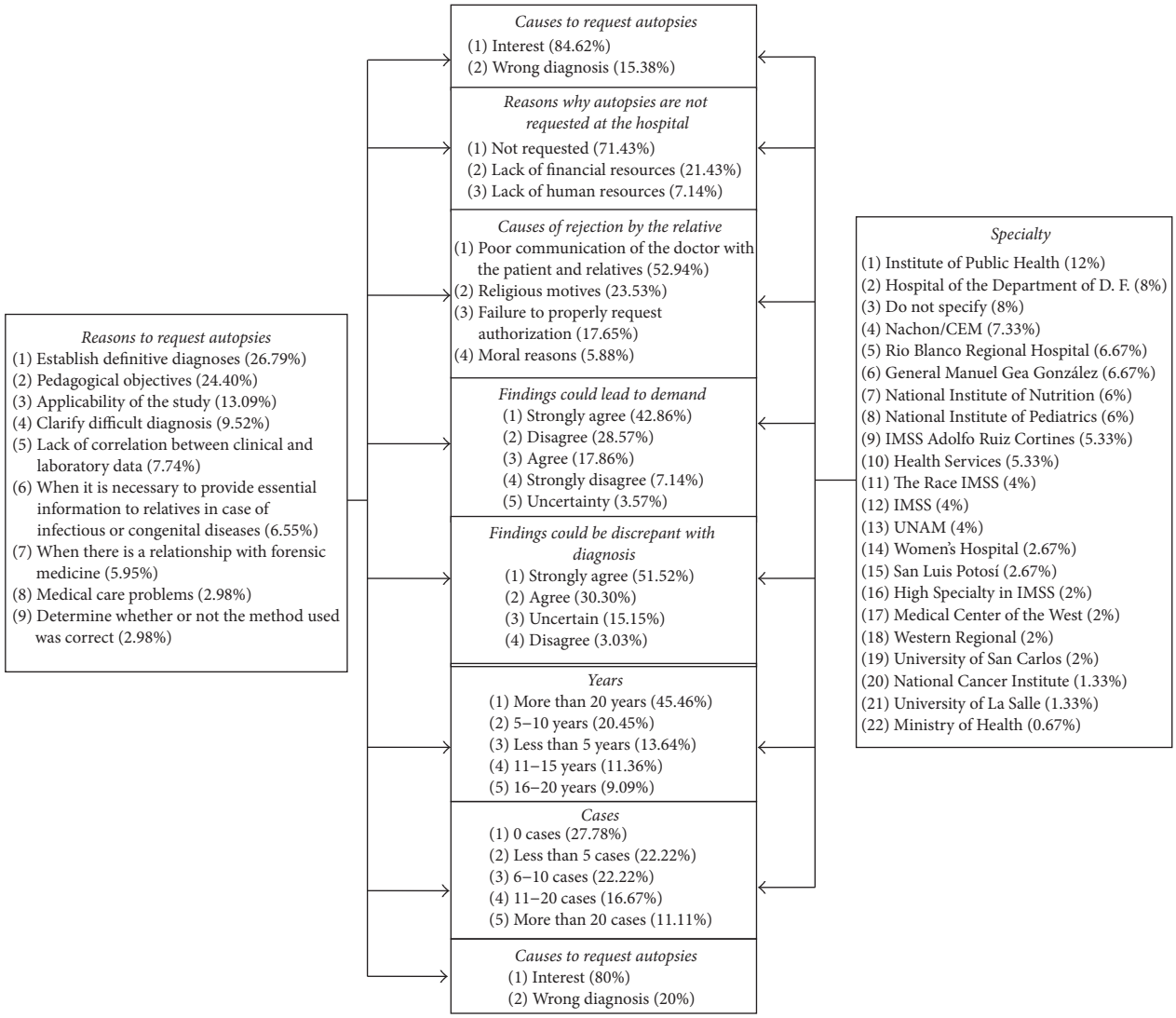


Figure 10: Situational diagnosis of autopsies in the hospital on the reasons for requesting autopsies.

through a web application or system developed especially for the case. The system provides a natural language explanation of the mining models results, so that the pathologist understands them. In this way, the specialist can evaluate the results by subjectively analyzing, based on his experience and knowledge, the information extracted by the models.

The rules generated by the association models instrumented for the research had 75% approval by the specialist. As for the algorithms, Tertius proved to be the most accurate, with 90% approval of its rules in the C data set and 100% in the D data set.

As future work we suggest studying data of the clinical records of the patients who died in the hospital and analyzing with real data the trend of the causes that lead to perform autopsies in some patients and not in others. This will confirm the veracity of the results of this research. We also recommend to perform similar studies in other parts of the country and to identify whether the medical opinions and the consequences of autopsies rejection differ by region.

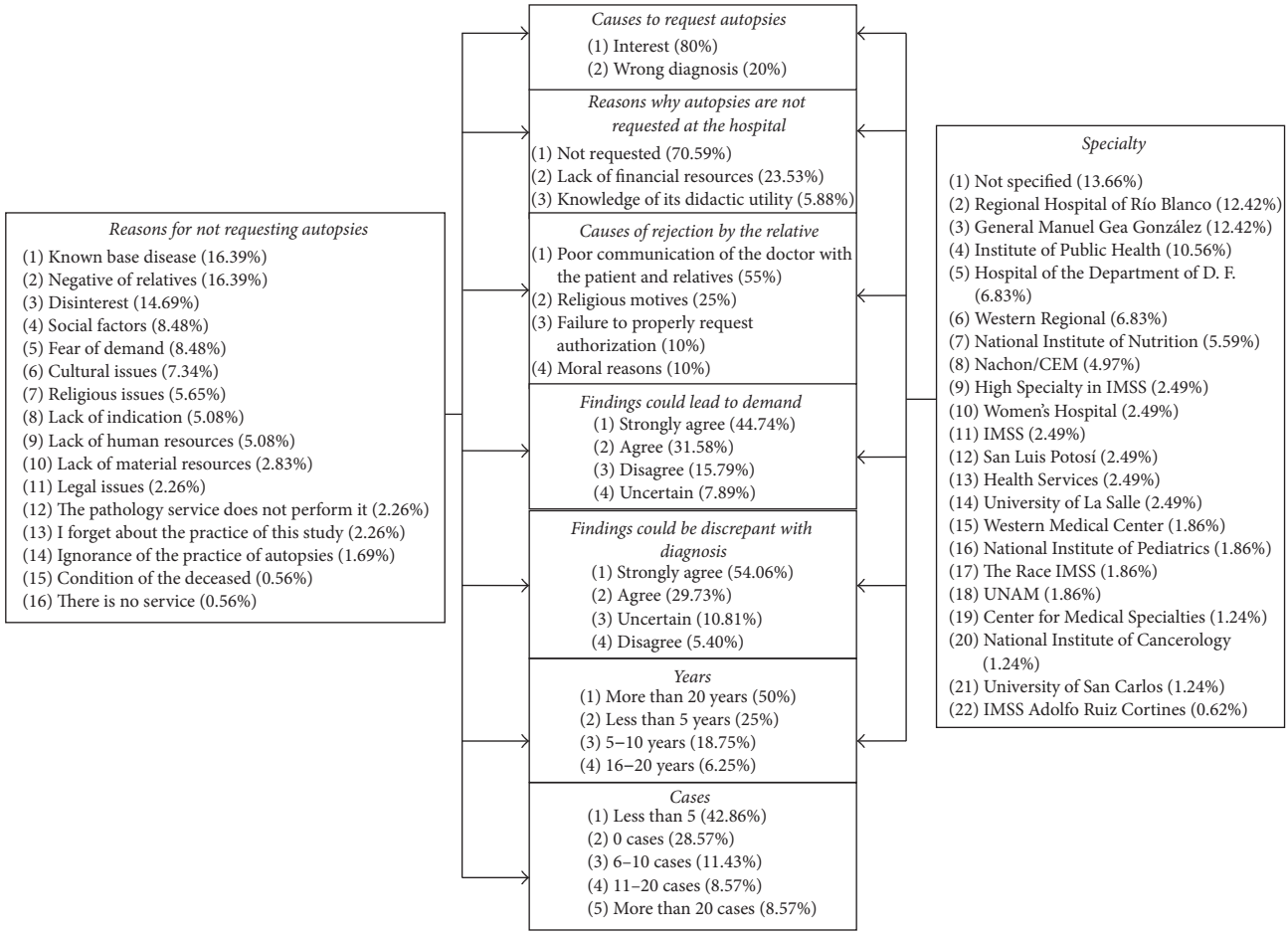


Figure 11: Situational diagnosis of autopsies in the hospital on the reasons for not requesting autopsies.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are very grateful to the National Technological of Mexico for supporting this work. Also, this research was sponsored by the National Council of Science and Technology (CONACYT), as well as by the Public Education Secretary (SEP) through PRODEP.

References

[1] J. Hurtado de Mendoza Amat, Autopsia. Garantía de calidad en la Medicina, La Habana: Editorial Ciencias Médica, 2009.

[2] E. Rubio Delgado, L. Rodríguez-Mazahua, S. G. Peláez-Camarena, J. Antonio Palet Guzmán, and A. López-Chau, "Preliminary results of an analysis using association rules to find relations between medical opinions about the non-realization of autopsies in a Mexican hospital," Second Internationl Work on Intelligent Decision Support System for Industry, Research in Computing Science 132, 23-32.

[3] E. Rubio Delgado, L. Rodríguez-Mazahua, S. G. Peláez-Camarena, J. Antonio Palet Guzmán, and A. López-Chau, "Association Analysis of Medical Opinions About the Non-realization of Autopsies in a Mexican Hospital," in *New Perspectives on Applied Industrial Tools and Techniques*, Management and Industrial Engineering, pp. 233–251, Springer International Publishing, Cham, 2018.

[4] J. Sanz-Ortiz, M. Mayorga, A. Marti, and A. Martı, "Autopsy in clinical oncology: Is it in crisis?" *Medicina Clínica*, vol. 137, no. 7, pp. 317–320, 2011.

[5] D. E. Suleiman, "Reviving hospital autopsy in Nigeria: An urgent call for action," *Annals of Nigerian Medicine*, vol. 9, no. 2, pp. 39-40, 2015.

[6] U. Bieri, H. Moch, S. Dehler, D. Korol, and S. Rohrmann, *Changes in autopsy rates among cancer patients and their impact on cancer statistics from a public health point of view: a longitudinal study from 1980 to 2010 with data from cancer registry zurich*, Springer-Verlag, Heidelberg, Germany, 2015.

[7] A. Turnbull, M. Osborn, and N. Nicholas, Hospital autopsy: Endangered or extinct? http://group.bmj.com, 1-4, 2015.

[8] H. Henshaw, L. Sharkey, D. Crowe, and M. Ferguson, The death of autopsy? Retrieved from The lancet: http://www.thelancet.com/, 2016.

[9] A. P. Qasim, K. U. Hashmi, M. Ahmad, and K. Naheed, "The value of autopsy in medical education: student's attitudes and opinion," *JUMDC*, pp. 17–25, 2015.

[10] E. A. Oviedo Carrascal, A. I. Oviedo, G. Vélez Saldarriaga, and G. Vélez Saldarriaga, "Minería de datos: aportes y tendencias en el servicio de salud de ciudades inteligentes," *Revista Politécnica*, vol. 11, no. 20, pp. 111–120, 2015.

[11] B. Cao, X. Kong, C. Kettering, P. Yu, and A. Ragin, "Determinants of HIV-induced brain changes in three different periods of the early clinical course: A data mining analysis," *NeuroImage: Clinical*, vol. 9, pp. 75–82, 2015.

[12] V. Vemulapalli, J. Qu, J. M. Garren et al., "Non-obvious correlations to disease management unraveled by Bayesian artificial intelligence analyses of CMS data," *Artificial Intelligence in Medicine*, vol. 74, pp. 1–8, 2016.

[13] B. J. Lee and J. Y. Kim, "Indicators of hypertriglyceridemia from anthropometric measures based on data mining," *Computers in Biology and Medicine*, vol. 68, no. 16, pp. 1756–1764, 2016.

[14] Y. Hayashi and S. Yukita, "Rule extraction using Recursive-Rule extraction algorithm with J48graft combined with sampling selection techniques for the diagnosis of type 2 diabetes mellitus in the Pima Indian dataset," *Informatics in Medicine Unlocked*, vol. 2, pp. 92–104, 2016.

[15] A. K. Uysal, "An improved global feature selection scheme for text classification," *Expert Systems with Applications*, vol. 43, pp. 82–92, 2016.

[16] G. Mujtaba, R. G. Raj, L. Shuib, R. Rajandram, and K. Shaikh, "Automatic text classification of ICD-10 related CoD from complex and free text forensic autopsy reports," in *Proceedings of the 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016*, pp. 1055–1058, December 2016.

[17] G. Mujtaba, L. Shuib, R. G. Raj, R. Rajandram, K. Shaikh, and M. A. Al-Garadi, "Automatic ICD-10 multi-class classification of cause of death from plaintext autopsy reports through expert-driven feature selection," *PLoS ONE*, vol. 12, no. 2, Article ID e0170242, 2017.

[18] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.

[19] P. Miasnikof, V. Giannakeas, M. Gomes et al., "Naive Bayes classifiers for verbal autopsies: Comparison to physician-based classification for 21,000 child and adult deaths," *BMC Medicine*, vol. 13, no. 1, article no. 286, 2015.

[20] G. Sumalatha and D. Muniraj, "Survey on medical diagnosis using data mining techniques," in *Proceedings of International Conference on Optical Imaging Sensor and Security*, Coimbatore, India, 2013.

[21] R. Sharma, S. Narayan, and S. Khatri, "Medical data mining using different classification and clustering techniques: a critical survey," in *Proceedings of Second International Conference on Computational Intelligence & Communication Technology*, IEEE, 2016.

[22] S. Hwang, D. Kwak, H. Kim, Y.-B. Park, and W.-S. Cha, "Association rule mining in korean herbal prescriptions of the early 20th century," *Integrative Medicine Research*, vol. 4, no. 1, 107 pages, 2015.

[23] R. Idoudi, K. S. Ettabaa, B. Solaiman, and K. Hamrouni, "Ontology Knowledge Mining Based Association Rules Ranking," *Procedia Computer Science*, vol. 96, pp. 345–354, 2016.

[24] V. Tang, S. W. Cheng, K. L. Choy, P. K. Siu, G. T. Ho, and H. Y. Lam, "An intelligent medical Replenishment System for managing the medical resources in the healthcare industry," 154-161, 2016.

[25] J. Muangprathub, Y. Jareonsuk, and A. Sealiw, "A web-based medical diagnostic system using data mining technique," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 8, no. 6, pp. 37–41, 2016.

[26] A. Franco Pérez and E. León Guzmán, "An approach to the risk analysis of diabetes mellitus type 2 in a health care provider entity of Colombia using business intelligence," in *Proceedings of the 6th Euro American Conference on Telematics and Information Systems (EATIS '12)*, do. Rogerio Patricio Chagas Nascimento, Ed., pp. 49–56, ACM, New York, NY, USA, 2012.

[27] A. A. Dange and S. Siddiqui, "Survey on Assess Co-Morbid Risk of Diabetes Mellitus by using Split and Merge Association Rule Summarization Techniques," *International Journal of Advance Scientific Research and Engineering Trends*, vol. 1, no. 6, pp. 136–140, 2016, http://ijasret.com/VolumeArticles/FullTextPDF/51_ijasret7773.pdf.

[28] D. Antonelli, E. Baralis, G. Bruno et al., "MeTA: Characterization of medical treatments at different abstraction levels," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 4, article no. 57, 2015.

[29] C.-W. Cheng, N. Chanani, K. Maher, and M. D. Wang, "IcuARM-II: Improving the reliability of personalized risk prediction in pediatric intensive care units," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM BCB 2014*, pp. 211–219, New York, NY, USA, September 2014.

[30] G. Duan, D. Ding, Y. Tian, and X. You, "An improved medical decision model based on decision tree algorithms," in *Proceedings of the IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)*, 2016.

[31] S. H. Song, Y. Choi, and T. Yoon, "Comparison of episodes of mosquito borne disease: Dengue, yellow fever, west nile, and filariasis with decision tree, apriori algorithm," in *Proceedings of the 18th International Conference on Advanced Communications Technology, ICACT 2016*, pp. 455–458, kor, February 2016.

[32] Y. Cho, Y. Ahn, S. Yoon, J. Kwon, and T. Yoon, *Analysis of Anticancer Cytokines by Apriori Algorithm, Decision Tree, and SVM*, BigComp. IEEE, 2015.

[33] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington, DC, 1993.

[34] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*, Elsevier, 2012.

[35] T. Scheffer, Finding Association Rules that Trade Support Optimally Against Confidence, 424-435, 2001.

[36] P. A. Flach and N. Lachiche, "Confirmation-guided discovery of first-order rules with Tertius," *Machine Learning*, vol. 42, no. 1-2, Article ID 279321, pp. 61–95, 2001.

[37] P. Felgaer, "Optimización de redes bayesianas basado en técnicas de aprendizaje por inducción. reportes técnicos en ingeniería del software," *Reportes Técnicos en Ingeniería del Software*, vol. 6, no. 2, pp. 64–69, 2004.

[38] S. I. Mariño and P. L. Alfonzo, "Simulación del razonamiento en el proceso de identificación botánica basado en redes bayesianas," *Investigacion Operativa*, vol. 24, no. 39, pp. 55–72, 2016.

[39] T. R. Patil and M. S. Sherekar, "Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification," in *Proceedings of the International Journal Of Computer Science And Applications*, vol. 6, pp. 256–261, 2013.

[40] H. Ibrahim, W. Yasin, N. I. Udzir, and N. A. W. A. Hamid, "Intelligent cooperative web caching policies for media objects based on J48 decision tree and Naïve Bayes supervised machine learning algorithms in structured peer-to-peer systems," *Journal of Information and Communication Technology*, vol. 15, no. 2, pp. 85–116, 2016.

[41] D. López, J. Hernández, and E. Rivas, "Algorithm and software based on multilayer perceptron neural networks for estimating channel use in the spectral decision stage in cognitive radio networks," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 12, pp. 1973–1977, 2016.

[42] D. D. Castillo, R. M. Pérez, L. H. Pérez et al., "Algoritmos de aprendizaje automático para la clasificación de neuronas piramidales afectadas por el envejecimiento," *Revista Cubana de Informática Médica*, vol. 8, no. 3, pp. 559–571, 2016.

Hindawi
Scientific Programming
Volume 2017, Article ID 1496104, 9 pages
https://doi.org/10.1155/2017/1496104

Research Article

Scalable Parallel Distributed Coprocessor System for Graph Searching Problems with Massive Data

Wanrong Huang, Xiaodong Yi, Yichun Sun, Yingwen Liu, Shuai Ye, and Hengzhu Liu

School of Computer, National University of Defense Technology, Deya Road No. 109, Kaifu District, Changsha, Hunan 410073, China

Correspondence should be addressed to Wanrong Huang; huangwr1990@163.com

Received 2 May 2017; Accepted 20 November 2017; Published 19 December 2017

Academic Editor: José María Álvarez-Rodríguez

The Internet applications, such as network searching, electronic commerce, and modern medical applications, produce and process massive data. Considerable data parallelism exists in computation processes of data-intensive applications. A traversal algorithm, breadth-first search (BFS), is fundamental in many graph processing applications and metrics when a graph grows in scale. A variety of scientific programming methods have been proposed for accelerating and parallelizing BFS because of the poor temporal and spatial locality caused by inherent irregular memory access patterns. However, new parallel hardware could provide better improvement for scientific methods. To address small-world graph problems, we propose a scalable and novel field-programmable gate array-based heterogeneous multicore system for scientific programming. The core is multithread for streaming processing. And the communication network InfiniBand is adopted for scalability. We design a binary search algorithm to address mapping to unify all processor addresses. Within the limits permitted by the Graph500 test bench after 1D parallel hybrid BFS algorithm testing, our 8-core and 8-thread-per-core system achieved superior performance and efficiency compared with the prior work under the same degree of parallelism. Our system is efficient not as a special acceleration unit but as a processor platform that deals with graph searching applications.

1. Introduction

Information technology, the Internet, and intelligent technology have ushered in the era of big data. Data-intensive applications, as a typical representative of big data applications represented by graph searching, have been receiving increased attention [1]. Many real-world applications could be abstracted as a large graph of millions of vertices, but this procedure is a considerable challenge for processing. These applications represent the connections, relations, and interaction among entities, such as social networks [2], biological interactions [3], and ground transportation [1]. Poor data-driven computation, unstructured organization, irregular memory access, and low computations-to-memory ratio are the prime reasons for parallel large-graph processing inefficiency [4]. To traverse larger graphs caused by data-intensive applications, a variety of scientific programming methods has been proposed [5, 6]. Tithi et al. [5] optimized the programme and used dynamic load balancing with Intel click++ language. Chen et al. [6] proposed a new parallel model called Codelet model. They all do a good job in speeding up access to memory. However, new parallel computing machines could provide a better platform for software methods. Heterogeneous processing, with reconfigurable logic and field-programmable gate array (FPGAs) as an energy efficient computing systems [7], performs competitively with the multicore CPUs and GPGPUs [4, 8]. The performance of breadth-first search (BFS) on large graphs is bound by the access to high-latency external memory. Thus, we designed considerable parallelism and relatively low clock frequencies to achieve high performance and customized memory architecture to deal with irregular memory access patterns.

The bottleneck of processing graph search is memory. Communication is a primary time overhead in the expansion of processors. In this study, we propose a scalable and novel FPGA-based heterogeneous multicore system for big data applications. The core is multithread for streaming processing, and the communication network is InfiniBand (IB) for scalability. The address mapping is a binary search algorithm mapping, and three levels of hierarchy of memory exist.

The remainder of this paper is organized as follows: Section 2 introduces the details of our 1D decomposition hybrid BFS algorithm. Section 3 shows the details of the proposed parallel system architecture. Section 4 describes the implementation of binary search address mapping. Section 5 provides details of the single processor architecture. Section 6 exhibits the three-level memory hierarchy. Section 7 analyzes the experiment results.

2. Related Works

While parallel computers with millions of cores are already in production, the trend is geared toward higher core densities with deeper memory hierarchies [10] even though other node resources (e.g., memory capacity per core) are not scaling proportionally [11]. Anghel et al. (2014) [12] analyzed node-to-node communications and showed that the application runtime is communication bound and that the communication makeup is as much as 80% of the execution time of each BFS iteration [13].

The Graph500 benchmark is the representative of the graph-based analytic class of applications and is designed to assess the performance of supercomputing systems by solving the BFS graph traversal problem [14, 15].

Fast graph traversal has been approached from a range of architecture methods. Fast graph traversal has been approached from a range of architecture methods. In general-purpose CPU and multicore/supercomputing approaches [16, 17], Agarwal et al. performed locality optimizations on a quad-socket system to reduce memory traffic [18]. A considerable amount of research on parallel BFS implementations on GPUs focuses on level-synchronous or fixed-point methods [19, 20]. The reconfigurable hardware approach in solving graph traversal problems on clusters of FPGAs is limited by graph size and synthesis times [4, 8]. Betkaoui et al. (2012) [4] and Attia et al. (2014) [8] explored highly parallelized processing elements (PEs) and decoupled computation memory. Umuroglu et al. (2015) [11] demonstrated the density, rather than the sparsity, of the treatment of the BFS frontier vector in yielding simpler memory access patterns for BFS, trading redundant computation for DRAM bandwidth utilization, and exploring faster graphs.

3. 1D BFS Algorithm for Testing

In 2013, Beamer et al. [21] proposed a bottom-up algorithm on BFS, which dramatically reduces the number of edges examined, and presented the combination of a conventional top-down algorithm and a novel bottom-up algorithm. The combined algorithm provides a breakthrough for level-synchronized parallel BFS in parallel computation, and this novel bottom-up algorithm is applied in 2D sparse matrix partitioning-based solutions. Today, the 2D bottom-up BFS implementation is a general application in Blue Gene architecture.

Experiments show that with a large number of processors relative to the 1D decomposition, the 2D decomposition can effectively reduce the total communication between processors. In the 2D decomposition, the BFS algorithm has better performance. By contrast, with a small number of processors, the BFS algorithm is suitable in the 1D decomposition. Moreover, our system has eight processors in parallel with unified fine-grained address mapping. Algorithm 1 uses 1D decomposition-optimized BFS algorithm which is proposed by Yasui et al. [22]. In Algorithm 1, V is the set of vertex in graph, while E is the set of edges; that is, $E(u, v) = 1$ means u and v are connected. The parent $[k]$ gives the parent of vertex k in the BFS tree whose source vertex is s; when k is unreachable from s, parent $[k] = -1$. $v \in V$; if v is in the frontier queue, then next $[v] = 1$. The same meaning is given to next (v) (next frontier for each BFS iteration) and visit (v) (when visit $(v) = 1$, v has been visited). Array next, visit, and frontier are stored as bitmap. A new vertex appears in the search; then end $= 0$. When no new vertex appears in the current iteration of BFS, the iteration will end.

4. Massive Parallel Coprocessor System Architecture

Our massive parallel coprocessor system architecture is organized by a single master processing node and large numbers of coprocessing nodes for special computation tasks. The master processing node is an embedded system with ARM processor as its core. The communication architecture of our system is the IB communication network.

The coprocessor is a development board with FPGA (Virtex-7), which is a reconfigurable processor for solving graph problems. Two blocks of DDR3 memory are integrated on each board, and data are transferred by a memory controller (MC). We modified the MC's IP core so that two blocks of DDR3 memory could be accessed in parallel. Any processing node would assign the tasks and transfer data to all coprocessors through the I/O interface and target channel adapter (TCA), which is the communication interface we implement based on the IB protocol. Communications data from the TCA are sent to the IB switch interface through a transmitter (TX) by the IB protocol, and the communication data from IB switch interface are received by TCA through the receiver (RX). The max theoretical line rate is 13.1 Gb/s, and the actual line rate is 10 Gb/s. We have four lines; thus, the communication bandwidth is 40 Gb/s. When the system is initialized, the master node distributes data to the DDR3 memory of each coprocessor via PCI-E bus. After the system has started, each processing node communicates through the IB communication network whose interface is TCA. The program in the master node sends its instructions or data after address mapping (i.e., AM in Figure 1) and each coprocessor communicates after the address mapping. Address mapping is implemented by the FPGA, and the scheme is a functional hardware unit for each node. The architecture is described in Figure 1.

The core is a streaming processor that uses a multithreading vector. Cross-multithreading is a fine-grained multithreading in which threads are executed alternately. Our massive parallel coprocessor system is a scalable system and a platform for parallel processing of big data applications.

```
     Input: V[1 ··· n], E[1 ··· n][1 ··· n], s (source vertex)
     Output: parent[1 ··· n].
(1)  for ∀v ∈ V do
(2)      visit[v] = 0
(3)      parent[v] = −1
(4)      frontier[v] = 0
(5)      next[v] = 0
(6)  frontier[s] = 1
(7)  level = 0
(8)  end = 0
(9)  while fend ≠ 1 do
(10)   end = 1
(11)   if (level < α) or (level > β) then
(12)     for ∀v ∈ V do
(13)       if frontier[v] = 1 then
(14)         for ∀u ∈ E(v) do
(15)           if visit[u] = 0 then
(16)             visit[u] = 1
(17)             next[u] = 1
(18)             parent[u] = v
(19)             end = 0
(20)   else
(21)     for ∀v ∈ V do
(22)       if visit[v = 0] then
(23)         for ∀u ∈ E(v) do
(24)           if frontier[u] = 1 then
(25)             visit[v] = 1
(26)             next[v] = 1
(27)             parent[v] = u
(28)             end = 0
(29)             BREAK
(30)   BARRIER
(31)   for ∀v ∈ V do
(32)     frontier[v] = next[v]
(33)     next[v] = 0
(34)   level = level + 1
(35)
```

Algorithm 1: Parallel 1-D BFS algorithm.

5. Binary Search Address Mapping Unit

The architecture of our address mapping is shown in Figure 2. In our scheme, the memory of DDR3 is divided into two areas: the local data blocks and the global translation blocks. The local data blocks store the data that the program needs from I/O requests and TCA. The global translation blocks hold the mapping of all data. Furthermore, the global translation blocks in each node are the same, and they are managed in a fine-grained page.

The basic idea of binary search is as follows: In ascending order of the tables, we take intermediate records as objects of comparison. If the given item is equal to the intermediate records, then the search is done. However, if the given item is smaller than the intermediate records, then we have a binary search in the first half of the ascending table; otherwise, we have a binary search in the bottom half of the ascending table.

The implementation of the binary search address mapping is a pipeline in which the virtual address is the input, and the output data are the physical address. We divided the registers in the pipeline storage unit into three groups. The first group contains the OMR, DVR, and MTR register, which stores the status of data in RAM. The RAM stores the address mapping of the visited arrays (visited array in the BFS algorithm). The input of the virtual address is the frontier arrays (frontier array in the BFS algorithm). This situation means that we could not find the corresponding mapping in RAM and we would obtain one of the address mappings of the frontier array from the DDR3 memory to be stored in RAMs. This situation is object missing (array missing) where no object is missing in the beginning of a binary search and after the first stack, but the data are missing. The range of virtual address in the array would be entered in the range register (RR). When the input address is not in the range of each RR, a warning that an object is missing is triggered. Then, we would update the data in RR; the order of updating uses the least recently used mechanism. The pipeline then stalls, and we acquire one dataset in the frontier array to continue the

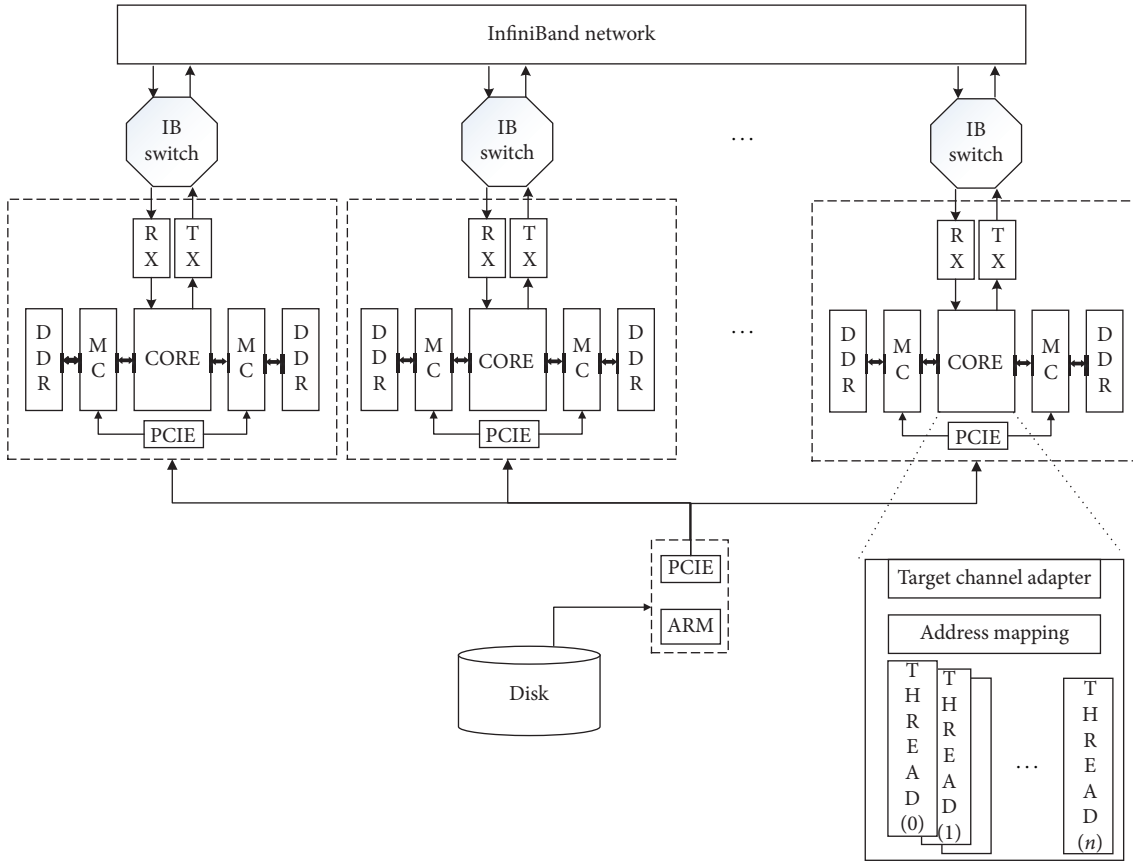


Figure 1: The overview of the system.

search. One data item of the frontier array exists in the current stack of RAMs. Thus, when searching the next stack, the next data in RAMs are invalid. We would obtain the address mapping data of the Frontier array from the DDR3 memory to be stored in this RAMs, and the situation is that the data are missing. The ith OMR is the object missing indicator of the ith stack in pipeline. Furthermore, the MTR is the data missing indicator. The 1 bit in DVR indicates the 64 bits of data in the RAMs that are currently stacked. When an object is missing, the DVR is 0. After updating in RAMs, the corresponding bit would be set to 1.

The second group contains the VAR and DVR, which are temporary storage that passes the data to the next stack when the pipeline is running. The third contains the LBR, RBR, and CAR. In the binary search algorithm, in each searching step, the smaller value is in the LBR, and the larger value is in the RBR. The output of the RAMs provides the intermediate element, and CAR records the comparable results in all steps. For instance, in a kth stack, when the virtual address is in the range of LBR and output of RAMs, the kth bit in CAR is 1 and the value of $(k-1)$th to 1 bit is the same as the value in CAR in the $(k-1)$th stack. The number of stacks is equal to the number of processors.

The LRU unit is implemented by a series of shift registers (the number of bits width is the number of shift register minus 1). When RR is updated, we set 1 to the corresponding shift register, and each register shifts left. The register, which is 0, is the least recently used. The LRU unit provides the number, and the corresponding RR is waiting to be updated in the next object missing.

Unlike time complexity $O(n)$ in direct search, the time complexity of binary search is $O(\log n)$. When the value of n increases, the advantages of the method are obvious.

6. Architecture of Streaming Processor

Our streaming processor design is on the basement of the multithreading vector. Cross-multithreading is a fine-grained multithreading, in which threads are executed alternately. This design requires the switching of threads in each clock whether the thread is stalling or not. This mechanism ensures that the pipeline constantly runs. When a stall exists in a thread, the latter would initially have a request. Every time the thread turns to use the pipeline, it would wait until the stall is handled. The execution of the thread using a cycle mode and its architecture is shown in Figure 3.

The architecture of the microprocessor without interlocked piped stages (MIPS) was simplified to efficiently process graph searching problems. The pipeline was divided into Thread Select (TS), Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), and Write Back (WB) sections.

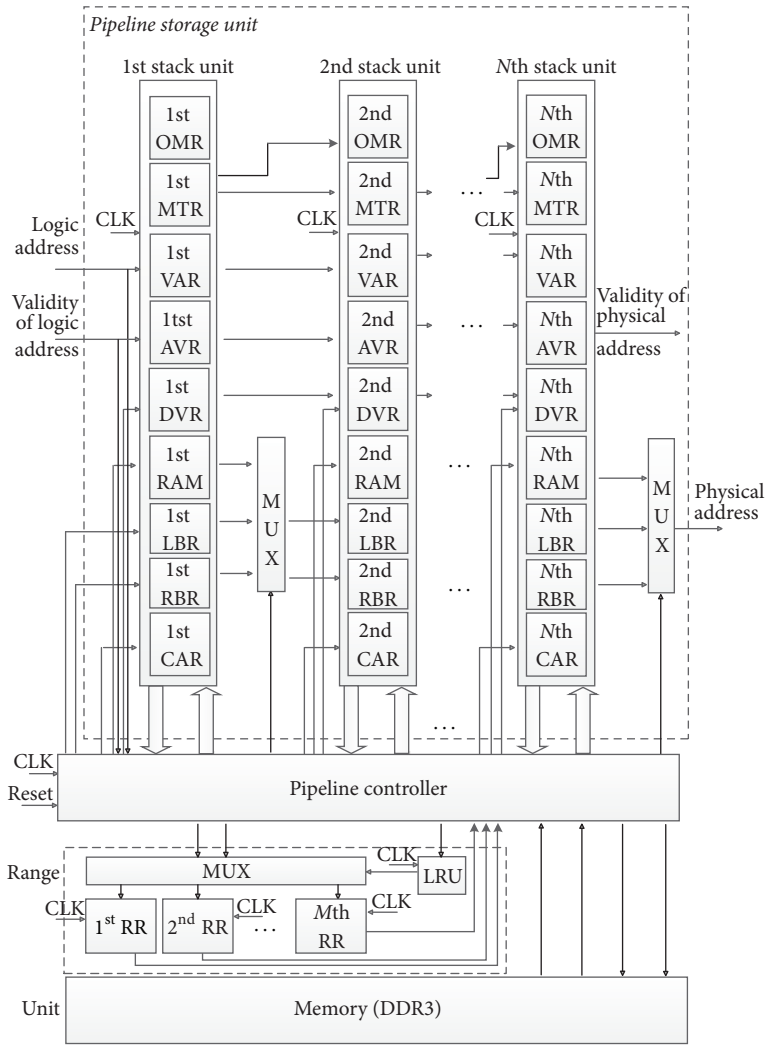


Figure 2: Implementation of binary search address mapping.

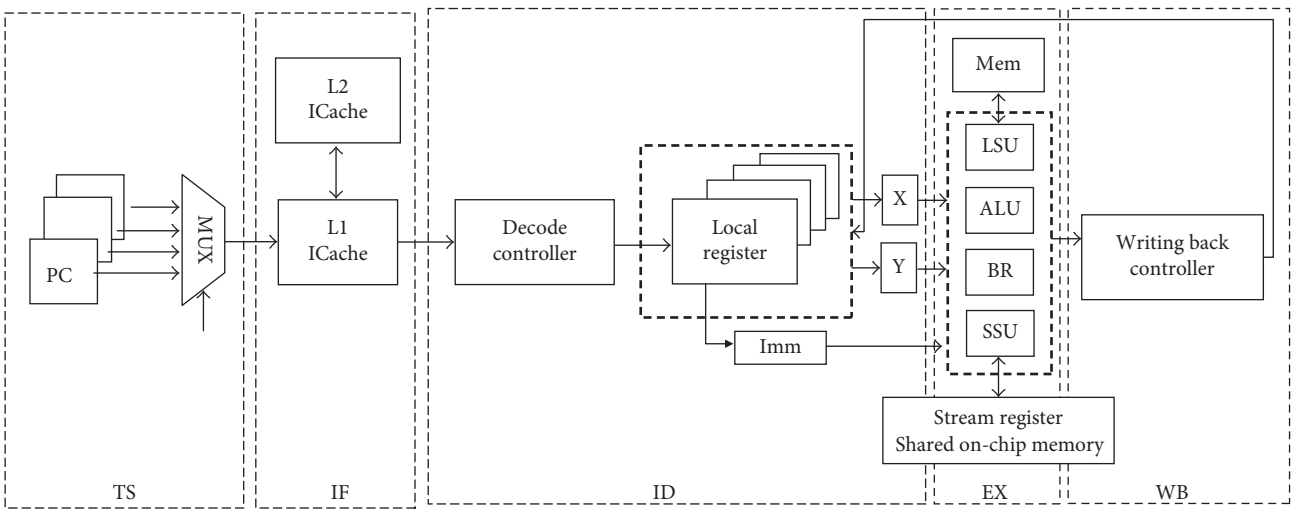


Figure 3: The architecture of streaming processor.

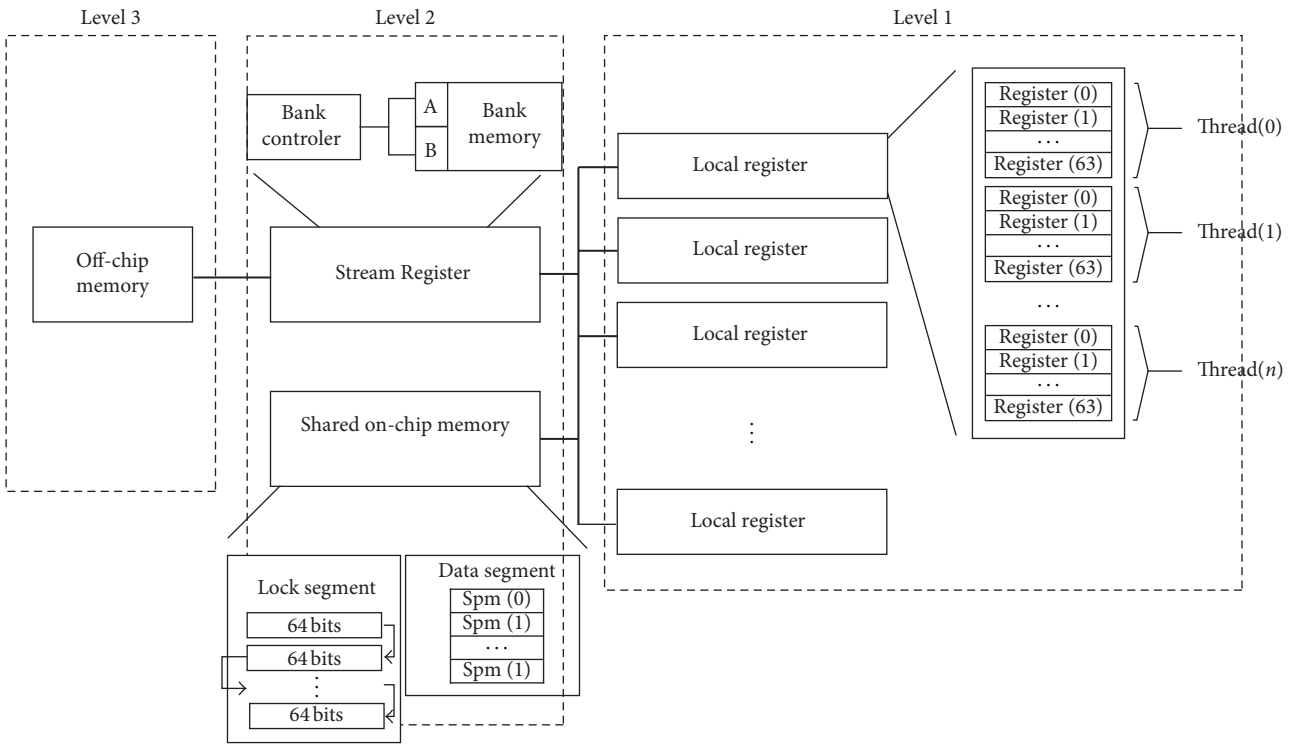


Figure 4: Three-level memory hierarchy.

In the Thread Select section, a thread would be selected and the value of the corresponding program counter (PC) register is obtained as the address in the IF section. Each thread has its corresponding PC register and the value in the register is updated according to the Next Program (NPC, in EX section). The update must be done before another thread is selected. Eight threads are implemented and used according to the instruction register in the cycle because of the limitation of the FPGAs resources.

In the Instruction Fetch section, the pipeline obtained the correct instruction according to the corresponding PC value. Two instruction caches exist: L1 and L2 caches. The L1 cache is private for a process, and the L2 cache is shared given more than one process. The L1 cache is 0.5 kB, and it adopts fully associative mapping programs. The L2 cache is 4 kB and adopts direct mapping programs. An instruction register is designed for each thread to store the last instruction. When the thread is blocked, it could take the last instruction from the instruction register.

In the Instruction Decode section, the decode controller matches the instruction and reads the correct data in the data register.

Four types of operation exist in the Execute section: the operation of arithmetic logic, load and store, access in stream register and shared on-chip memory, and branches jump instruction. In the Write Back section, the result in the Execute section is written in the local register. The result could be from LSU (load and store unit), ALU, and SSU (stream register and shared on-chip memory unit). The address is in the instruction.

Load and store unit (LSU) is designed for operation access, which includes vector or scalar access. The load operation is from the memory to the stream register. The store operation is from the stream register to the memory. Shared memory on-chip unit (SSU) is to perform operation accessed stream register or shared on-chip memory. The structures of LSU and SSU are similar.

7. Three-Level Memory Hierarchy

The stored data in the graph search problems has two characteristics: sparsity of the storage and lack of locality in memory accessing. The optimization in cache cannot effectively use the locality of the memory access. By drawing on the experience of the design of stream architecture, a new three-level memory hierarchy is proposed.

Figure 4 shows that the first-level memory is a local register. Our architecture is a multicore and multithread structure. The local register distributes in the inner part of processor. When multithreads exist in the execution in each process, the processor needs to protect the state. A distributed thread registers to create each thread with its own private registers to store their own states. As they have a local register, no access conflicts occurred between the threads. Implement register mapping is unnecessary. Thus, a small access in the address space and low access delay and power exist.

The local register is implemented by a block RAM resource in FPGAs. In the multithread execution, the local register would simultaneously process two requests of reading from decoding and one request of writing from writing back.

Considering that a single block RAM has read-and-write ports, we copy a local register for each thread. When two requests for reading exist, they could be read separately; however, if one request for writing exists, they would be written simultaneously.

The second level is the stream register and the shared memory on the chip. The stream register is a data buffer on the chip that is implemented by block RAM and combined with a bank of stream register to support multirequest from the local register. A bank of stream register contains a bank memory and a controller (as shown in Figure 4). The bank memory provides two read-and-write ports (ports A and B), and they are the interfaces of the local register and the off-chip memory.

The controller of the stream register has three functions. One is handling the data and the request from the local register. Another is handling the data and request from off-chip memory. The last is coordinating the read-and-write ports that consume the data writing or reading in the stream register with a correct sequence.

The shared on-chip memory is a special RAM on a chip. Furthermore, it is controlled through software and is programmable, which is different from cache. The data is accessed by addresses, and it is shared by processors (if we have) and threads. This study's design of a shared on-chip memory is a sharing and communicating interface for the processors and threads. The function of the design is data transaction and synchronization. The shared on-chip memory is divided with the lock and data segments. When more than one processor or thread exists to read and write in the same address, the atomic lock is supported by the lock segment to ensure correct program sequence. A base address exists in the lock segment. The bits begin with the base address, and one bit determines an atomic lock. Each process or thread could read only one bit in a lock segment at a time. The data segment is special shared data in a multithread or multiprocess program execution. Barrier synchronous counter is one of the special shared data items. The data segment is divided into several 64 bits block, and one local register is composed of 64 bits.

Owing to the width in a transaction burst at 512 bits, the width between the interface of the off-chip memory and the stream register is also 512 bits. The data width between the local register and stream register is 64 bits, and it is the same as the data in the local register. The valid data transaction between the stream and local registers is identified by a mask. The mask is given by a program instruction and its range is from 0 to 7.

The third-level memory hierarchy is the off-chip memory. Our off-chip memory is a dual-data rate (DDR3) and is provided by FPGAs.

8. Results and Comparison

The proposed scalable parallel distributed coprocessor system has 8 cores, and each core has 8 threads. We used the 8 Xilinx Virtex-7 FPGA VC709 evaluation board (xc7vx485t-2ffg1761) and a commercial switchboard called Mellanox IS5030, which is based on the IB protocol, to implement the system. The Xilinx Virtex-7 FPGA VC709 evaluation board has 2 SODIMM DDR3 memory with a storage capacity of 4 GB. Eight channel PCIE interfaces and four line GTH transceivers are included. The communication bandwidth is 40 GB/s, and the memory bandwidth in each core is 10 GB/s in theory. Moreover, two computers that run Linux are used. One of them is responsible for the initialization of the switchboard, and the other is responsible for generating the BFS algorithm, loading data, and receiving returned results. The number of nodes in the system can be expanded as needed. We use the Verilog HDL to achieve a parallel architecture system in Xilinx Vivado 2013.4, which is written to the FPGA chip through the JTAG interface.

In accordance with the Graph500 benchmark, we generated a series of information through a Kronecker graph generator. Then, the information is converted to any type of data structure, which is the input of the BFS algorithm. We verified the results after execution. In the above steps, the creation of the data structure and the design of the BFS algorithm can be customized by the user.

In the Graph500, a fair comparison of the processor with different test bench is obtained using TEPS. According to the performance, which is calculated in Graph500, we proposed a formula to calculate performance P. The details are shown in (1). When the dataset and the root node are determined, E is a constant.

E is the number of edges in the connected region of the root node in the graph. f is the working frequency, which is 200 MHz in our implementation. T_{clk} indicates the number of clock cycles between the beginning and the end of the program. We obtain the T_{clk} through the chip scope. The test dataset in Graph500 is used. Table 1 presents our performance and comparison.

$$P = \frac{E \times f}{T_{\text{clk}}}. \tag{1}$$

We run the parallel BFS algorithm described as Algorithm 1 on our prototype system for testing. The scale of graph searching using the BFS is 19 to 23, which means that the scale of graph data is 2^{19} to 2^{23}, and the edge factor of the graph is 16.

In the first experiment step, we use Vivado 2014.1 to load the test data to FPGAs with a computer. The test data is from Graph500. Then we run the BFS programme in Linux which is running in ARM cortex. The ARM cortex is provided by the evaluation board. The ARM cortex initializes the searching and gets the results from FPGAs through PCIE bus. Finally we use ChipScope which is in Vivado 2014.1 to analyze the performance.

As most works targeting high-performance BFS use MTEPS as a metric, comparing raw traversal performance is possible but the available memory bandwidth in the hardware platform sets a hard limit on achievable BFS performance. Our experimental results are from a Virtex-7 platform with much less (utilization of bandwidth is 64%, the theoretical bandwidth is 10 GB/s, the actual bandwidth is 6.4 GB/s) memory bandwidth and work frequency (200 MHz) than platforms in prior work; thus, it is comparatively slow. Our

Table 1: Comparison to prior work.

Work	Platform	No. of parallel units	Avg. MTEPS	BW (GB/s)	MTEPS/BW
[4]	Convey HC-2	512	1600	80	20
[4]	Convey HC-2	256	980	80	12.5
[4]	Convey HC-2	128	510	80	6.375
[4]	Convey HC-2	64	350	80	4.375
[4]	Convey HC-2	32	210	80	2.625
[8]	Convey HC-2	64	1900	80	23.75
[9]	Nehalem + Fermi	32	800	128	6.25
This work	Virtex-7 & InfiniBand	8	169	10	16.9
This work	Virtex-7 & InfiniBand	64	763	80	9.54

system has 8 cores and supports 8 threads per core, which is equivalent to 64 threads in parallel. Considering the memory bandwidth, the traversals per unit bandwidth is used as a metric to enable fair comparison with prior work. Table 1 allows the comparison with several related works on the average performance, available memory bandwidth, and traversals per bandwidth over RMAT graphs.

9. Conclusions

We can draw the following conclusions:

(1) Compared with the approach of Betkaoui et al. [4], our system is more efficient. The performance of 64 parallel units is similar to that of approximately 256 parallel units in [4]. Our data of traversals per unit bandwidth in 64 parallel units are between 256 units and 128 parallel units in Betkaoui et al. [4].

(2) Attia et al. [8] is on BFS algorithm optimization, and our system is a scalable general processor platform that performs instruction set decoding and the address mapping. Attia et al. [8] is limited to scale, and it is a special acceleration unit.

(3) Our data of traversals per unit are twice that of Hong et al. [9], and the performance is approximately equal. Moreover, our proposed system has the advantages of power and scalability.

(4) The proposed system can be used as a scalable general processing system for graph application with big data.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (no. 61602496).

References

[1] C. Demetrescui, A. Goldberg, and D. Johnson, "The Shortest Path Problem: Ninth DIMACS Implementation Challenge, Proceedings.dimacs Book.ams," in *Proceedings of the The Shortest Path Problem: Ninth DIMACS Implementation Challenge, Proceedings.dimacs Book.ams*, p. 4, 2006.

[2] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2, Article ID 026113, pp. 1–26113, 2004.

[3] D. Bu, Y. Zhao, L. Cai et al., "Topological structure analysis of the protein-protein interaction network in budding yeast," *Nucleic Acids Research*, vol. 31, no. 9, pp. 2443–2450, 2003.

[4] B. Betkaoui, Y. Wang, D. B. Thomas, and W. Luk, "A reconfigurable computing approach for efficient and scalable parallel graph exploration," in *Proceedings of the 2012 IEEE 23rd International Conference on Application-Specific Systems, Architectures and Processors, ASAP 2012*, pp. 8–15, Netherlands, July 2012.

[5] J. J. Tithi, D. Matani, G. Menghani, and R. A. Chowdhury, "Avoiding locks and atomic instructions in shared-memory parallel BFS using optimistic parallelization," in *Proceedings of the 2013 IEEE 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum, IPDPSW 2013*, pp. 1628–1637, USA, May 2013.

[6] C. Chen, S. Koliai, and G. Gao, "Exploitation of locality for energy efficiency for breadth first search in fine-grain execution models," *Tsinghua Science and Technology*, vol. 18, no. 6, Article ID 6678909, pp. 636–646, 2013.

[7] A. Putnam, A. M. Caulfield, E. S. Chung et al., "A reconfigurable fabric for accelerating large-scale datacenter services," in *Proceedings of the ACM/IEEE 41st International Symposium on Computer Architecture (ISCA '14)*, pp. 13–24, IEEE, Minneapolis, Minn, USA, June 2014.

[8] O. G. Attia, T. Johnson, K. Townsend, P. Jones, and J. Zambreno, "CyGraph: A reconfigurable architecture for parallel breadth-first search," in *Proceedings of the 28th IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2014*, pp. 228–235, usa, May 2014.

[9] S. Hong, T. Oguntebi, and K. Olukotun, "Efficient parallel graph exploration on multi-core CPU and GPU," in *Proceedings of the 20th International Conference on Parallel Architectures and Compilation Techniques, PACT 2011*, pp. 78–88, USA, October 2011.

[10] R. Pearce, M. Gokhale, and N. M. Amato, "Scaling techniques for massive scale-free graphs in distributed (external) memory," in *Proceedings of the 27th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2013*, pp. 825–836, USA, May 2013.

[11] Y. Umuroglu, D. Morrison, and M. Jahre, "Hybrid breadth-first search on a single-chip FPGA-CPU heterogeneous platform,"

in *Proceedings of the 25th International Conference on Field Programmable Logic and Applications, FPL 2015*, UK, September 2015.

[12] A. Anghel, G. Rodriguez, and B. Prisacari, "The importance and characteristics of communication in high performance data analytics," in *Proceedings of the 2014 IEEE International Symposium on Workload Characterization, IISWC 2014*, pp. 80-81, USA, October 2014.

[13] A. Amer, H. Lu, P. Balaji, and S. Matsuoka, "Characterizing MPI and hybrid MPI+threads applications at scale: Case study with BFS," in *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015*, pp. 1075–1083, China, May 2015.

[14] M. Anderson, "Better benchmarking for supercomputers: The usual yardstick is not a good metric," *IEEE Spectrum*, vol. 48, no. 1, pp. 12–14, 2011.

[15] A. D. Bader, J. Berry, S. Kahan, and R. Murphy, "The graph 500 list," 2010, http://graph500.org.list.

[16] R. Berrendorf and M. Makulla, "Level-synchronous parallel breadthfirst search algorithms for multicore and multiprocessor systems," in *Proceedings of the the Sixth Intl. Conf. on Future Computational Technologies and Applications*, vol. 26, 2014.

[17] M. Ceriani, G. Palermo, S. Secchi, A. Tumeo, and O. Villa, "Exploring manycore multinode systems for irregular applications with FPGA prototyping," in *Proceedings of the 21st Annual International IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM 2013*, p. 238, USA, April 2013.

[18] V. Agarwal, F. Petrini, D. Pasetto, and D. A. Bader, "Scalable graph exploration on multicore processors," in *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2010*, USA, November 2010.

[19] M. Bisson, M. Bernaschi, and E. Mastrostefano, "Parallel distributed breadth first search on the Kepler architecture," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 2091–2102, 2015.

[20] U. A. Acar, A. Charguéraud, and M. Rainey, *Fast parallel graph-search with splittable and catenable frontiers, Inria*, 2015.

[21] S. Beamer, K. Asanović, and D. Patterson, "Direction-optimizing breadth-first search," *Scientific Programming*, vol. 21, no. 3-4, pp. 137–148, 2013.

[22] Y. Yasui, K. Fujisawa, and K. Goto, "NUMA-optimized parallel breadth-first search on multicore single-node system," in *Proceedings of the 2013 IEEE International Conference on Big Data, Big Data 2013*, pp. 394–402, USA, October 2013.

Hindawi
Scientific Programming
Volume 2017, Article ID 1083062, 10 pages
https://doi.org/10.1155/2017/1083062

Research Article

Design and Solution of a Surrogate Model for Portfolio Optimization Based on Project Ranking

Eduardo Fernandez,[1] Claudia Gómez-Santillán,[2] Laura Cruz-Reyes,[2] Nelson Rangel-Valdez,[2,3] and Shulamith Bastiani[2,4]

[1] *Faculty of Civil Engineering, Autonomous University of Sinaloa, Calz. de las Americas Nte., S/N, Cd. Universitaria, 80013 Culiacán Rosales, SIN, Mexico*
[2] *Postgraduate & Research Division, National Mexican Technology/Madero Institute of Technology, Juventino Rosas 206, 89440 Ciudad Madero, TAMPS, Mexico*
[3] *CONACyT and Postgraduate & Research Division, National Mexican Technology/Madero Institute of Technology, Ciudad Madero, TAMPS, Mexico*
[4] *National Mexican Institute of Technology/Tijuana Institute of Technology, Blvd. Alberto Limón Padilla y Av. ITR Tijuana, S/N, Mesa Otay, 22500 Tijuana, BC, Mexico*

Correspondence should be addressed to Claudia Gómez-Santillán; cggs71@hotmail.com

Received 14 June 2017; Revised 1 November 2017; Accepted 16 November 2017; Published 7 December 2017

Academic Editor: Giner Alor-Hernández

Characterizing the preferences of a decision maker in a multicriteria decision is a complex task that becomes even harder if the information available is limited. This paper addresses a particular case of project portfolio selection; in this case, the measures of project impacts are not assumed, and the available information is only projects' ranking and costs. Usually, resource allocation follows the ranking priorities until they are depleted. This action leads to a feasible solution, but not necessarily to a good portfolio. In this paper, a good portfolio is found by solving a multiobjective problem. To effectively address such dimensionality, the decision maker's preferences in the form of a fuzzy relational system are incorporated in an ant-colony algorithm. The Region of Interest is approached by solving a surrogate triobjective problem. The results show that the reduction of the dimensionality supports the decision maker in choosing the best portfolio.

1. Introduction

Resource allocation in institutions should address the proper distribution of a given budget among a set of available projects [1]. The type of projects can vary according to the area, and they might be involved in a wide range of tasks or activities, such as improving the skills of a professional sports team, the selection of R&D project portfolios in enterprises, the founding of projects by a government program, and supporting environmental regulations [2].

The construction of the best portfolio that accomplishes a certain balance among the selected projects and that is subject to a budget has been approached in the scientific literature (e.g., [3–8]). This problem can be defined as follows:

$$\max_{x \in R_F} \left\{ \left\langle z_1(x), z_2(x), \ldots, z_p(x) \right\rangle \right\}, \quad (1)$$

where R_F is the space of feasible portfolios and $z(x) = \langle z_1(x), z_2(x), \ldots, z_p(x) \rangle$ represents the functions z_i that characterize the impact of a portfolio x over the considered criteria.

Problem (1) has evolved in different particular cases. For example, recently, the *Portfolio Selection Problem on a Set of Ordered Projects*, or PSPSOP, has been reported to focus on the construction of a portfolio from a set of projects subject to a limited budget (cf. Bastiani et al. [9]). This case also involves a distinctive feature, which is that the only information available about the projects is their rank; that is, they are ordered according to the *decision maker's* (DM) preferences. The importance of the study of this problem arises from the fact that in most situations a DM prefers simple decision methods, and the decision process in such

methods has conditions that involve limited resources such as time and information.

The task in PSPSOP is closely related to R&D projects funded by enterprises; it can be seen in the discussion presented in Cooper et al. [10]. Particularly, in government organizations, R&D projects pursue the satisfaction of the citizenship, and, for this purpose, they try to identify programs that focus on the priorities of social sectors. The *participatory budgeting* aids in the definition of such priorities, and it combines the efforts of both citizens and government to establish them (cf. Fernandez and Olmedo [11]). With a proper definition of priorities, all that remains is the construction of a portfolio in which the costs of the projects adjust to the approved budget. This last task remains a challenge in the general case, and it has been studied from different approaches in the literature.

The construction of portfolios from projects that include information on ranking and costs is a problem that has been addressed in the scientific literature. The Ranking method is commonly used, and it involves the ordered selection of projects, previously ranked by priorities established by a DM, according to an available budget. This method has been criticized because the final portfolio that it creates only guarantees that the most important projects will be supported without considering a balance between the priorities and costs, which usually increases the number of supported projects and their impact in society (cf. [12, 13]). Additionally, it could be possible that a DM becomes reluctant to follow such a construction if he considers that the ranking information is not reliable [14].

Bastiani et al. [9] propose a method to solve PSPSOP based on the cardinality and the discrepancies present in a portfolio, where *cardinality* refers to the total number of projects involved in it, and the term *discrepancy* is a concept that reflects the negative effect that is applied over the DM's thinking because one of the projects, when it is compared against others, seems to have merits that belong to the portfolio but it is not in it. The strategy balances the priorities and the number of projects in the final portfolio through a model that minimizes discrepancies concerning the ranking and costs and maximizes its cardinality, all of which are defined in ten objectives. Given that this model handles a large number of objective functions, this paper proposes a method to reduce the original many-objective problem to a surrogate one with three-objective functions. The reduced problem can be easily solved by an ant-colony algorithm that incorporates a knowledge mechanism (like the one proposed by Cruz et al. [15]).

This paper is organized into six sections. Section 2 presents a criticism of previous related approaches and describes the many-objective optimization model proposed by Bastiani et al. [9] for PSPSOP. Section 3 details the decision-support mechanisms used for the PSPSOP; this mechanism is based on DM preferences. Section 4 describes the metaheuristic optimization method. Finally, the experiments and results and conclusions are shown in Sections 5 and 6, respectively.

2. A Brief Outline of Previous Approaches

According to the reviewed scientific literature, there are different approaches that solve PSPSOP with a lack of available information. The works of [16–18] are similar to Scoring and Ranking methods [10] or additive functions [12, 19] in that they prioritize projects according to a certain utility function to measure their importance. Alternatively, the use of proxy variables [9, 11, 14] has offered versatile and satisfactory results that extend the information derived from a ranking of projects. However, none of those approaches offer a strategy that incorporates the DM's preferences, the key element in the present work that guides the construction of better solutions. Table 1 analyzes the state of the art related to the research proposed in these works. The first column presents the work, and the second column describes it.

From the information provided in Table 1, all the strategies followed to solve PSPSOP are based on ranking, costs, discrepancies, and cardinality. Our interest is focused on the definition and management of such elements given by the work of Bastiani et al. [9]. The remainder of this section briefly discusses their definition, but it is left up to the reader whether to get a deeper understanding of the concepts by reviewing the work in [9].

The model proposed by Bastiani et al. [9] defines three objectives per category plus one for the power indicator P. Hence, the number of proxy variables is $M + 1$, where M is the number of categories in the instance of PSPSOP that is solved. The model uses a reference portfolio C_{ref} to reflect the DM's disappointment in combination with the statement "$\text{rank}(x)$ *is better than* $\text{rank}(y) \Rightarrow (y \in C_{\text{ref}} \Rightarrow x \in C_{\text{ref}})$" to define three types of discrepancies over a built portfolio C, which are initially based on the idea that a discrepancy occurs when in such a portfolio $a \in C_{\text{ref}}$ but $a \notin C$. These discrepancies are as follows: (a) weak discrepancy n_{wk} that occurs when the budget of a project $x \in C_{\text{ref}}$, which is not in C, is much higher than the average budget in category k; (b) strong discrepancy n_{sk} that occurs when the budget of a project $x \in C_{\text{ref}}$, which is not in C, is considerably higher than the average budget in category k; and (c) unacceptable discrepancy that occurs whenever a project $x \in C_{\text{ref}}$, which is not in C, has a budget that is not significantly higher than the average budget in category k, or the budget required by another project $y \in C$ that has a lower rank than x. The first two discrepancies, in combination with the cardinalities per category and the power of the set of objectives to the model, are proposed in [9]. The unacceptable discrepancy is used to constrain the feasible region of PSPSOP.

The model of Bastiani et al. [9], referred to as Bastiani's model in this work, succeeds in providing more information to a DM related to the construction of a particular portfolio. The success is derived from the specialization per category of the information derived from the ranking. However, due to this situation, the number of objectives involved in the model is not fixed, and it varies with the number of considered categories. This situation allows growth in the dimensionality, which in turn increases the difficulty in approximating the Region of Interest, that is, the zone in the Pareto frontier that fits a DM's preferences. For this reason, the question of

Table 1: Optimization models for PSPSOP based on proxy variables.

Research work	Description
Fernandez et al. [14]	To the best of our knowledge, this is the first published article that employs a set of proxy variables to solve PSPSOP. The model proposed works on discrepancies D_S, D_W and cardinality N_C of portfolios C; it is as follows: $\min_{C\in R_F}(D_S, D_W)$, $\max_{C\in R_F} N_C$.
Fernandez and Olmedo [11]	This work also uses the cardinality of the portfolio n_{pr} but it extends the types of discrepancies used in [14] to include the set of *absolute discrepancies* n_a, which considers the absolute difference between the considered categories. The model is as follows: $\max_{C\in R_F}\{\langle n_{pr}\rangle\}$, $\min_{C\in R_F}\{n_a, n_s, n_w\}$.
Bastiani et al. [9]	This work improves the multicriteria description of the portfolio qualities, provided by approaches such as [11, 14], through the incorporation of cardinality and discrepancy information per category. Additionally, it includes the *power score* P, an indicator of the number of high-rank projects in a portfolio. The proposed model is as follows: $\operatorname{Optimize}_{C\in R_F}\{\langle N_1, n_{\mathrm{wd}1}, n_{\mathrm{sd}1}, \ldots, N_M, n_{\mathrm{wd}M}, n_{\mathrm{sd}M}, P\rangle\}$, where the power P and cardinalities $N_1, N_2, \ldots, N_M$ are maximized and the discrepancies are minimized.

whether it is possible to keep the same level of information in the construction of a portfolio through an optimization approach but reduce the dimensionality remained open. Our work in this paper is dedicated to answering this question positively. The proposed approach is presented in the following section.

3. Proposed Surrogate Model for PSPSOP

According to Cruz et al. [15], metaheuristic approaches are a viable strategy to solve multiobjective optimization problems. These approaches rely on their ability to approximate the Pareto frontier to provide a set of solutions (or populations) that might satisfy complex constraints and objectives functions, a feature rarely shared with approaches based on mathematical programming, which normally yield a single solution and are limited to a narrower range of variations in the definition of an optimization problem.

During the optimization, it is possible to find some pitfalls derived from a problem with many objectives. One of the pitfalls arises if it is difficult to generate a proper population that lies on the Pareto frontier. Another pitfall corresponds to the increments in the number of *Dominance Resistant Solutions*, that is, solutions that are hard to dominate even though they are not Pareto optimal and are hard to take out of the final set of solutions. Another concern is related to the appropriate selection of one solution from the set given by metaheuristics. This situation leads to a process in which a human must perform a cognitive effort to choose the desired alternative. However, according to Miller [20], the human mind's capacity to handle information is diminished when it increases, for example, when the number of objectives that he/she must use to make a decision increases. All these drawbacks have been observed and/or discussed in several research studies, as shown in [21–24].

The high dimensionality of a problem can be addressed through the incorporation of preference information. According to the work in [15], a DM that is willing to provide his/her preferences aids in the localization of the Pareto frontier zone known as the *Region of Interest* (RoI); this zone can be understood as a relaxation of the feasible region defined by the multiobjective optimization problem that only covers solutions that are of interest for a DM. Several research papers, such as the one by Fernandez et al. [25], present strategies that address multiobjective problems using surrogate models that approximate the RoI. The remaining part of this section presents the adaptation of such model to solve the PSPSOP.

The research work in [25] proposes a surrogate model based on the credibility index $\sigma(x, y)$ of the statement "x is at least as good as y." The value of $\sigma(x, y)$ is computed using the method ELECTRE (cf. [26, 27]), and it is integrated into a relational system of preferences (as described by Roy [26]) to model the DM preferences. The preferences defined in [27] for pairs of alternatives x and y are as follows: (1) strict preference xPy; (2) indifference xIy; (3) weak preference xQy; (4) incomparability xRy; and (5) k-preference xKy.

To define a surrogate model for PSPSOP, this work uses the net flow score defined through the objectives of Bastiani's model, as shown in Table 2. The *net flow score*, denoted $F_n(x)$ or NF, was introduced in Fernandez et al. [25] as a measure to enhance preference information towards a better characterization of the DM preferences in the *nonstrictly outranked set*, denoted by NS. Based on the elements $y \in$ NS,

Input: P_r, B, max_iter, n_a
Output: The new set of portfolios NS
Begin PROCEDURE
(1) Initialize Iter = 0, pheromone matrix and $N_S = \emptyset$
(2) Construct an initial portfolio C_{ref}
(3) **Repeat**
(4) Initialize $S_F = \emptyset$
(5) S_F <- GenerateFeasibleSolutions(P_r, B, n_a)
(6) ${S^*}_F$ <- Perform local search to the set S_F
(7) Calculate objective functions of Problem (2) on the set S_F
(8) F <- Generate non-dominated fronts on ${S^*}_F$
(9) Updating the pheromone matrix with the set F_0
(10) Assign $N_S = N_S \cup F_0$
(11) iter = iter + 1
(12) **until** (iter = max_iter)
(13) F^* <- Generate non-dominated fronts on N_S
(14) **return** ${F_0}^*$
End PROCEDURE

Algorithm 1: ACO-SOP.

Table 2: Computation of the *net flow score*.

Set/measure	Conditions
S	$S(O,x) = \{y \in O \mid yPx\}$.
NS	$\text{NS}(O) = \{x \in O \mid S(O,x) = \emptyset\}$.
W	$W(O,x) = \{y \in \text{NS}(O) \mid yQx \vee yKx\}$
Net flow score (NF)	$F_n(x) = \sum_{y \in \text{NS}(O)\setminus\{x\}} [\sigma(x,y) - \sigma(y,x)]$, where $F_n(x) > F_n(y)$ denotes a certain preference of x over y
F	$F(O,x) = \{y \in \text{NS}(O) \mid F_n(y) > F_n(x)\}$

the following sets are defined: (a) the set S of alternatives y that strictly outrank x; (b) the set W of portfolios y that weakly outrank a given portfolio x; and (c) the set F of alternatives y with greater net flow score. All the previous elements are defined in Table 2, where O is the set of feasible portfolios.

The combination of the net flow score $F_n(x)$ with the definition of the sets S and W allows the formulation of the optimization problem defined in

$$\min_{x \in O} \{\langle |S(O,x)|, |W(O,x)|, |F(O,x)| \rangle\}. \quad (2)$$

With the previous problem, it is demonstrated that Bastiani's model can be mapped to problem (2) independently of the original objective space dimension. The approach by Fernandez et al. [25] was successfully applied to Problem 1 in [15]. Combined with an ant-colony metaheuristic, the approach in Bastiani et al. [9] is extended here to solve PSPSOP thru the optimization model defined in problem (2). To summarize, the contribution of this work is the solution of PSPSOP with lack of information using a surrogate objective model (which is of smaller dimension than those reported in previous approaches) and the incorporation of DM's preferences in the search process to approximate the RoI. The details of the algorithm and each component of the *ACO-SOP* are presented in the next section.

4. An Ant-Colony Optimization Algorithm

This work proposes an approach based on an ant-colony optimization, denoted by *ACO-SOP (Ant-Colony Optimization for Solving Portfolio Problems with Ordinal Information about Projects)*. The approach takes ideas from Dorigo's ACS [28], Cruz et al. [15], and Bastiani et al. [9] to solve PSPSOP, first searching in the wider feasible region defined by Bastiani's model and then searching in the smaller space defined by the model in problem (2).

The core strategy of ACO-SOP uses the pheromone representation, selection rule, and local search function defined in the algorithm ACO-SPRI proposed by Bastiani et al. [9], with a small variation in the construction of the nondominated fronts used in their computations. The general idea of ACO-SOP is depicted in Algorithm 1. The algorithm is characterized by five elements: (a) the initialization function; (b) the construction of a feasible set of solutions; (c) the improvement phase based on local search; (d) the construction of nondominated fronts; and (e) the updating of the pheromone matrix. These elements are briefly defined in this section (cf. Bastiani et al. [9] for further details).

The algorithm ACO-SOP requires as input the set P_r of ranked projects, the budget B, the maximum number of iterations *max_iter*, and the maximum number of ants n_a. Inside this algorithm, the initialization function constructs the reference portfolio C_{ref} (see Line (2)); it is done by the ordered selection of projects from P_r by rank and in agreement with the available budget B (ties are broken arbitrarily).

The phase of construction of feasible solutions (see Line (5)) involves the construction of one portfolio by each of the n_a ants. Each ant starts with an empty portfolio C and adds to it one project at a time based on the budget, global knowledge

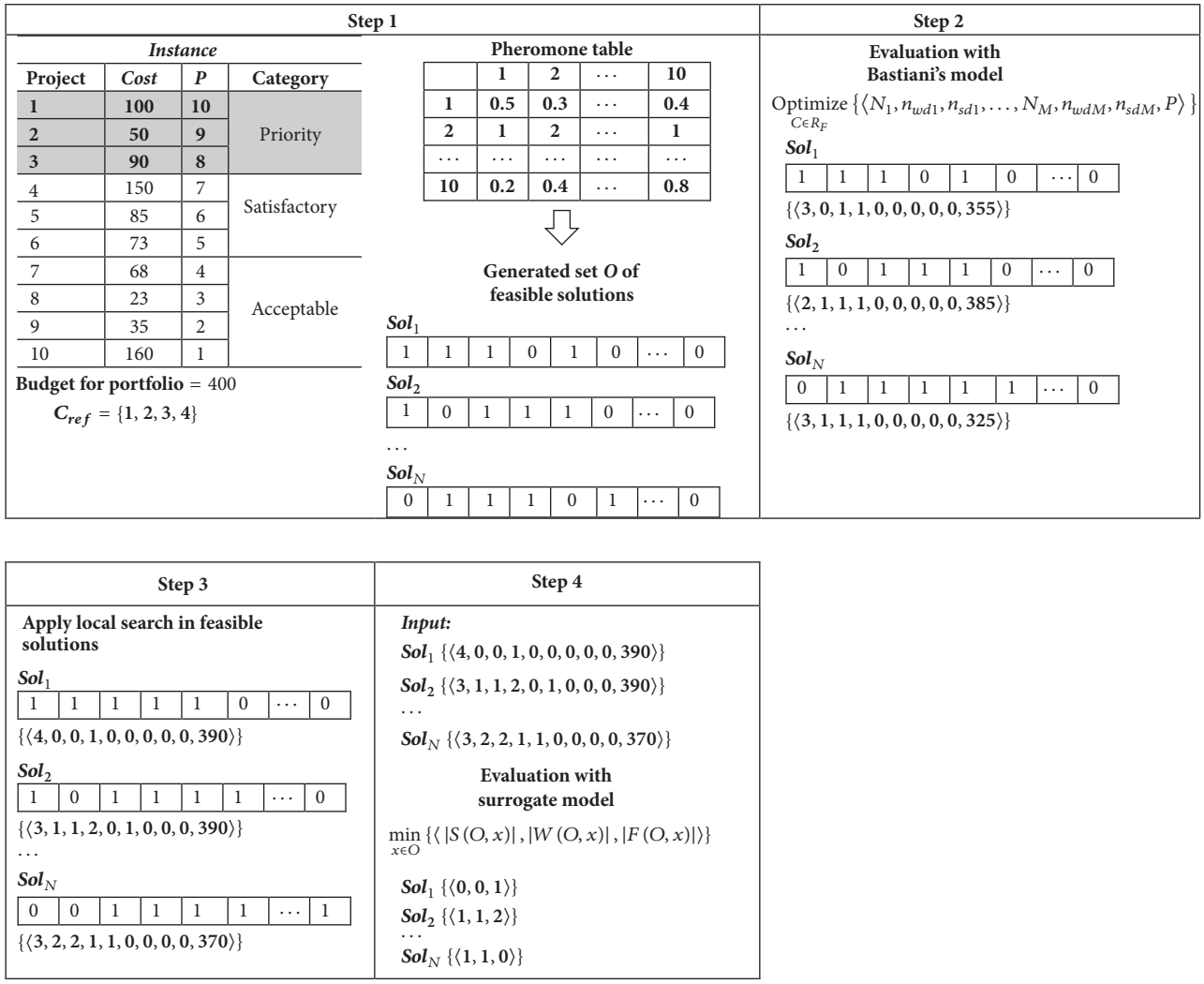


Figure 1: Schematic example of construction process of ACO-SOP.

derived from the pheromone matrix, and local knowledge obtained from the preferences of the DM, as defined in the selection rule by Bastiani et al. [9]. At the end of this phase, a set S_F of feasible portfolios is delivered.

Following the process of forming S_F, the algorithm ACO-SOP performs a local search over each C in S_F, just as it is done by Bastiani et al. [9] in their local search scheme. The algorithm creates a new portfolio C'_{new} with every combination derived from the inclusion and/or exclusion of v projects of P_r chosen at random. Each portfolio C'_{new} is repaired such that constraints in rank and budget are held. Finally, the best portfolio C^*_{new} is chosen as the local search improvement for C. The union of all the new portfolios C^*_{new} will form the new set of feasible portfolios S^*_F.

The construction of nondominated fronts (see Line (8)) is based on the objectives defined in problem (2). The output of the algorithm in this phase is the set of fronts $F = \{F_0, F_1, \ldots, F_k\}$ obtained from S^*_F. Each set F_i is composed of a subset of portfolios in S^*_F that are exclusively dominated by exactly i portfolios, where $0 \leq i \leq k$ (following the dominance criterion established in Cruz et al. [15]). Instead of computing the dominance from the objectives of Bastiani's model, this work uses the objectives of problem (2), a key feature in this research because it makes it more manageable for algorithms to address large dimensions.

Finally, the last and most important element that defines ACO-SOP is the pheromone matrix, and the process to construct it and update it is described herein. The bidimensional matrix τ represents the knowledge that the ants have gained during the construction of portfolios. They represent that knowledge in the form of pairs of projects (i, j) and the gain of having them together in the same portfolio (denoted as $\tau_{i,j}$). The portfolios S^*_F constructed by the ants at each iteration are used to update τ (see Line (9)) based on the number of fronts k constructed and in the front where each portfolio is found. This strategy and the one incorporated to prevent premature convergence are detailed in Bastiani et al. [9].

The algorithm ACO-SOP accumulates in N_S the front F_0 constructed at each iteration (see Line (10)). Then, using the last set N_S it forms the final set of fronts F^* and returns as its solution the front F^*_0. Figure 1 presents a

Table 3: Parameters of the outranking model.

Values of the preference model									
W		Q		P		U		V	
		Indifference thresholds		*Strict preference threshold*		*Pre-veto threshold*		*Veto threshold*	
W_1	0.21	Q_1	0	P_1	1	U_1	2	V_1	3.5
W_2	0.14	Q_2	0	P_2	1	U_2	2.5	V_2	5.5
W_3	0.21	Q_3	0	P_3	1	U_3	1.5	V_3	3
W_4	0.14	Q_4	0	P_4	1	U_4	8	V_4	11.5
W_5	0.05	Q_5	0	P_5	1	U_5	-	V_5	-
W_6	0.12	Q_6	0	P_6	1	U_6	-	V_6	-
W_7	0.04	Q_7	0	P_7	1	U_7	-	V_7	-
W_8	0.01	Q_8	0	P_8	2	U_8	-	V_8	-
W_9	0.04	Q_9	0	P_9	1	U_9	-	V_9	-
W_{10}	0.04	Q_{10}	0.5	P_{10}	1.5	U_{10}	-	V_{10}	-
Credibility threshold $\lambda = 0.67$									
Unacceptability threshold $\gamma = 0.20$									

Note. Veto power is not allowed in criteria 5–10.

brief example of the construction process of ACO-SOP. Step 1 shows how the instance information is transformed with the aid of the pheromone matrix into a set of N feasible solutions $O = \{\text{Sol}_1, \text{Sol}_2, \ldots, \text{Sol}_N\}$, where each solution Sol_i is binary vector representing the projects in a portfolio. Step 2 shows the computed values of the 10 objectives from Bastiani's model for each solution. After that, in step 3, ACO-SOP shows an improved set of N solutions from the local search. Finally, step 4 presents the transformation of O into the surrogate model, that is, a set O with the objective values defined in problem (2). To exemplify the transformation, let us take the values of the solution $x = \text{Sol}_1 = \langle 0, 0, 1\rangle$, which indicate that there is no other solution $y = \text{Sol}_i$, for $i \neq j$, that is strictly or weakly preferred over x; that is, there is no y such that yPx or yQx or yKx, and there is only one solution that has greater net flow score than x; that is, $F_n(y) > F_n(x)$, taking into account the fact that y is in the *nonstrictly outranked set* NS(O).

The following section presents the experimental design performed to evaluate the performance of ACO-SOP when solving problem (2). The following section also shows the versatility of the approach to tackle cases of large dimensions.

5. Solving Some Computer Experiments

In this section, we develop a set of experiments to verify the validity of and to validate the advantages gained by our solution method, especially addressing problems with many objectives. Based on these goals, the exposure and resolution of two different cases of study are presented in the following paragraphs.

To test the performance of ACO-SOP, the algorithm was implemented using the programming language Java. The experimental design was made using a computer with an Intel Core i7 2.8 GHz CPU, 4 GB of RAM, and Mac OS X Lion 10.7.5 (11G63b) as operating system. The instance used for the experimentation is the one defined in Bastiani et al. [9]. The tuning process involved selecting different combinations of values similar to those previously reported for these parameters. According to the results of the fine-tuning process, the best configuration of values was as follows: *tot_iter* = 200 iterations, $n_a = 300$ ants, $\omega = 0.63$, $\eta = 0.1$, $\rho = 0.9$, $\alpha_1 = 0.65$, and $\alpha_2 = 0.75$. These parameter values were used to obtain the experimental results reported in this section. The incorporation of preferences through the outranking model is basic in order to map the problem in Bastiani et al. [9] into problem (2). The parameters of the outranking model were set as provided in Table 3.

The ACO-SOP strategy requires, besides the ranking information and costs of the PSPSOP instance, the definition of a reference portfolio C_{ref} and the definition of the projects' categories. The heuristic followed to construct C_{ref} is based on the traditional Scoring and Ranking method and takes the projects one at a time in order of their rank until the available budget is consumed. The categories considered during the experiment were (1) priority, (2) satisfactory, and (3) acceptable, and the projects were distributed uniformly among them based on their ranks.

5.1. Analysis of Results. The results from the experiments performed on the proposed approach (ACO-SOP) are analyzed from three points of view. The first analysis involves the quality of the solutions provided by ACO-SOP in relation to their meaning to a DM and his/her preferences. The second and third analyses study the performance of ACO-SOP in comparison with a similar approach of the state of the art. These experiments differ on the quality indicator used during the comparison. Let us indicate that the considered budget was 2,500 units and that the reference portfolio C_{ref} is formed by the first 22 best ranked projects in the instance taken from Bastiani et al. [9].

The indicators used to measure the results from the experiment were I_1, I_2, and I_3. The indicator $I_1 = \{|S(O, x)|, |W(O, x)|, |F(O, x)|\}$ uses the three objectives associated with the preference information derived from the proposed surrogate model (see problem (2)). The indicator

Table 4: Nonstrictly outranked solutions.

Solution	I_1						I_2							
x	$\|S(O,x)\|$	$\|W(O,x)\|$	$\|F(O,x)\|$	N_1	n_{wd1}	n_{sd1}	N_2	n_{wd2}	n_{sd2}	N_3	n_{wd3}	n_{sd3}	P	Card
$ACO\text{-}SOP_1$	0	0	0	27	2	0	1	0	0	0	0	0	85.10	28
$ACO\text{-}SOP_2$	0	0	1	27	4	0	6	0	0	0	0	0	81.03	33
$ACO\text{-}SOP_3$	0	2	2	27	2	1	3	0	0	0	0	0	83.5	30

Table 5: Solutions suggested in [11] and the two best solutions to the problem in Bastiani et al. [9].

Solution	N_1	n_{wd1}	n_{sd1}	N_2	n_{wd2}	n_{sd2}	N_3	n_{wd3}	n_{sd3}	P	Card (C)
Sol_1	24	0	0	0	0	0	0	0	0	87.62	24
Sol_2	25	0	0	0	0	0	0	0	0	86.96	25
$ACO\text{-}SOP_1$	27	2	0	1	0	0	0	0	0	85.10	28
$ACO\text{-}SOP_2$	27	4	0	6	0	0	0	0	0	81.03	33

$I_2 = \{N_1, n_{wd1}, n_{sd1}, N_2, n_{wd2}, n_{sd2}, N_3, n_{wd3}, n_{sd3}, P\}$ uses the ten objectives associated with ranking and derived from the Bastiani model (see Section 2). Finally, the indicator I_3 is the content of the portfolio, that is, the projects that are included in it.

The first analysis involved the solution of the instance with ACO-SOP, and its results are summarized in Table 4. This table provides the values of the indicators I_1 and I_2 computed from the three best nonstrictly outranked solutions that were consistently obtained in 30 runs (the average computer time required for this experiment was 130 minutes). The solutions ACO-SOP_1, ACO-SOP_2, and ACO-SOP_3 of Table 4 are ordered according to the indicator I_1; in this ordering, ACO-SOP_1 is the best because it reduced all the values of the indicator to zero, while ACO-SOP_2 did it only for the metrics $|S(O,x)|$ and $|W(O,x)|$ but it improves ACO-SOP_3 in the last one, $|F(O,x)|$. This relation of order is associated with the DM's preferences, and evidence of that can be supported by analyzing the values of the indicator I_2, regarding the number of projects involved in the solutions.

First, it is possible to observe a great level of dissatisfaction in the DM when the provided solutions have strong discrepancies (see value n_{sd1} of ACO-SOP_3), because it means that some important projects that are in the reference portfolio C_{ref} are left out of the solution when they could have been included in place of others that were less relevant. Second, the DM could have a low level of dissatisfaction if the solutions only have weak discrepancies (see value n_{wd1} of ACO-SOP_2), because it means that only a rather small number of important projects had been left out. Third, an almost null level of dissatisfaction can be noted in the DM when the solution involved a small number of weak discrepancies and a high power (see values n_{wd1} and P of ACO-SOP_1), because this is an indicator that an adequate number of high-ranked projects are in the solution.

The second analysis compares the two best solutions ACO-SOP_1 and ACO-SOP_2 against the solutions Sol_1 and Sol_2, as reported in [11]. Table 5 summarizes these results, with each of its columns representing a metric used by indicator I_2. Observe that the lack of discrepancies on Sol_1 and Sol_2 (see n_{wd1}, n_{sd1}, n_{wd2}, n_{sd2}, n_{wd3}, and n_{sd3}) could allow the DM to make a decision using only two criteria, the power P and the cardinality N_1; this is a convenient situation for a DM that willingly accepts a solution with only high-ranked projects, or with a large cardinality, but does not care about their balance (because high values in P are normally associated with small cardinalities and high-ranked but costly projects). But what happens if the DM has a different point of view? For example, how are these solutions affected if, for the DM, the objectives are not equally important? The latter case involves considering solutions, such as ACO-SOP_1 and ACO-SOP_2, with an acceptable level of discrepancies and a better balance between P and the cardinalities (e.g., N_1 and N_2); these solutions slightly increase the discrepancies (as in metric n_{wd1}). Hence, while approaches such as the ones presented in [11] or in Bastiani et al. [9] succeeded in managing problems with limited information, they do not include DM preferences. This analysis showed evidence that the solutions provided by the proposed model allow the inclusion of DM's preferences through the parameters of the outranking model (see Table 3). The preference model can be used to redefine what is acceptable or not as a solution for a DM; for example, the solutions Sol_1 and Sol_2 do not satisfy the DM's preferences in Table 3 because they have unacceptable discrepancies that cause great discomfort to the DM; this situation is detailed in the following analysis.

Table 6 allows the analysis of the configurations of the portfolios obtained from Sol_1, Sol_2, ACO-SOP_1, and ACO-SOP_2; it shows the projects in each solution and in bold those in Category 1. One important difference between these solutions is the situation of Project 20 (or Pj_{20}); while it appears on the portfolio of ACO-SOP_1 and ACO-SOP_2, it does not in Sol_1 and Sol_2. The relevance of this project is that it belongs to the reference portfolio $C_{ref} = \{1, 2, 3, \ldots, 22\}$, and the fact that it does not appear in Sol_1 and Sol_2 constitutes an unacceptable discrepancy for the DM characterized in Table 3.

To explain this situation, let us note that Pj_{20} is in Category 1 and has an average cost of $\text{Cost}(Pj_{20}) = 97.5$ and the projects of Sol_1 that are in the same category

Table 6: Projects of the portfolios created from solutions Sol_1, Sol_2, ACO-SOP_1, and ACO-SOP_2.

Solution	Projects in the portfolio	Avg (Cat_1)
Sol_1	{**1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **10**, **11**, **12**, **13**, **14**, **15**, **16**, **17**, **18**, **19**, **21**, **24**, **25**, **26**, **27**}	102.19
Sol_2	{**1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **10**, **11**, **12**, **13**, **14**, **15**, **16**, **17**, **18**, **19**, **21**, **24**, **25**, **26**, **27**, **28**}	99.55
ACO-SOP$_1$	{**1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **10**, **11**, **12**, **13**, **14**, **15**, **16**, **17**, **18**, **20**, **22**, **25**, **26**, **27**, **28**, **29**, **30**, **33**, 34}	91.29
ACO-SOP$_2$	{**1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **10**, **11**, **12**, **13**, **14**, **15**, **16**, **17**, **18**, **20**, **25**, **26**, **27**, **28**, **29**, **30**, **31**, **32**, **33**, 34, 35, 36, 37, 38, 39}	85.31

have an average cost of $\text{Avg}(\text{Cat}_1) = 102.19$. Hence, the unacceptability level of Pj_{20} is $\text{UNAc}(Pj_{20}) = (\text{Cost}(Pj_{20}) - \text{Avg}(\text{Cat}_1))/\text{Avg}(\text{Cat}_1) = -0.04$. The value of $\text{UNAc}(Pj_{20})$ is smaller than the unacceptability threshold $\gamma = 0.20$ defined for the DM; that is, the projects in the portfolio are too costly to exclude Pj_{20} from C_{ref} and possibly some others, a situation that occurs with Sol_1 and Sol_2 and that generally does not satisfy the DM's preferences. Such solutions where the DM's preferences are not considered can be constructed using approaches from [9, 11], but the present approach overcomes it by including preferences during the search process; as a result, the solutions contain portfolios more convenient for the DM, because they include an increment in the number of projects of up to 20%, with the guarantee that this still satisfies his/her preferences.

To summarize, this section has presented the results from the experiment to analyze the quality of the solutions provided by ACO-SOP and their robustness. The robustness was demonstrated by providing evidence that ACO-SOP provides solutions of the same quality as the other approaches, whenever they satisfy the DM's preferences; evidence of this is provided in Table 4, where solutions of similar quality are presented, solutions that are in the RoI for a DM. In general, whenever a solution fits the interest of the DM expressed in the parameter values of the preference model, this solution can be found in the search space, showing that the present approach maintains the same level of information. Finally, the lack of preferences on approaches such as [11] or [9] can lead to the construction of solutions that are not in the RoI for a DM, that is, solutions the DM is not interested in.

6. Some Conclusions

This paper proposes a knowledge-based decision-support system for the Portfolio Selection Problem on a Set of Ordered Projects. This problem commonly has a lack of available information, which is why the incorporation of knowledge mechanisms in the solution strategies can improve the quality of the solutions.

This work is a crucial refinement from a recent proposal that modeled the DM's attitude through a many-objective optimization problem. The high dimensionality of this problem is a great concern for metaheuristic approaches in generating an acceptable approximation to the Pareto frontier, and it is also a great concern for the DM in making the final decision due to human cognitive limitations. Here, we have presented a method that incorporates the DM's preferences through a fuzzy outranking-based relational model. This allows mapping the original many-objective problem into a surrogate three-objective problem. With this transformation, the process of identifying the Region of Interest (the preference privileged zone of the Pareto frontier) becomes easy. Another advantage of the proposal is to be very robust with respect to an increasing number of objective functions. Robustness is important to make a finer representation of the DM's preferences, including project synergic effects. So this contribution should be significant in this particular field of project portfolio optimization.

The strongly nonlinear surrogate problem is solved by using an ant-colony algorithm. This algorithm includes a knowledge and learning mechanism in the form of a pheromone matrix and integrates a decision-support mechanism based on the DM's preferences. The experimental results show its capacity to obtain consistent solutions in the Region of Interest. These solutions surpass in quality others obtained by alternative proposals taking into account the amount and impact of the supported projects.

Let us observe that the surrogate model not only reduces the dimensionality of the problem but also reduces the number of alternatives that result from the multiobjective optimization process, because it provides only nonoutranked solutions that are on the privileged zone of the known Pareto frontier. Comparisons between pair of alternatives are not so difficult as the number of objectives seems to suggest. Note that the objectives N_1 and n_{sd1} are strongly more important than the remaining ones. So, the DM can apply a sort of lexicographic priority that allows a radical reduction of the set of solutions. Even though it is possible that ACO-SOP produces a number of portfolios that could not be easily handled by the DM, this situation can be addressed by the support of an expert analyst who, using a multicriteria decision method, provides a sufficiently small set of alternatives that can be handled by the DM.

Another point that is worthy of discussion is that the projects might involve objectives that are hard to judge, as is the case of the *risk*, because they are complex in both quantitative and qualitative elements. However, it is important to note that such cases are handled in the present approach through the ranking. Following this idea, let us note that this work addresses the special case of PSPSOP that has as the only available information the ranking of the projects and their costs and that it is the rank that integrates all the previous existing information about the criteria that characterized the projects. In other words, the instances of PSPSOP already capture information about criteria such as risk in the rank, and the methodology followed to do such integration is out of the scope of the present research.

However, if it were the case that an alternative to solve the problem of integrating objectives into ranking should be proposed, it could be done through the family of ELECTRE methods [26], which can address qualitative and quantitative information in a unique aggregation model of preferences.

Finally, some situations that can be present in the instances of the special case of PSPSOP studied here are the existence of interdependence among criteria and/or projects. Fuzzy multicriteria methods derived from outranking approaches can handle interdependence among criteria and reflect it on the rank ordering of candidate projects (cf. [29]). So, the rank used by our proposal takes into account the criterion interdependence. On the other hand, the project interdependence remains a challenge for the proposed model; it is defined as part of the future work of this research to explore the possibility of integrating synergy negative and positive effects among projects through the incorporation of artificial projects that are related to their interdependence.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been partially supported by the following CONACyT projects: (a) Fronteras de la Ciencias Project 1340; (b) Consolidation National Lab Project 280712; (c) Projects 236154 and 269890; (d) Project 280081, Red Temática para el Apoyo a la Decisión y Optimización Inteligente de Sistemas Complejos y de Gran Escala (OPTISAD), Universidad Autónoma de Nuevo León; and (e) Project 3058 from the program Cátedras CONACyT.

References

[1] D. Kleinmuntz, "Resource allocation decisions," in *Advances in Decision Analysis: From Foundations to Applications*, W. Edwards, D. von Winterfeldt, and R. F. Miles, Eds., pp. 400–410, Cambridge University Press, Cambridge, UK, 2007.

[2] A. Salo, J. Keisler, and A. Morton, "An invitation to portfolio decision analysis," in *Portfolio Decision Analysis: Improved Methods for Resource Allocation*, A. Salo, J. Keisler, and A. Morton, Eds., pp. 3–27, Springer Science & Business Media, New York, NY, USA, 2011.

[3] M. A. Coffin and B. W. Taylor III, "Multiple criteria R&D project selection and scheduling using fuzzy logic," *Computers & Operations Research*, vol. 23, no. 3, pp. 207–220, 1996.

[4] J. Klapka, P. Piňos, and V. Ševčík, "Multicriterial Projects Selection," *Intelligent Systems Reference Library*, vol. 38, pp. 245–261, 2013.

[5] C. Stummer and K. Heidenberger, "Interactive R&D portfolio analysis with project interdependencies and time profiles of multiple objectives," *IEEE Transactions on Engineering Management*, vol. 50, no. 2, pp. 175–183, 2003.

[6] J. L. Ringuest, S. B. Graves, and R. H. Case, "Mean-Gini analysis in R&D portfolio selection," *European Journal of Operational Research*, vol. 154, no. 1, pp. 157–169, 2004.

[7] C. Carlsson, R. Fullér, M. Heikkilä, and P. Majlender, "A fuzzy approach to R&D project portfolio selection," *International Journal of Approximate Reasoning*, vol. 44, no. 2, pp. 93–105, 2007.

[8] X. Zhao, Y. Yang, G. Wu, J. Yang, and X. Xue, "A dynamic and fuzzy modeling approach for multiobjective rd project portfolio selection," *Journal of Convergence Information Technology*, vol. 7, no. 1, pp. 36–44, 2012.

[9] S. S. Bastiani, L. Cruz-Reyes, E. Fernandez, and C. Gomez, "Portfolio optimization from a set of preference ordered projects using an ant colony based multi-objective approach," *International Journal of Computational Intelligence Systems*, vol. 8, pp. 41–53, 2015.

[10] R. Cooper, S. Edgett, and E. Kleinschmidt, "Portfolio management for new product development: Results of an industry practices study," *R&D Management*, vol. 31, no. 4, pp. 361–380, 2001.

[11] E. Fernandez and R. Olmedo, "Public project portfolio optimization under a participatory paradigm," *Applied Computational Intelligence and Soft Computing*, vol. 2013, Article ID 891781, 13 pages, 2013.

[12] S. A. Gabriel, S. Kumar, J. Ordóñez, and A. Nasserian, "A multiobjective optimization model for project selection with probabilistic considerations," *Socio-Economic Planning Sciences*, vol. 40, no. 4, pp. 297–313, 2006.

[13] Y. Yang, S. Yang, and Y. Ma, "A literature review on decision making approaches for research and development project portfolio selection," in *Proceedings of the CSAMSE Conference*, 2012.

[14] E. Fernandez, L. F. Felix, and G. Mazcorro, "Multi-objective optimisation of an outranking model for public resources allocation on competing projects," *International Journal of Operational Research*, vol. 5, no. 2, pp. 190–210, 2009.

[15] L. Cruz, E. Fernandez, C. Gomez, G. Rivera, and F. Perez, "Many-objective portfolio optimization of interdependent projects with 'a priori' incorporation of decision-maker preferences," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, pp. 1517–1531, 2014.

[16] S. Altuntas and T. Dereli, "A novel approach based on DEMATEL method and patent citation analysis for prioritizing a portfolio of investment projects," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1003–1012, 2015.

[17] M. Corazza, S. Funari, and R. Gusso, "An evolutionary approach to preference disaggregation in a MURAME-based creditworthiness problem," *Applied Soft Computing*, vol. 29, pp. 110–121, 2015.

[18] M. Collan and P. Luukka, "Evaluating R & D projects as investments by using an overall ranking from four new fuzzy similarity measure based TOPSIS variants," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 505–515, 2014.

[19] G. Mavrotas, D. Diakoulaki, and Y. Caloghirou, "Project prioritization under policy restrictions. A combination of MCDA with 0-1 programming," *European Journal of Operational Research*, vol. 171, no. 1, pp. 296–308, 2006.

[20] G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information," *The Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[21] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: a short review," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 2419–2426, June 2008.

[22] D. W. Corne and J. D. Knowles, "Techniques for highly multiobjective optimisation: Some nondominated points are better than others," in *Proceedings of the 9th Annual Genetic and*

Evolutionary Computation Conference, GECCO 2007, pp. 773–780, New York, NY, USA, July 2007.

[23] J. López, C. A. Coello Coello, K. Oyama, and K. Fujii, "Alternative preference relation to deal with many-objective optimization problems," in *Proceedings of the Evolutionary Multi- Criterion Optimization (EMO 2013)*, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds., Lecture Notes in Computer Science, pp. 291–306, Springer, 2013.

[24] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimisation: an exploratory analysis," in *Proceedings of the Congress on Evolutionary Computation (CEC '03)*, vol. 3, pp. 2066–2073, December 2003.

[25] E. Fernandez, E. Lopez, G. Mazcorro, R. Olmedo, and C. A. Coello Coello, "Application of the non-outranked sorting genetic algorithm to public project portfolio selection," *Information Sciences*, vol. 228, pp. 131–149, 2013.

[26] B. Roy, "The Outranking Approach and the Foundations of ELECTRE methods," in *Readings in Multiple Criteria Decision Aid*, C. Baena e Costa, Ed., vol. 31, pp. 133–161, Springer Science & Business Media, Berlin, Germany, 1990.

[27] J. Figueira, V. Mousseau, and B. Roy, "Electre methods," in *Multiple Criteria Decision Analysis: State of the Art Surveys*, J. Figueira, S. Greco, and M. Ehrgott, Eds., vol. 78 of *International Series in Operations Research & Management Science*, pp. 164–195, Springer, New York, NY, USA, 2005.

[28] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[29] J. R. Figueira, S. Greco, and B. Roy, "ELECTRE methods with interaction between criteria: an extension of the concordance index," *European Journal of Operational Research*, vol. 199, no. 2, pp. 478–495, 2009.

Hindawi
Scientific Programming
Volume 2017, Article ID 7831897, 10 pages
https://doi.org/10.1155/2017/7831897

Research Article

Semantic Annotation of Unstructured Documents Using Concepts Similarity

Fernando Pech,[1] Alicia Martinez,[1] Hugo Estrada,[2] and Yasmin Hernandez[3]

[1]*National Center of Research and Technological Development (CENIDET), Cuernavaca, MOR, Mexico*
[2]*Center for Research and Innovation in Information and Communications Technologies, Ciudad de México, Mexico*
[3]*National Institute of Electricity and Clean Energy (INEEL), Cuernavaca, MOR, Mexico*

Correspondence should be addressed to Fernando Pech; fpech@cenidet.edu.mx

Received 17 June 2017; Revised 2 October 2017; Accepted 8 November 2017; Published 7 December 2017

Academic Editor: José María Álvarez-Rodríguez

There is a large amount of information in the form of unstructured documents which pose challenges in the information storage, search, and retrieval. This situation has given rise to several information search approaches. Some proposals take into account the contextual meaning of the terms specified in the query. Semantic annotation technique can help to retrieve and extract information in unstructured documents. We propose a semantic annotation strategy for unstructured documents as part of a semantic search engine. In this proposal, ontologies are used to determine the context of the entities specified in the query. Our strategy for extracting the context is focused on concepts similarity. Each relevant term of the document is associated with an instance in the ontology. The similarity between each of the explicit relationships is measured through the combination of two types of associations: the association between each pair of concepts and the calculation of the weight of the relationships.

1. Introduction

The rapid growth of the web has generated an enormous amount of information in the form of unstructured documents. Search engines have become common and basic tools for users. However, engines still have difficulties in performing searches because search methods are based on keywords, and they do not capture and do not explore the meaning and context of the need of the user. This challenge has drawn attention of several research groups which are interested in solving the issues associated with information storage and search and retrieval of information in this enormous cumulus of data.

On the other hand, the continuous growth of the Semantic Web has motivated the development of knowledge structures on different domains and applications, like Wikipedia [1], Linked Open Data (LOD) [2], DBpedia [3], Freebase [4], and YAGO [5], among other applications. Additionally, some ontologies for several domains have been developed, such as Snomed CT [6] and UMLS [7] for the medical field and AGROVOC [8] for the agricultural field. An ontology is a formal representation of knowledge, which plays a very important role in the semantic web because of its capability to express meanings and relationships. Ontologies have been valuable in knowledge extraction technologies, especially in the aggregation of knowledge from unstructured documents. Ontologies are a key component of semantic association, which is the process to formalizing knowledge through the linking of words or phrases of plain text (mentions or named entities) with elements of the ontology (concepts or entities).

The semantic annotation of a document consists in finding mappings between text chunks of a document and the instances or individuals in ontology. The annotation plays an important role in a variety of semantic applications, such as generation of linked data, extraction of open information, alignment of ontologies, and semantic search. Specifically, semantic search allows users to express their information needs in terms of the knowledge base concepts. Unlike traditional keyword-based search, semantic search can make use of semantic relationships in the ontology to accomplish new tasks, such as refining user queries with broader or more specific concepts.

The semantic annotation has been applied in different areas of knowledge. For example, it has been applied in biological systems for the identification of biomedical entities such as genes, proteins, and their relationships; also, it has been applied in news analysis for identification of people, organizations, and places.

At the present, semantic annotation strategies are carried out without regard to context [9–11]; these works do not analyze the meaning or semantics of the terms. Generally, authors assume that lexicons are enough to express the meaning of the terms in a document. However, to a large degree, the semantic of a concept depends of the context in which it occurs. Therefore, the identification of meaning could lead to problems of ambiguity. Several research works have demonstrated the complexity of word-sense disambiguation (WSD), where traditionally a term is searched in a data dictionary (e.g., WordNet) [12]. Other approaches have chosen to analyze the context of the terms to improve the annotation process [13]. The problems related to semantic annotation are still an open research topic.

The annotation process could be a source of different types of problems, for example, (i) ambiguous annotations, when entities have been assigned to more than one concept in the ontology, (ii) erroneous annotations, when the meaning of a text is not found in the ontology, and, (iii) false annotations, when the annotation does not provide any value for the realization of a semantic search. In this sense, this paper presents a strategy of semantic annotation in unstructured documents. Our approach is based on ontologies and on the extraction of contextual semantic information from entities of the ontology. The semantic context of an entity is determined by their relationships in the ontology. Therefore, we propose to extract the semantic context of the entities by calculating the similarity of association between each pair of concepts and the calculation of the weights of the relationships of the entities. With this strategy, we deal with the problems of ambiguous, erroneous, and false annotations. Our method of semantic annotation is part of a semantic search system in natural language and it has been evaluated with the corpus compiled by Lee and Welsh [14] and DBpedia.

This paper is organized as follows: Section 2 describes the background of our proposal, Section 3 presents the related work, Section 4 presents the architecture of the system, Section 5 presents the evaluation of the proposed approach, and finally, Section 6 provides some conclusions and an outlook for future work.

2. Foundations

This section presents the concepts and foundations of the proposed semantic annotation approach.

2.1. Ontology. An ontology is composed of a schema and instances (see Figure 1). A schema is defined as $\langle C, D, P\rangle$ where C is the set of classes/concepts $C = \{c_1, c_2, \ldots, c_n\}$, D is the set of data types, and P is the set of properties $P = \{p_1, p_2, \ldots, p_n\}$ which are the relationships between classes. Instances represent knowledge and denote an instanced class and their relationships. Instances can be defined as a graph

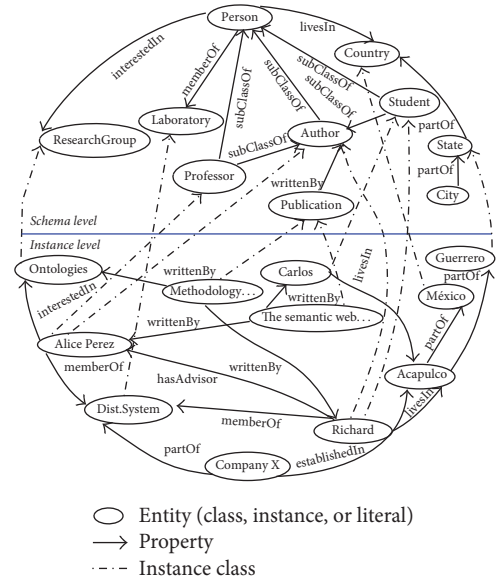


Figure 1: Two-level ontology: schema and instances.

$G = \langle V, E\rangle$, where V is the set of instances, and E is the set of relationships or predicates binding the instances.

In an ontology, classes, properties, data types, and instances are explicitly identified by Uniform Resource Identifiers (URI). In addition, they represent entities within the ontology, which are characterized by their textual description declared in the property *rdfs:label*. This may have lexical variations defined as *rdfs:label* = {"*text1*","*text2*"}.

Figure 1 shows a fragment of an ontology for the research domain. The schema level defines classes such as *Laboratory* and *Professor*, and properties such as *interestedIn*.

The instance level indicates the instantiated schemas. For example, *ontologies* is an instance of the class *ResearchGroup; Methodology,* and *Alice Perez* are related to the property *writtenBy* and belong to the classes *Publication* and *Author*, respectively. The *Acapulco* instance contains its textual description with two lexical variations *rdfs:label* = {"*Acapulco*","*Acapulco de Juárez*"}.

2.2. Semantic Annotation. The semantic annotation is fundamental to obtaining better results in the semantic search because the documents are represented in a conceptual space.

The semantic annotation of a document d consists in linking the terms t in $d = \{t_1, t_2, \ldots, t_n\}$ with the entities in the ontology which describe the content of the term in its textual description best (see Figure 2). Namely, let an entity-term pair be $\langle c, t\rangle$, where c is an entity in the ontology and t is a term/phrase of d, so that there is a mapping between the textual descriptions defined in the label *rdfs:label* of c and t.

In semantic annotation techniques, a document is analyzed in order to identify its relevant terms and to define the importance of each term. There are tools to identify mentions, such as TagMe [15] and Spotlight [16].

When the semantic annotations are made without regard to the context, its terms or mentions are linked with the

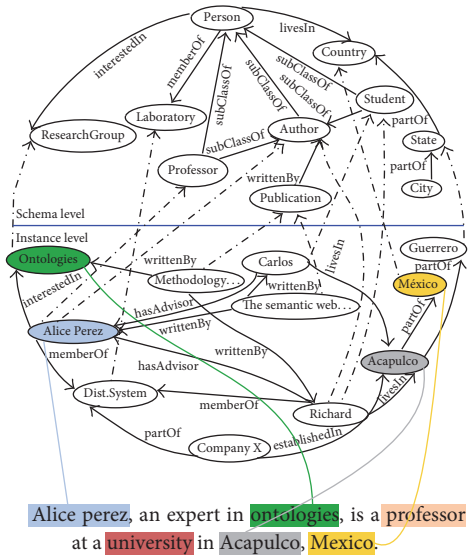


FIGURE 2: Link between terms (mentions) of a document and the ontology entities.

entities in the ontology without taking into account their meaning. This causes ambiguous or erroneous annotations.

Our research work proposes to analyze the context of the annotations in order to identify their meaning through the entities in the ontology, and in this way to avoid ambiguities. In the extraction of the context, the explicit relationships of each entity in the ontology are analyzed. For example, Figure 2 shows the relationship between the *Ontologies* entity and *ResearchGroup* and *Alice Perez*.

3. Related Work

The semantic search involves different components: (i) preprocessing, (ii) semantic query translator, (iii) semantic annotation and indexing, (iv) retrieval of semantic content, and (v) semantic ranking.

Currently, there are several research works with different contributions in the area of the semantic web. Several general-purpose tools have been developed to support the annotation process, and, also, specific domain ontologies and knowledge bases have been proposed by research groups.

General-Purpose Tools. There are several available services for annotation of named entities in documents that could be accessed using RESTful APIs such as the case of OpenCalais [17].

Let us remark that, AlchemyAPI [18] and OpenCalais [17] use context-based statistical techniques to disambiguate the candidate instances to annotate a term. These tools use proprietary vocabularies and ontologies whose instances are linked to DBpedia through the *owl:sameAs* relationship. However, OpenCalais provides some limited linkage to DBpedia. Also, OpenCalais is mainly focused on organizations. This approach has two disadvantages. Firstly, it only explores the surface of the graph for each DBpedia instance considering the labels, abstract, links to Wiki pages, and synonyms. Secondly, this approach annotates a term with only one instance of DBpedia. Therefore, this approach does not exploit the semantic information available in DBpedia to disambiguate the instance annotating a given term.

DBpedia Spotlight [16] is a semantic annotation tool for data entities in a document and it is based on DBpedia for the annotation. Also, this tool provides interfaces for disambiguation, including a Web API which supports XML, JSON, and RFD formats.

Gate [19] is a tool for text engineering to help users in the process of text annotation manually. This tool provides basic processing functionalities, such as recognition of entity named, sentence dividers, markers, and so on.

Ontea [20] is a tool for semantic metadata extraction from documents. This tool uses regular expressions patterns as text analysis tool, and it detects semantically equivalent elements according to the domain ontology defined in the tool. This tool creates a new individual ontology from a defined class and it assigns the detected elements as properties in the ontology class. The patterns of regular expressions are used to annotate the text without format with elements in the ontology.

These approaches have two main drawbacks. On the one hand, they just explore the surface of the graph for each DBpedia instance; they mainly consider label, abstract, links to Wiki pages, and synonyms. Therefore, these approaches do not exploit the semantic information available in DBpedia to disambiguate the instance annotating a given term. Another disadvantage of this work lies in the fact that it discards the relationship, which contains relevant information about a term. That is, they do not enrich the description of relevant terms with the semantic graphs that contain the DBpedia instances related to the context of the document. Some works do face these drawbacks by annotating their documents with graphs extracted from DBpedia.

Specific Domain Tools. There are specific tools for biomedical annotations such as MetaMap [8], Whatizi [21], and Semantator [22]. Most of this approaches and tools are based on a strategy to search terms in thesaurus. These methods consist in finding occurrences of a concept chain in a text fragment using strict coincidence of terms.

Semantic Annotation Approaches Based on Information Retrieval Techniques. Popov and colleagues [23] present KIM, a platform for information and knowledge management, annotation, and indexed and semantic retrieval. This tool provides a scalar infrastructure for personalized information extraction and also for documents management and its corresponding annotations. The main contribution of KIM is the recognition of the named entities according to ontology.

Castells et al. [24] propose an information retrieval model using ontologies for the annotation classification. This model uses an ontology-based schema for the semiautomatic semantic annotation of documents. This research was extended by Fernández et al. [25] to provide natural language queries.

Berlanga et al. [26] propose a semantic annotation/query strategy for a corpus using several knowledge bases. This

method is based on a statistical framework where the concepts of the knowledge bases and the corpus documents are homogeneously represented through statistical models of language. This enables the effective semantic annotation of the corpus.

Nebot and Berlanga [27] explore the use of semantic annotation in the biomedical domain. They present a scalable method to extract domain-independent relationships. They propose a probabilistic approach to measure the synonymy relationship and also a method to discover abstract semantic relationships automatically.

Fuentes-Lorenzo et al. [28] propose a tool to improve the quality of results of the Web search engines, performing a better classification of the query results.

In the literature we can find several approaches to optimize query results. Swoogle [29] is a raster based system to discover, index, and query RDF documents. SemSearch [30] is another search engine relying on semantic indexes and is based on Sesame [31] and Lucene. The ranking algorithm was specifically designed for the extraction of ontologies through annotation. In [32] a search engine is proposed to infer the context of Web pages and also to create links to relevant Web pages. Lopez et al. [33] developed an information retrieval system based on ontologies. This system takes as input a natural language query and converts it to semantic entities using a question-answering system. PowerAqua [33] is a system to recover and to classify documents through TF-IDF measures [34].

4. Semantic Annotation Architecture

This paper presents a novel semantic annotation approach based on ontologies for the improvement of information search in unstructured documents. We present an approach to annotation that enriches and semantically describes the content of a document using the similarity of entities of an ontology. Specifically calculating (1) the association between each concept pair and (2) the relationships weight.

The goals of our approach are (a) to link the entities with their meaning in order to be annotated and (b) to provide a framework for semantic searches using natural language processing. The semantic annotation approach extracts the semantic context through the similarity analysis calculating the association of the explicit relationships and the weight of the relationships of the entities involved. Figure 3 shows an overview of our proposed solution for the semantic annotation.

4.1. Documents Indexing. Commonly, Natural Language Processing (NLP) is used for the analysis of unstructured documents, and also for the recognition and extraction of mentions or named entities [35].

In this approach, the indexing of unstructured web documents generates inverted indexes, which contain the set of terms to be compared with the entities in the ontology. We propose an algorithm for the indexing of documents using Lucene. The output of this algorithm is an inverted index containing the list of terms or keywords and a set of documents where the terms appear.

Therefore, the algorithm provides a mapping from terms to documents and a mechanism for annotating search results. Also, it obtains the position of the information: the list of terms IDs, the association with the ID of the document, and its position.

4.2. Entity Identifications. Given a document d and a knowledge base, the objective of this phase is to extract the textual descriptions and the semantic context of all the information about d from the knowledge base.

Identification of Mentions. Documents are analyzed to detect terms. Generally, this process is known as acknowledgment of mentions or named entities [35]. A mention is a term/phrase in the text which may correspond to an entity in the knowledge base.

From the ontological point of view, an entity can denote classes, relationships, or instances. Entities can represent people, organizations, locations, and so on. There are different tools to define entities, like Spotlight [16] and TagMe [15], among others. TagMe uses Wikipedia as a dictionary of terms for mentions detection. We have used this tool with the same purpose.

TagMe analyzes the input text and detects mentions using a dictionary of entities/words (surface form). For each word, it registers the set of entities recognized by that name. This dictionary is constructed by extracting the words from four sources: Wikipedia papers, redirected pages, Wikipedia page titles, and other variants.

Words with few occurrences and single-character words are discarded. Finally, an additional filtering to discard words with low link probability is done (e.g., less than 0.001). The link probability is defined as stated in

$$p\text{link}(m) = P(\text{link} \mid m)\,\frac{\text{link}(m)}{\text{freq}(m)}, \tag{1}$$

where link(m) is the number of times the mention m appears as a link and freq(m) denotes the number of times the mention m occurs in Wikipedia.

The detection of mentions is carried out by comparing the n-grams (until $n = 6$) of the document.

4.2.1. Extraction of Instances. Each mention detected in document d is searched in the ontology, and if an instance matches its textual description, it is extracted from the label *rdfs:label*. All the values contained in *rdfs:label* (lexical variations) are considered as labels that are later compared in the document index.

Figure 4 shows a fragment of the *México* entity code containing URI, class, and textual description with two lexical variations *México* and *Estados Unidos Mexicanos*.

4.2.2. Extraction of the Instances Semantic Context. In this process, the semantic context of the instances is extracted to be analyzed in detail. The explicit relationships in the *URI* are also analyzed. Several strategies have been proposed to evaluate the proximity of entities according to their semantic characteristics [21]. The use of the semantic measure based on graphs allows us to compare concepts, terms, and instances.

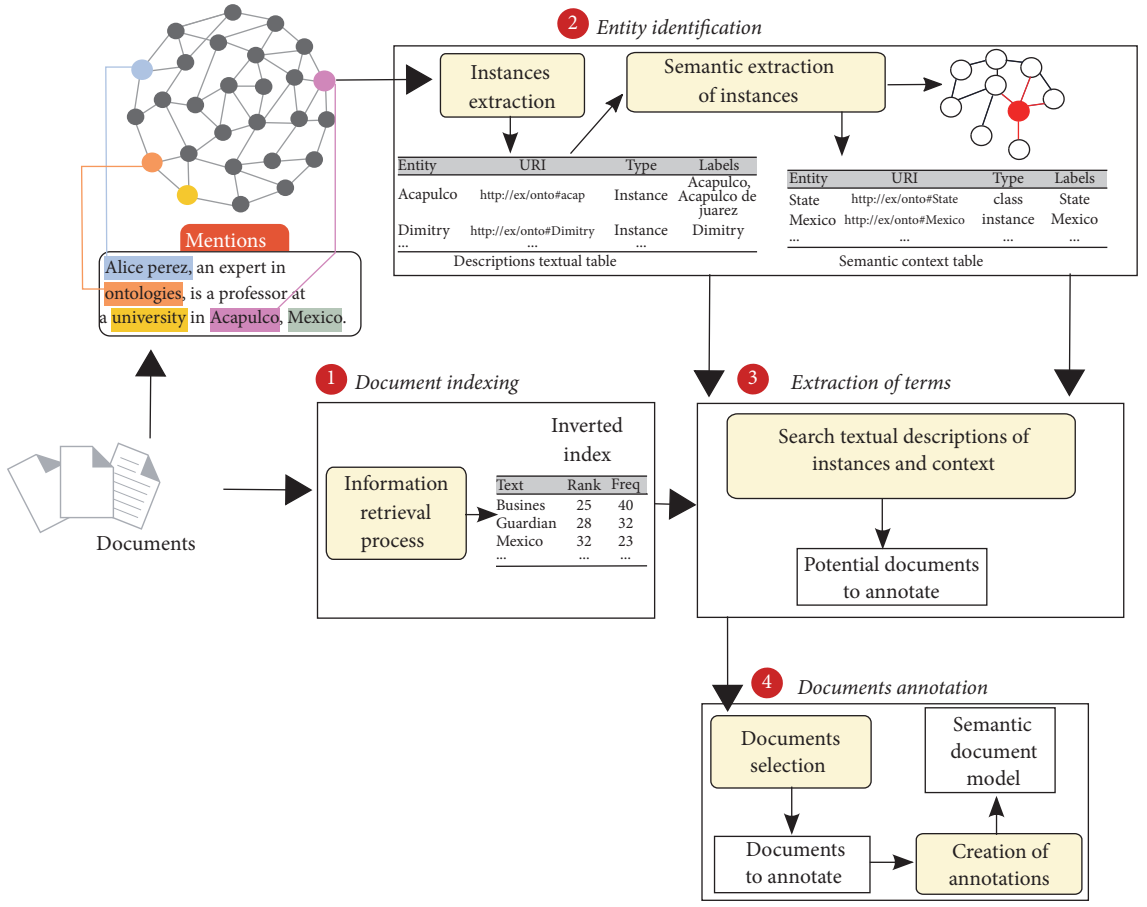


Figure 3: Methodological proposal for semantic annotation process.

```
<owl:NamedIndividual
   rdf:about="http://example/ontea#Mexico">
      <rdf:type rdf:resource="http://example/ontea#Country"/>
      <rdfs:label rdf:datatype="http://www.w3.org/2001/
                                XMLSchema#string">
          Mexico, Estados Unidos Mexicanos
      </rdfs:label>
</owl:NamedIndividual>
```

Figure 4: Code fragment of an instance in the ontology.

This measure is represented as an edge in a semantic graph in order to determine the relationship strength among the ontology concepts.

Therefore, this research work uses the semantic measure as a strategy to measure the strength of the explicit relationship between entities. Two types of measures are considered: (1) the association between each concept pair and (2) the relationships weight. Each measure reflects the similarity degree or relationship between the ontology entities according to its meaning.

Concept Pairwise Association. An entity is explicitly related to other concepts in the ontology. To measure the association strength between each pair of concepts c_1 and c_2, we compare each pairwise by calculating similarity. Figure 2 shows the *Acapulco* entity with four explicitly related concepts (*Carlos*, *Guerrero*, *México*, and *Richard*).

The association strength between each pairwise can be measured taking into account different characteristics, such as the shortest path between concepts pairwise, the depth of their common ancestor, and information content [36].

We have adopted the Resnik approach [37] to measure the similarity between two concepts c_1 and c_2 according to the information content, using the formula

$$\mathrm{Sim}\left(p\left(c_1, c_2\right)\right) = \frac{\mathrm{IC}\left(\mathrm{MSCA}\left(c_1, c_2\right)\right)}{\mathrm{IC}\left(c_1\right) + \mathrm{IC}\left(c_2\right)}, \tag{2}$$

where $\mathrm{MSCA}(c_1, c_2)$ denotes the common ancestor of c_1 and c_2 with the higher information content. IC is the information content calculated for each node c in the ontology, whereas the more specific the node in the ontology is, the greater its information content is. There are different metrics to calculate IC [36].

Generally, these metrics are intrinsic. Namely, they are based on the topological information of the ontology and consider the instances occurrence. This approach considers the occurrence of an instance x quantified as $l(x) = \log_2 \mathrm{pr}(x)$, which has been reformulated as stated in

$$\mathrm{IC} = -\log_2 \frac{I(D(c))}{I(C)}, \tag{3}$$

where $I(D(c))$ denotes the number instances of the concept c and $I(C)$ represents the number of instances on the ontology.

From the ontology in Figure 2 which contains 1000 resources including the entities *Person*, *Publication*, and *ResearchGroup,* we can see a group of 600 people interested in some research group *(ResearchGroup)* and 100 people *(Author)* who wrote some publications *(Publication)*. The information content in *interestedIn* and *writtenBy* is obtained as stated in

$$\begin{aligned}
&\mathrm{IC}(interestedIn(Person, ResearchGroup)) \\
&\quad = -\log_2 \mathrm{pr}(interestedIn(Person, ResearchGroup)) \\
&\quad = -\log_2 \frac{600}{1000} = -\log_2 0.6 \approx 0.73, \\
&\mathrm{IC}(writtenBy(Publication, Author)) \\
&\quad = -\log_2 \mathrm{pr}(writtenBy(Publication, Author)) \\
&\quad = -\log_2 \frac{100}{1000} = \log_2 0.1 \approx 3.32.
\end{aligned} \tag{4}$$

The information content in a property represents the strength of the discrimination among the relationships. However, this is not enough to determine the meaning of the entity. We propose to measure the weight of each property linked to a concept c.

Relationships Weight. Based on information theory, the amount of information contained in a random variable over another variable is measured by mutual information. This strategy has been proposed by Cover [38] and we have adapted it to measure the relationship strength of pairwise c_1 and c_2.

$$\mathrm{MI}(p(c_1, c_2)) = \sum \sum \mathrm{pr}(c_1, c_2) \cdot \log_2 \frac{\mathrm{pr}(c_1, c_2)}{\mathrm{pr}(c_1) \cdot \mathrm{pr}(c_2)}, \tag{5}$$

where $\mathrm{pr}(c_1, c_2)$ is the probability of relationship e belonging to a set of properties of c_1 and c_2. $\mathrm{pr}(c_1)$ is the probability of relationship belonging to set of properties of c_1, whereas $\mathrm{pr}(c_2)$ is the probability of relationship e belonging to set of properties of c_2.

Figure 5 shows the relationships *writtenBy*, *memberOf*, *hasAdvisor*, and *livesIn* belonging to *Richard* entity in the ontology. The instances of these relationships are shown in Figure 6.

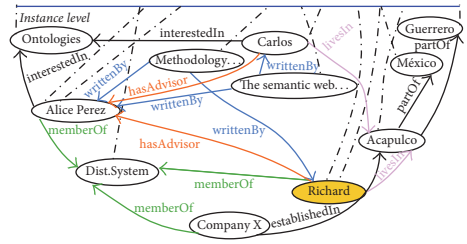


FIGURE 5: Relationships *memberOf*, *writtenBy*, *hasAdvisor*, and *livesIn* in the ontology.

As an example, let us calculate the relationship weight between *Richard* and *Methodology*, which is *writtenBy*, and it is computed as stated in

$$\begin{aligned}
&\mathrm{MI}(writtenBy(Publication, Author)) \\
&\quad = I(Methodology, Richard) \\
&\qquad \cdot \log_2 \left(\frac{I(Methodology, Richard)}{I(Methodology) \cdot I(Richard)} \right) \\
&\qquad + I(Metodology, AlicePerez) \\
&\qquad \cdot \log_2 \left(\frac{I(Methodology, AlicePerez)}{I(Methodology) \cdot I(AlicePerez)} \right) \\
&\qquad + I(TheSemanticWeb, AlicePerez) \\
&\qquad \cdot \log_2 \left(\frac{I(TheSemanticWeb, AlicePerez)}{I(TheSemanticWeb) \cdot I(AlicePerez)} \right) \\
&\qquad + I(TheSemanticWeb, Carlos) \\
&\qquad \cdot \log_2 \left(\frac{I(TheSemanticWeb, Carlos)}{I(TheSemanticWeb) \cdot I(Carlos)} \right) \\
&\quad = \frac{1}{4} \cdot \log_2 \left(\frac{1/4}{(1/2) \cdot (1/4)} \right) + \frac{1}{4} \\
&\qquad \cdot \log_2 \left(\frac{1/4}{(1/2) \cdot (1/2)} \right) + \frac{1}{4} \cdot \log_2 \left(\frac{1/4}{(1/2) \cdot (1/2)} \right) \\
&\qquad + \frac{1}{4} \cdot \log_2 \left(\frac{1/4}{(1/2) \cdot (1/4)} \right) = 0.5.
\end{aligned} \tag{6}$$

It should be noted that a relationship can have many instances. Consequently, calculating the relationships weight would have a high computational cost. Thus, we calculate the mutual information as stated in

$$\mathrm{MI}(e) = \approx \log_2 \left(\frac{1/[I(e)]}{(1/I(c_1)) \cdot (1/I(c_2))} \right), \tag{7}$$

where $[I(e)]$ represents all relationships e in the relationships set, $I(c_1)$ represents all relationships in c_1 (subject), and $I(c_2)$ represents all relationships in c_2 (object).

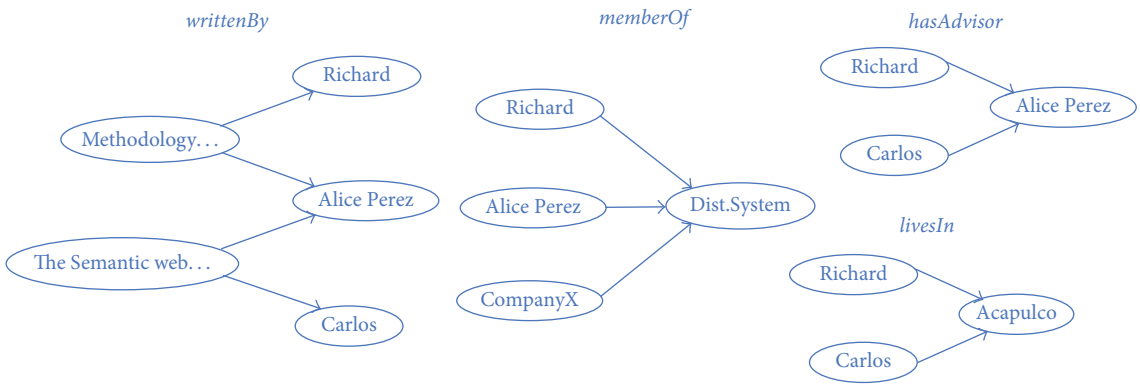


Figure 6: Examples of *writtenBy*, *memberOf*, and *hasAdvisor* entities and *livesIn* property.

Table 1: Document annotation.

Entity	Document	Weight
http://ex/onto#State	D1	0.5
http://ex/onto#State	D2	0.2
http://ex/onto#State	D87	0.67
http://ex/onto#Mexico	D1	0.45
http://ex/onto#Mexico	D23	0.6

Combining Association and Relationship Weights. The combination of weights requires considering several methods of aggregation, such as average, addition, and multiplication. A weighted sum as combination method to adjust the influence of each factor on the total weight was selected. Finally, to combine the association between each pair of concepts (see (2)) and the weights of the relationships (see (7)), we calculate the final weight to obtain the entities context, as stated in

$$W\left(P\left(c_i, c_j\right)\right) = \alpha \cdot \mathrm{Sim}\left(c_1, c_2\right) + \beta \cdot \mathrm{MI}\left(p\left(c_1 c_2\right)\right), \quad (8)$$

where $0 \leq \alpha, \beta \leq 1$. Sim and MI were normalized to be in the 0, 1 range by unit-based normalization [13], stated in

$$\frac{\mathrm{Sim} - \min_{p \in P}\mathrm{Sim}}{\max_{p \in P}\mathrm{Sim} - \min_{p \in P}\mathrm{Sim}}, \quad \frac{\mathrm{MI} - \min_{p \in P}\mathrm{MI}}{\max_{p \in P}\mathrm{MI} - \min_{p \in P}\mathrm{MI}}. \quad (9)$$

4.3. Terms Extraction and Documents Annotation. The textual descriptions of instances and entities semantic context obtained in the previous stage are searched in the inverted index to extract and generate a documents' annotation table containing the ontology entity, the belonging document, and its weight (see Table 1).

The annotations weight is done by means of TF-IDF algorithm. Term frequency (TF) is the local weighting factor reflecting the importance of a term within a document. Document frequency (DF) is the global weighting factor considering the importance of a term within the document collection. Inverse document frequency (IDF) calculates the frequency of a document within the collection. TF and IDF are calculated using the formulas stated in (10) and (11).

$$\mathrm{TF} = \frac{\mathrm{freq}_{x,d}}{\max_y \mathrm{freq}_{y,d}}, \quad (10)$$

where $\mathrm{freq}_{x,d}$ is the number of occurrences of term x within document d and $\max_y \mathrm{freq}_{y,d}$ is the number of occurrences of all terms within document d.

$$\mathrm{IDF} = \log \frac{|N|}{df}, \quad (11)$$

where $|N|$ is the total number of documents in the collection and df represents the documents where term x appears. The weight dx for x in d is the combination of TF $*$ IDF.

Finally, the annotations are represented in the form of serialized triplets in JSON-LD.

5. Evaluation

Pearson and Spearman correlation were used in order to measure the agreement with the human judgments. Pearson correlation measures the linear correlation between two variables, uses the ranges, orders numbers of each group of subjects, and compares those ranges. Spearman is a correlation measure between two continuous random variables.

Experimental Setup

Ontology and KIM Platform Knowledge Base [23]. This ontology has 271 classes, and 120 relationships and attributes. Some declared classes are of general importance such as People, Organizations, Government, and Location. The knowledge base consists of 200,000 instances, 50,000 locations, 130,000 organizations, 6,000 people, and more.

DBpedia [3]. DBpedia is general-purpose and multilingual in nature and has comprehensiveness. For this reason, it was selected for our experimentation. The English version contains 685 classes and 2795 properties; and the knowledge base is more than 4 million instances. DBpedia contains multiple classification systems, such as YAGO, Wikipedia Categories, and the hierarchical subgraph of the DBpedia Ontology. The

Table 2: Summary of Corpus LP50 annotations with KIM and DBpedia.

#doc	Words	Mention detection	Linked KIM	Linked DBpedia
(1)	80	13	8	30
(2)	98	21	10	37
(3)	98	17	7	34
(4)	106	24	4	42
(5)	80	13	9	47
(6)	97	15	14	43
(7)	97	27	8	39
(8)	82	24	10	35
(9)	126	12	7	28
(10)	76	23	11	41
(11)	83	17	7	31
(12)	67	15	8	38
(13)	103	4	10	21
(14)	105	16	9	24
(15)	90	17	12	45
(16)	75	18	11	41
(17)	73	15	8	29
(18)	62	16	7	25
(19)	103	27	13	33
(20)	122	19	11	25
(21)	94	18	6	31
(22)	61	12	6	22
(23)	72	13	7	23
(24)	54	13	5	16
(25)	57	13	5	29

Wikipedia Category system has the highest coverage of entities among all three options. To overcome these issues, we use the Wikipedia Category Hierarchy by Kapanipathi et al. [39].

Data Sets. LP50 are data sets of documents compiled by Lee and Welsh [14], which was used for our experimentation. LP50 is composed of 50 general-purpose news documents with lengths between 50 and 126 words.

Lucene. The Lucene's documents were indexed to generate a documents index that includes the list of mentions and documents where they appear. Also, the TagMe tool was used for mentions detection in the documents. We used the Jena library for the analysis and extraction of the entities in the ontology. We use Jena TDB triple store to operate DBpedia locally.

For space issues, Table 2 shows only the results of the first 25 annotated documents. Column 2 shows the number of words in each document. Column 3 shows the mentions detected in each document. The columns 4 and 5 show the mentions linked in the KIM and DBpedia ontologies, respectively.

Table 2 shows only few mentions linked with KIM knowledge base. This is mainly due to the fact that (i) ontologies and instances are limited and (ii) the entities must have a value in rdfs: label.

Table 3: Precision, recall, F-measure, and accuracy of semantic annotations between context-free and context-based semantic annotation.

Means	Context-free	Context-based
Precision	0.621	0.893
Recall	0.839	0.799
F-measure	0.678	0.815
Accuracy	0.644	0.835

In the first case, if an ontology and knowledge base have a limited scope, a mention in the ontology could not exist. Therefore, ontology with a larger population (as DBpedia) will cover most of the mentions obtained in the documents.

In the second case, the entities must have value in rdfs: label, since this depends on links between the mentions and entities. DBpedia has more mention-entity link since it contains more than 4 million instances.

Table 3 shows the results of the semantic annotation evaluation DBpedia. The standard measures precision, recall, F measure, and accuracy were used for evaluating the annotations obtained. Precision is the rate between the relevant instances of the ontology and the total number of instances retrieved, and recall is the rate between the number of relevant instances retrieved and the total number of relevant instances existing in the ontology:

$$\begin{aligned} \text{Precision} &= \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|}, \\ \text{Recall} &= \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|}, \end{aligned} \tag{12}$$

where TP (True Positives) are the set of retrieved instances that are relevant, FP (False Positives) are the set of retrieved instances that are not relevant, and FN (False Negatives) are the set of instances that are wrongly retrieved as nonrelevant.

The results show that our proposed method of context-based semantic annotation improves the results of the context-free annotation method.

Comparison to State of the Art. The results of our similarity calculation approach were compared with different strategies shown of the state of the art. Some approaches only take into account the weight of the edges, the association between each pairwise concept, and the ontology structure. We compared our approach with different methods in the literature that measure document similarity and use the LP50 data set. Among the methods analyzed are Latent Semantic Analysis (LSA) [40], Explicit Semantic Analysis (ESA) [41], Salient Semantic Analysis (SSA) [40], Graph Edit Distance (GED) [42], and ConceptsLearned [43].

The results obtained of comparison of our approach with other methods using LP50 dataset are shown in Table 4. The values of Pearson and Spearmen correlation of our approach were 0.745 and 0.65, respectively. This result was best compared to the results of other approaches. Thus, our approach significantly outperforms, to our knowledge, the most competitive related approaches, although ConceptsLearned has

Table 4: Comparing our approach with other methods using LP50 dataset.

Approach	Pearson correlation	Spearman correlation
LSA	0.59	0.53
ESA	0.68	0.59
SSA	0.71	0.64
GED	0.72	0.64
Our approach	0.745	0.65
ConceptsLearned	0.81	0.75

Table 5: Information content with extrinsic and intrinsic approaches.

Parameter	Metric	Pearson correlation
Depth	Intrinsic	0.743
Descendant	Intrinsic	0.743
Instances	Extrinsic	0.745

a better correlation of Pearson and Spearman (0.81 and 0.75). This is because ConceptsLearned uses 17 more features compared to ours, but the computational cost is high.

Comparison with Other Metrics for Information Content (IC) Calculation. We performed tests with different metrics to calculate the information content and use the extrinsic approach. The information content with the intrinsic approach can be performed using two parameters: (1) the depth of the class and (2) the descendants of a class.

Table 5 shows the slight advantage of considering the ontology instances with the extrinsic information content.

6. Conclusions

In this paper, we have presented a semantic annotation of unstructured documents approach. Which considers concepts similarity in ontology through its semantic relations.

The unstructured documents are represented as graphs, the nodes represent the mentions, and the edges represent the semantics and relationships. Each semantic relationship has a weighting measure assigned. Thus, the significant relationships have a higher weight.

The context extraction was done through the computation of association between pairwise concepts and the weight of entity relations. The sum of the two values is the one that measures the meaning or context of an entity. We also took advantage of instances in the knowledge base to measure the information content classes and relationships.

According to the state of the art the results obtained with our approach give the best results.

As future work, we are trying to reduce the knowledge base by selecting the entities whose definition is more likely to be used in the corpus. Additionally, Word2vec tool for semantic extraction of terms and documents can be used.

Finally, this approach also has been compared with other proposals available in the literature.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research work has been partially funded by European Commission and CONACYT, through the SmartSDK project. It also has been partially funded by TecNM with the project 6021.17-P.

References

[1] T. Zhang, K. Liu, and J. A. Zhao, "Graph-based similarity measure between wikipedia concepts and its application in entity linking system," *Journal of Chinese Information Processing*, vol. 29, no. 2, pp. 58–67, 2015.

[2] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data—the story so far," *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, article 122, 2009.

[3] C. Bizer, J. Lehmann, G. Kobilarov et al., "DBpedia—a crystallization point for the Web of Data," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154–165, 2009.

[4] K. Bollacker, R. Cook, and P. Tufts, "Freebase: a shared database of structured general human knowledge," in *In Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI '07)*, vol. 2, pp. 1962-1963, AAAI Press, British Columbia, Canada, July 2007.

[5] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 697–706, Alberta, Canada, May 2007.

[6] L. Bos and K. Donnelly, "The advanced terminology and coding system for ehealth," *Studies in Health Technology and Informatics*, vol. 121, pp. 279–290, 2009.

[7] J. M. Ruiz-Martínez, R. Valencia-García, J. T. Fernández-Breis, F. García-Sánchez, and R. Martínez-Béjar, "Ontology learning from biomedical natural language documents using UMLS," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12365–12378, 2011.

[8] C. Caracciolo, A. Stellato, A. Morshed et al., "The AGROVOC linked dataset," *Journal of Web Semantics*, vol. 4, no. 3, pp. 341–348, 2013.

[9] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: historical perspective and recent advances," *Journal of the American Medical Informatics Association*, vol. 17, no. 3, pp. 229–236, 2010.

[10] R. Berlanga, V. Nebot, and E. Jiménez, "Semantic annotation of biomedical texts through concept retrieval," *Procesamiento del Lenguaje Natural*, vol. 45, pp. 247–250, 2010.

[11] M. Dai, N. Shah, W. Xuan et al., "An efficient solution for mapping free text to ontology terms," in *Proceedings of the American Medical Informatics Association Symposium on Translational BioInformatics (AMIA-TBI '08)*, Washington, DC, USA, November 2008.

[12] R. Navigli and M. Lapata, "An experimental study of graph connectivity for unsupervised word sense disambiguation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 678–692, 2010.

[13] E. Agirre, O. L. de Lacalle, and A. Soroa, "Random walks for knowledge-based word sense disambiguation," *Computational Linguistics*, vol. 40, no. 1, pp. 57–84, 2014.

[14] M. Lee and M. Welsh, "An empirical evaluation of models of text document similarity," in *Proceedings of the 27 Annual Conference of the Cognitive Science Society (CogSci '05)*, pp. 1254–1259, Erlbaum, Stresa, Italy, July 2005.

[15] P. Ferragina and U. Scaiella, "Fast and accurate annotation of short texts with wikipedia pages," *IEEE Software*, vol. 29, no. 1, pp. 70–75, 2012.

[16] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "DBpedia spotlight: shedding light on the web of documents," in *Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS '11)*, pp. 1–8, Graz, Austria, September 2011.

[17] OpenCalais, 2014, http://www.opencalais.com/.

[18] IBM, AlchemiLanguage, 2015, https://alchemy-language-demo.mybluemix.net/.

[19] D. O. C. S. The University of Sheffield, Developing Language Processing Components with GATE, 8 edition, 2017 https://gate.ac.uk/userguide.

[20] M. Laclavík, M. Šeleng, M. Ciglan, and L. Hluchý, "Ontea: platform for pattern based automated semantic annotation," *Computing and Informatics*, vol. 28, no. 4, pp. 555–579, 2009.

[21] D. Rebholz-Schuhmann, M. Arregui, S. Gaudan, H. Kirsch, and A. Jimeno, "Text processing through web services: calling Whatizit," *Bioinformatics*, vol. 24, no. 2, pp. 296–298, 2008.

[22] C. Tao, D. Song, D. Sharma, and C. G. Chute, "Semantator: semantic annotator for converting biomedical text to linked data," *Journal of Biomedical Informatics*, vol. 46, no. 5, pp. 882–893, 2013.

[23] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov, "KIM—semantic annotation platform," in *Proceedings of the 2nd International Conference on Semantic Web Conference (ISWC '03)*, vol. 2870 of *Lecture Notes in Computer Science*, pp. 834–849, Springer, Sanibel Island, Fla, USA, October 2003.

[24] P. Castells, M. Fernández, and D. Vallet, "An adaptation of the vector-space model for ontology-based information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 2, pp. 261–272, 2007.

[25] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced information retrieval: an ontology-based approach," *Journal of Web Semantics*, vol. 9, no. 4, pp. 434–452, 2011.

[26] R. Berlanga, V. Nebot, and M. Pérez, "Tailored semantic annotation for semantic search," *Journal of Web Semantics*, vol. 30, pp. 69–81, 2015.

[27] V. Nebot and R. Berlanga, "Exploiting semantic annotations for open information extraction: an experience in the biomedical domain," *Knowledge and Information Systems*, vol. 38, no. 2, pp. 365–389, 2014.

[28] D. Fuentes-Lorenzo, N. Fernández, J. A. Fisteus, and L. Sánchez, "Improving large-scale search engines with semantic annotations," *Expert Systems with Applications*, vol. 40, no. 6, pp. 2287–2296, 2013.

[29] L. Ding, T. Finin, A. Joshi et al., "Swoogle: a search and metadata engine for the semantic web," in *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM '04)*, pp. 652–659, Washington, DC, USA, November 2004.

[30] Y. Lei, V. Uren, and E. Motta, "SemSearch: a search engine for the semantic web," in *Managing Knowledge in a World of Networks*, S. Staab and V. Svtek, Eds., vol. 4248 of *Lecture Notes in Computer Science*, pp. 238–245, Springer, Berlin, Germany, 2006.

[31] C. Tao, Z. Yongjuan, Z. Shen, C. Chengcai, and C. Heng, "Building semantic information search platform with extended sesame framework," in *Proceedings of the 8th International Conference on Semantic Systems (ISEMANTICS '12)*, pp. 193–196, New York, NY, USA, September 2012.

[32] S. Saha, A. Sajjanhar, S. Gao, R. Dew, and Y. Zhao, "Delivering categorized news items using RSS feeds and web services," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology (ScalCom '10)*, pp. 698–702, Bradford, UK, July 2010.

[33] V. Lopez, M. Fernández, E. Motta, and N. Stieler, "PowerAqua: supporting users in querying and exploring the Semantic Web," *Journal of Web Semantics*, vol. 3, no. 3, pp. 249–265, 2012.

[34] A. Singhal, G. Salton, M. Mitra, and C. Buckley, "Document length normalization," *Information Processing & Management*, vol. 32, no. 5, pp. 619–633, 1996.

[35] I. Augenstein, L. Derczynski, and K. Bontcheva, "Generalisation in named entity recognition: a quantitative analysis," *Computer Speech and Language*, vol. 44, pp. 61–83, 2017.

[36] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL '94)*, pp. 133–138, ACM, Las Cruces, NM, USA, June 1994.

[37] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, vol. 2, pp. 448–453, Morgan Kaufmann Publishers Inc., Quebec, Canada, August 1995.

[38] T. M. Cover and J. A. Thomas, *Elements of Information Theory Wiley Series in Telecommunications and Signal Processing*, John Wiley & Sons, New York, NY, USA, 2007.

[39] P. Kapanipathi, P. Jain, C. Venkataramani, and A. Sheth, "Hierarchical interest graph," 2016, http://wiki.knoesis.org/index.php/Hierarchical_Interest_Graph.

[40] S. Hassan and R. Mihalcea, "Semantic relatedness using salient semantic analysis," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 884–889, San Francisco, Claif, USA, August 2011.

[41] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI '07)*, pp. 1606–1611, Hyderabad, India, January 2007.

[42] M. Schuhmacher and S. P. Ponzetto, "Knowledge-based graph document modeling," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14)*, pp. 543–552, ACM, New York, NY, USA, February 2014.

[43] L. Huang, D. Milne, E. Frank, and I. H. Witten, "Learning a concept-based document similarity measure," *Journal of the Association for Information Science and Technology*, vol. 63, no. 8, pp. 1593–1608, 2012.

Hindawi
Scientific Programming
Volume 2017, Article ID 6459582, 6 pages
https://doi.org/10.1155/2017/6459582

Research Article

Applying Softcomputing for Copper Recovery in Leaching Process

Claudio Leiva,[1] Víctor Flores,[2] Felipe Salgado,[1] Diego Poblete,[1] and Claudio Acuña[3]

[1]*Department of Chemical Engineering, Universidad Católica del Norte, Angamos Av. 0610, Antofagasta, Chile*
[2]*Department of Computing & Systems Engineering, Universidad Católica del Norte, Angamos Av. 0610, Antofagasta, Chile*
[3]*Department of Chemical and Environmental Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile*

Correspondence should be addressed to Víctor Flores; vflores@ucn.cl

Received 5 May 2017; Revised 10 October 2017; Accepted 30 October 2017; Published 5 December 2017

Academic Editor: Jezreel Mejia-Miranda

The mining industry of the last few decades recognizes that it is more profitable to simulate model using historical data and available mining process knowledge rather than draw conclusions regarding future mine exploitation based on certain conditions. The variability of the composition of copper leach piles makes it unlikely to obtain high precision simulations using traditional statistical methods; however the same data collection favors the use of softcomputing techniques to enhance the accuracy of copper recovery via leaching by way of prediction models. In this paper, a predictive modeling contrasting is made; a linear model, a quadratic model, a cubic model, and a model based on the use of an artificial neural network (ANN) are presented. The model entries were obtained from operation data and data of piloting in columns. The ANN was constructed with 9 input variables, 6 hidden layers, and a neuron in the output layer corresponding to copper leaching prediction. The validation of the models was performed with real information and these results were used by a mining company in northern Chile to improve copper mining processes.

1. Introduction

Due to the complexity of extraction mining worldwide, computer models are becoming an essential tool for reducing production costs [1]. Computational techniques have recently been incorporated into the copper industry to improve both the process and the results obtained through leaching, thus achieving tangible results such as higher levels of production, identification of low grade ore, and reduction of production costs [2].

Copper mining in Chile is the most profitable industry and contributes to approximately 12% of world copper production [3, 4]. In recent years, the mining industry in Chile has begun to utilize and show an increasing interest in Artificial Intelligence techniques in processes such as copper production predictions, where the traditional methods tend to lack certainty and knowledge, and one of these techniques is softcomputing [5].

Softcomputing is the branch of Artificial Intelligence that groups paradigms and techniques working with incomplete and imprecise information in critical processes, in order for the company to obtain useful solutions for tasks such as prediction or discovery of information or knowledge [4–6]. A definition of softcomputing is *"an emerging approach to computing which parallels that remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision"* [7].

This paper describes the work done with ANN to generate behavior predictions of the pile in the copper leaching domain. The use of ANNs has been used in combination with other techniques commonly used as a predictor in the mining industry, such as linear regression, with the aim of contrasting and enhancing the quality of results. An ANN can be defined as a set of computational units (neurons) that are highly interconnected. Each neuron is also called node which represents the biological neuron, and the connection among them represents the biological neuronal network [8]. The basic element of data processing in a neural network is the perceptron, a simple element that with an input vector provides a single output, for which it uses an activation

function. An extended and well-accepted way to classify ANN is according to architecture and learning. Architecture refers to the arrangement and connections between neurons, number of input and output variables, and number of hidden layers, while learning classification refers to training the network by way of patterns, and the patterns are performed iteratively until the restriction $|O_d - O_i| \leq \delta$ is satisfied, O_d being the desired output, O_i the output of the nth iteration, and δ a small numerical value [9, 10].

This paper describes the copper recovery prediction process for an extraction mine in northern Chile, using both a statistical model (methods traditionally used in the copper leaching industry) and a neural model known as multilayer perceptron with backpropagation [11, 12]. The multilayer perceptron model is considered the most widely used neural network due to its efficiency and ease of understanding and interpretation of both the network and its results. The parameters of entry to the ANN were selected with respect to the objectives and conditions of the company. Specifically, this paper reflects the processes and results obtained in the SCM Franke Company, of the international group KGHM International, which has been operating in Chile since 2009.

2. Background and Related Works

The process of applying different techniques to adjust the weights related to each of the input variables of neurons in an ANN is known as learning [8, 13]. For this process, which is also known as network training, several techniques can be followed but the most common ones are as follows: supervised and unsupervised training [14]. The supervised training consists of using a series of pairs (I_i, O_i) where I_i represents an input data vector and O_i represents the desired output vector. The training consists of using an algorithm to make the values of I_i as close as possible to the vector O_i, while in the unsupervised training only the vector I_i is available and the training algorithm attempts to find the hidden structures between these data to adjust the weights of each neuron in the network [15]. In both cases, the learning rules act by modifying the weights to achieve the goal, for which there are several techniques as well. One of the most used techniques is error correction via adjustment of weights that essentially consists of obtaining a value δ very close to 0.

Currently, the applications that include ANN are in all areas of science and engineering, that is, energy production estimation systems [16]. In engineering in particular, the increase in ANN usage in recent decades is significant, being applied to tasks such as prediction and regression. For example, in [17] a comparison between ANN and regression analysis is described, in terms of notation and implementation of both paradigms, and also highlights the advantages in terms of implementation. In [4, 18] works are presented related to metal-mechanical industry that compares the performance of the Bayesian networks with ANNs in the prediction of surface quality in high-speed machining processes. The most frequently used techniques for this process are ANN [13] and linear or multiple regression methods [14]. In [19], ANNs are used to predict the influence of biological and nonbiological parameters on the precipitation of ferric iron through a bioleaching process. In [20] a process of construction of a three-layer backpropagation ANN to predict the concentration of heavy metals (Zn, Ni, Cd, and Pb) as waste in a zinc leach cell is described.

In [21] an ANN used to predict the copper production process is presented as well. Moreover, in the work recently reported in [22] an ANN is used to predict copper flotation rates under different operating conditions; different dosages of chemical reagents used in the process, feed rate, and granulometry are used for this process. A three-layer backpropagation neural network (input layer, two hidden layers, and one output), topology (9-10-10-3), was used, and the quality of the prediction in the testing process was 93%.

3. Materials and Methods

The SCM Franke Company uses three industrial processes widely known in the copper industry to produce metallic copper via hydrometallurgy: leaching in dynamic cells, solvent extraction, and electrowinning. The goal in these processes is to achieve the highest copper production by saving resources and having the lowest possible environmental impact. In this sense, the company considered carrying out simulations to predict the copper production through leaching processes. The objective was to produce an efficient model (with an accuracy greater than 95%) for the estimation of copper leaching recovery based on historical data.

A previous analysis by the SCM Franke Company conducted to determine which of these processes is the least controlled (number of influencing factors and homogeneity of irrigation) found it to be the leaching process in dynamic stacks. Based on this previous analysis, the parameters to be used in the prediction models and the desired quality level (95%) were identified and employed as the desired adjustment value in the simulation to test the results of the models.

The study summarized the prediction of copper extraction in dynamic stacks, applying mathematical models (statistical model) that require complete information to give precise results and an ANN, which is considered to have advantages over the treatment of incomplete information in terms of generating predictions in industrial production.

Pile leaching copper is a percolation process that operates above ground. The procedure is illustrated in Figure 1. The oxide copper ore is piled up on leach pads which have an inclination of 3° approx. and a rubber lining that seals the ground beneath. These heaps are 3-meter-high and 72 meters in length with a base area of 2,880 square meters, covering 120,000 tons of ore.

A sprinkler system is installed on top of each heap allowing diluted sulfuric acid to be fed uniformly over the ore. While the solution percolates through the pile, copper is leached out of the ore. The pregnant leaching solution is collected by a drainage system below the pile and is led through a collection ditch into a pond.

4. Methodology

The research consists of two stages, the first called "copper recovery modeling" in which models are generated with each

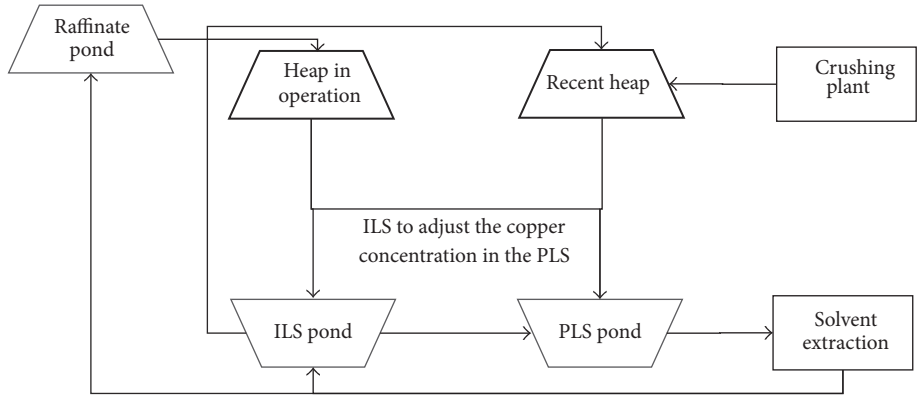


Figure 1: Process flow diagram of the pile leaching process.

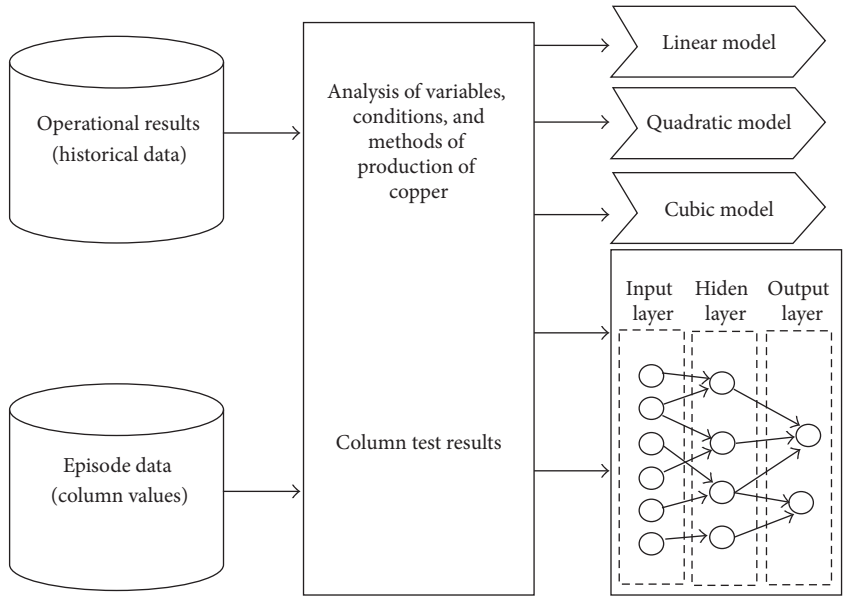


Figure 2: Work scheme for copper recovery prediction in dynamic stacks.

of the aforementioned methods and the second stage called "Evaluation" where the results are compared and the quantity and quality of information are obtained.

Both stages are beneficial for the SCM Frank Company in order to understand the leaching process with the characteristics of the ore and to be able to predict the recovery of copper in dynamic stacks, focused on achieving a less than 5% margin of error in estimation. These two stages are possible due to the previous analysis and selection of variables and parameters representing the inputs to the prediction methods; Figure 2 graphically shows this work scheme.

4.1. Statistical Models. In order to perform the modeling, the most recent literature in leaching process was considered, taking into account the variables that affect the recovery of copper. In addition, the historical operational results data and process-related pilot testing were taken into account, which we call pilot data. Pilot tests consist of testing columns with strict control measures regarding irrigation rates, acid concentrations in irrigation solutions, and operating cycles, conditions that vary depending on the test to be performed. A case study was generated with a database of approximately 30,000 pieces of data.

4.1.1. Data Collection. For the purpose of this study, the historical database corresponding with industrial and piloting performance from SCM Franke Company was used. The data stored in the database correspond to both plant data operation (weighted values and accumulated daily) and pilot data.

The operating plant data were obtained at a frequency of 4 hours for 1 year. For some periods, the irrigation was stopped on some cells or modules in service; due to these periods, inconsistent results were disregarded and corresponding data was not considered for the data collection process; some "noise" in the system and useless information were

Table 1: Entry parameters to the statistical model.

Notation	Name and description	Optimum values
x_1	Monoclass granulometry (refers to the mineral)	Between 11,5 mm and 15 mm
x_2	Irrigation rates	Between 14 (l × h)/m^2 and 6 (l × h)/m^2
x_3	Total acid added	Between 0,5 g/l and 100 g/l
x_4	Pile high	Between 1 m and 5 m
x_5	Total copper grade	Between 0,5% and 2%
x_6	CO_3 grade	Between 0,5% and 10%
x_7	Leaching ratio	Less than 15 m^3/TMS in pilot and Less than 8 m^3/TMS in plant
x_8	Operation days	Between 90 and 120 days
x_9	Soluble copper grade stacked	70% of the total copper grade

Table 2: Comparative of R^2 values.

Model	R^2 value
Lineal model	69,8%
Quadratic model (standard deviation = 9,4)	89,9%
Cubic model (standard deviation = 8,3)	92,3%
ANN model	97,9%

disregarded as well. Regarding the pilot data, the method of information collection was the same used for the data of operation.

The next step was to identify the parameters that affect the copper recovery and to form a robust database with this information for use in the preparation and evaluation of process models using different techniques in order to determine one that meets the plant requirements (adjustment greater than 95%) with respect to plant and pilot operating data. The parameters considered in the statistical model are detailed in Table 1.

A search was performed to find the combination of suitable variables to produce the most successful model for the desired response. The system response was defined as the percentage of copper extraction. To validate the model, the response variable was monitored. The model was monitored in order to obtain the best fit for the operating conditions and variations. When model misalignment was below acceptable (much lower than the desired 95%), the model was readjusted and remade considering the operating ranges that were initially not considered; parameters that were initially not considered were now carried out.

To select the subset of variables that ensure a practical model, coefficients were generated using the Minitab software. In detail, the calculated coefficients were as follows: Vars, *R*-cuad, adjusted *R*-cuad, *C*p of Mallows, and *S* (standard dispersion). A preliminary statistical model was made considering the Press factor, used to avoid overadjustment. Additionally, for this model the following restrictions were considered: (1) that the adjusted values of *S*, *R*-cuad (R^2), and *R*-squared were the highest possible and (2) that the value of the Press factor was the highest grade possible.

Originating from the preliminary model and utilizing the Minitab tools, three statistical models were generated: linear adjustment model, quadratic adjustment model, and cubic adjustment model. Table 2 summarizes each of these models and the adjustments results according to the value R^2.

For the lineal model (1),

$$Y = \sum_{i=1}^{9} \alpha_i * \chi_i + C. \tag{1}$$

For the quadratic and cubic models (2)

$$\begin{aligned} Y &= \sum_{i=1}^{9} \alpha_i * \chi_i + \sum_{j=1}^{9}\sum_{i=1}^{9} b_i * \chi_i * \chi_j + C, \\ Y &= \sum_{i=1}^{9} a_i * \chi_i + \sum_{j=1}^{9}\sum_{i=1}^{9} b_i * \chi_i * \chi_j \\ &\quad + \sum_{k=1}^{9}\sum_{j=1}^{9}\sum_{i=1}^{9} c_i * \chi_i * \chi_j * \chi_k + C, \end{aligned} \tag{2}$$

where

(i) χ_i, χ_j, and χ_k represent the input value of the variable;

(ii) a_i, b_i, and c_i represent the weight of the input;

(iii) C represents a constant

4.2. ANN Model Base. To perform the RNA modeling, the MATLAB program was applied. For the network programming, the following parameters were used: 9 income variables

(see Table 1), hidden layers, 60% of the information for the network, 40% of the information for network validation, 0.2 of error as cycle-error, or maximum of 500 iterations. The data for the training and validation was randomly selected.

In the variables selection the following was considered: the experience accumulated by operation experts of the SCM Franke Company, the results of the piloting, bibliography, and the results observed with the statistical models described above. In the ANN configuration, nine input variables were used (see Table 1) for a multilayer perceptron model with backpropagation algorithm.

The output layer was formed by a corresponding neuron to the prediction of copper recovery in dynamic piles, equally focusing on achieving a less than 5% error in the prediction estimation. The mathematical expression used in each neuron to obtain the output value is indicated in (3) where (x_i) represents the input value to the neuron and the variable (w_{ij}) represents the weight associated with the neuron. The task of initializing the weights associated with each input variable was performed randomly in a range normalized between $(-1, 1)$ and using the MAPMINMAX function of MATLAB (a function used to normalize the input and output values of the neurons), and threshold values that were used in the activation functions (sigmoid function) were initialized in a similar way (these data were not reflected in this document because they are subject to confidentiality agreements with the company). The mathematical expression of the sigmoid function is indicated in

$$y = \sum_{j=1}^{9} L_j \sum_{i=9}^{9} w_{ij}^{i} * x_i, \tag{3}$$

where

(i) x_i represents the ith input value to the neuron;

(ii) w_{ij}^{i} represents the ith weight associated with the neuron i of the layer j;

(iii) L_j represents the constant of the jth layer.

5. Results

Table 2 shows the R^2 values obtained by both, algorithmic models and ANN model; as seen in the table, the second-best value of R^2 corresponds to the cubic fit model. None of these values exceed the desired minimum value (95%); however, the values obtained in the simulation, using cubic algorithmic models, are close to what was obtained.

The best value corresponding to the ANN model during the learning phase and validation of the ANN, a 97,9% of adjustment, was obtained. Figure 3 shows the adjustment performed with the ANN.

6. Discussion

The result obtained through linear modeling was far from the required (89.9% v/s 95%), which indicates that the capacity to make the needed adjustment far exceeds what can be delivered by this type of modeling. The adjustment results

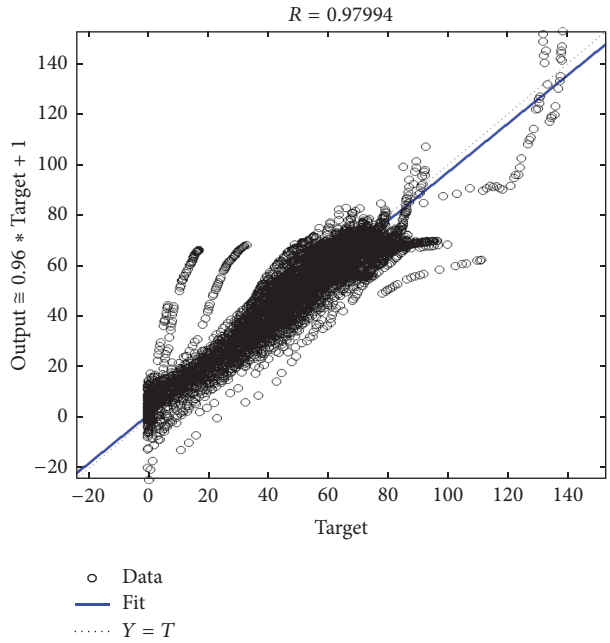


Figure 3: ANN validation model.

of the quadratic model were approximately 22% better than those obtained previously and the R^2 value was very similar to R^2 (adjusted), which means the quadratic model is better able to generate more realistic data projections. With respect to the cubic model, the observed adjustment was close to what was obtained using the quadratic model, which indicates that the use of higher-grade models does not present a significant improvement in relation to the model adjustment, due to the limitations given by the techniques used and the high variability presented by the data used.

The use of ANN for system modeling led to an improvement from 92.3% to 97.9%, compared to the cubic model. It was observed that the use of ANN achieved the objective of adjusting to fit, which was due to the complex network that this process uses to adjust the system response to the existing parameters.

To achieve these results through the use of ANN, the use of a program such as MATLAB or another advanced calculation program is required. Due to the high complexity of such a modeling technique, it requires a database with an abundance of information so as to give the program the information necessary to evaluate as many cases as possible. On the other hand, ANN has the disadvantage of not being easily detected compared to statistical models, so this could present a problem for new users.

7. Conclusion

SCM Franke has carried out simulations on other copper production processes and knows from experience that the prediction is much cheaper than the experimental work, but, in this case, our study describes a prediction model that represents the behavior of the leaching plant, considering the

variables initially defined with adjustment results higher than 95% obtained.

Accordingly, it was determined that ANN was the best model for SCM Franke mining leach plant, due to the high variability of the existing plant results and tests. The adjustment obtained was 97.9%, which is higher than the adjustment of 95% initially requested.

This study has served to obtain a comparison between prediction models and it can be intuited by the precision of the adjustments that the neural model has potential for use in future copper production prediction process. Furthermore, experience was gained in defining the model of an ANN, which can be used in future process simulations related to the improvement of copper attainment via softcomputing techniques in the SCM Franke Company or in similar companies.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Thanks are due to SCM Franke for supporting this project, specifically for the collaboration on data provision and storing and for the experience made available to the authors to select the parameters and criteria used for modeling.

References

[1] M. Milivojevic, S. Stopic, B. Friedrich, B. Stojanovic, and D. Drndarevic, "Computer modeling of high-pressure leaching process of nickel laterite by design of experiments and neural networks," *International Journal of Minerals, Metallurgy and Materials*, vol. 19, no. 7, pp. 584–594, 2012.

[2] H. Kamran Haghighi, D. Moradkhani, and M. M. Salarirad, "Modeling of synergetic effect of LIX 984N and D2EHPA on separation of iron and zinc using artificial neural network," *Transactions of the Indian Institute of Metals*, vol. 67, no. 3, pp. 331–341, 2014.

[3] ICSG PRESS RELEASE. Copper: Preliminary Data for September 2013, 2013.

[4] V. Flores, Y. Hadfeg, J. Bekios, A. Quelopana, and C. Meneses, "A method for automatic generation of explanations from a rule-based expert system and ontology," *Advances in Intelligent Systems and Computing*, vol. 537, pp. 167–176, 2017.

[5] L. A. Zadeh, "Fuzzy logic, Neural networks, and soft computing," *Communications of the ACM*, vol. 37, no. 3, pp. 77–84, 1994.

[6] L. G. Bergh, J. B. Yianatos, and C. A. Leiva, "Fuzzy supervisory control of flotation columns," *Minerals Engineering*, vol. 11, no. 8, pp. 739–748, 1998.

[7] L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48–56, 1994.

[8] J. J. Hopfield, *Artificial Neural Networks*, vol. 4, Berlin, 1988.

[9] A. J. Maren, C. T. Harston, and R. M. Pap, *Handbook of Neural Computing Applications*, San Diego, Calif, USA, 2014.

[10] A. Freeman James and M. Skapura David, *Neural Networks: Algorithms, Applications and Programming Techniques*, Addison-Wesley, New York, NY, USA, 1991.

[11] M. Cilimkovic, "Neural Networks and Back Propagation Algorithm," *FettTu-SofiaBg*, 2010.

[12] V. Skorpil and J. Stastny, "Neural networks and back propagation algorithm," *Electron Bulg Sozopol*, pp. 20–22, 2006.

[13] S. S. Lee and J. C. Chen, "On-line surface roughness recognition system using artificial neural networks system in turning operations," *The International Journal of Advanced Manufacturing Technology*, vol. 22, no. 7-8, pp. 498–509, 2003.

[14] C.-X. Feng and X.-F. Wang, "Surface roughness predictive modeling: Neural networks versus regression," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 35, no. 1, pp. 11–27, 2003.

[15] H.-Y. Shu, H.-C. Lu, H.-J. Fan, M.-C. Chang, and J.-C. Chen, "Prediction for energy content of taiwan municipal solid waste using multilayer perceptron neural networks," *Journal of the Air & Waste Management Association*, vol. 56, no. 6, pp. 852–858, 2006.

[16] T. C. Ogwueleka and F. N. Ogwueleka, "Modelling energy content of municipal solid waste using artificial neural network," *Journal of Environmental Health Science and Engineering*, vol. 7, no. 3, pp. 259–266, 2010.

[17] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: a review of applications," *Expert Systems with Applications*, vol. 36, no. 1, pp. 2–17, 2009.

[18] Correa M. Explanation of a Bayesian network classifier by means of decision trees. n.d.

[19] H. Golmohammadi, A. Rashidi, and S. Safdari, "Predvidstrokeanje precipitacije feri jona u procesu bioluženja primenom parcijalnih najmanjih kvadrata i veštačke neuronske mreže," *Chemical Industry & Chemical Engineering Quarterly*, vol. 19, no. 3, pp. 321–331, 2013.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 6, pp. 323–5336, 1986.

[21] H. Kamran Haghighi, M. Rafie, D. Moradkhani, B. Sedaghat, and A. Abdollahzadeh, "Modeling on Transition of Heavy Metals from Ni–Cd Zinc Plant Residue Using Artificial Neural Network," *Transactions of the Indian Institute of Metals*, vol. 68, no. 5, pp. 741–756, 2015.

[22] O. Salmani Nuri, E. Allahkarami, M. Irannajad, and A. Abdollahzadeh, "Estimation of selectivity index and separation efficiency of copper flotation process using ANN model," *Geosystem Engineering*, vol. 20, no. 1, pp. 41–50, 2017.

Hindawi
Scientific Programming
Volume 2017, Article ID 8131390, 19 pages
https://doi.org/10.1155/2017/8131390

Research Article

A Heterogeneous System Based on Latent Semantic Analysis Using GPU and Multi-CPU

Gabriel A. León-Paredes,[1,2] Liliana I. Barbosa-Santillán,[2] and Juan J. Sánchez-Escobar[3]

[1] *Universidad Politécnica Salesiana, Cuenca, Ecuador*
[2] *Universidad de Guadalajara, Guadalajara, JAL, Mexico*
[3] *Technical and Industrial Teaching Center, Guadalajara, JAL, Mexico*

Correspondence should be addressed to Gabriel A. León-Paredes; gleon@ups.edu.ec

Received 15 June 2017; Accepted 26 September 2017; Published 5 November 2017

Academic Editor: José María Álvarez-Rodríguez

Latent Semantic Analysis (LSA) is a method that allows us to automatically index and retrieve information from a set of objects by reducing the term-by-document matrix using the Singular Value Decomposition (SVD) technique. However, LSA has a high computational cost for analyzing large amounts of information. The goals of this work are (i) to improve the execution time of semantic space construction, dimensionality reduction, and information retrieval stages of LSA based on heterogeneous systems and (ii) to evaluate the accuracy and recall of the information retrieval stage. We present a heterogeneous Latent Semantic Analysis (hLSA) system, which has been developed using General-Purpose computing on Graphics Processing Units (GPGPUs) architecture, which can solve large numeric problems faster through the thousands of concurrent threads on multiple CUDA cores of GPUs and multi-CPU architecture, which can solve large text problems faster through a multiprocessing environment. We execute the hLSA system with documents from the PubMed Central (PMC) database. The results of the experiments show that the acceleration reached by the hLSA system for large matrices with one hundred and fifty thousand million values is around eight times faster than the standard LSA version with an accuracy of 88% and a recall of 100%.

1. Introduction

Latent Semantic Analysis (LSA) is a method that allows us to automatically index and retrieve information from a set of objects by reducing a term-by-document matrix using term weighting schemes such as Log Entropy or Term Frequency-Inverse Document Frequency (TF-IDF) and using the Singular Value Decomposition (SVD) technique. LSA improved one of the main problems of information retrieval techniques, that is, handling polysemous words, by assuming there is some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice [1]. LSA uses statistical techniques to estimate this latent structure and get rid of the obscuring "noise." Also, LSA has been considered as a new general theory of acquisition of similarities and knowledge representation, which is helpful in simulating the learning of vocabulary and other psycholinguistic phenomena [2].

Latent Semantic Analysis, from its beginnings to the present, has been implemented in several research topics, for example, in applications to predict a reader's interest in a selection of news articles, based on their reported interest in other articles [3]; in the self-diagnosis of diseases through the description of medical imaging [4]; in applications to detect cyberbullying in teens and young adults [5]; in the field of visual computers by improving techniques for tracking moving people [6]; in applications for the classification of less popular websites [7].

LSA has a computational complexity of $O(d^2k^3)$, where d is the smaller value between the number of documents and the number of terms and k is the number of singular values [8]. LSA takes a considerable amount of time to index and to compute the semantic space, when it is applied to large-scale datasets [9–11].

An introduction for a parallel LSA implementation based on a GPU has achieved acceleration of five to seven times with

large matrices divisible by 16- and 2-fold for matrices with another size. The GPU is being used for the tridiagonalization of matrices, and the routines that compute the eigenvalues and eigenvectors of matrices are still being implemented on the CPU. The results present that the accuracy and speed needed further research in order to produce an effective fully implementable LSA algorithm [12].

A technique called *index interpolation* is presented for a rapid computation of the term-by-document matrix for large documents collections; the associated symmetric eigenvector problem is then solved by distributing its computation among any number of computational units without increasing the overall number of multiplications. The experiments took 42.5 hours to compute 300,000 terms on 16 CPUs [13].

We present a fully *heterogeneous system based on Latent Semantic Analysis* (hLSA), which utilizes the resources of both GPU and CPU architectures to accelerate execution time. Our aim is to compute, reduce, and retrieve information faster than standard LSA versions and to evaluate the accuracy and recall of the information retrieval procedure in the hLSA system. The performance of hLSA has been evaluated, and the results show that acceleration as high as eight times could be achieved, with an accuracy of 88% and a recall of 100%. An early version of the hLSA system has been presented as a poster in the GPU Technology Conference [14].

The rest of the paper is organized as follows. Section 2 introduces the related background of LSA. In Section 3, we present our heterogeneous Latent Semantic Analysis system. Section 4 gives a description of the design of experiments. Section 5 presents the results of the experiments. Finally, Section 6 concludes the work.

2. Background

LSA takes a matrix of a term-by-document (M) and constructs a semantic space wherein terms and documents that are closely associated are placed near one another. Normally, the constructed semantic space has as many dimensions as unique terms. Additionally, instead of working with count data, the entries of matrix M are weighted with a representation of the occurrence of a word token within a document. Hence, LSA uses a normalized matrix which can be large and rather sparse. For this research, two types of weightings scheme are used.

(i) The first is a logarithmic local and global entropy weighting, known as a Log Entropy scheme. That is, if $M_{[w,d]}$ denotes the number of times (frequency) that a word w appears in document d and N is the total number of documents in the dataset, then

$$A_{[w,d]} = \log\left(M_{[w,d]} + 1\right) \times \left(1 + \frac{\sum_{i=1}^{N} P_{[w,i]} \times \log\left(P_{[w,i]}\right)}{\log\left(N\right)}\right), \quad (1)$$

where $P_{[w,i]}$ is the fraction of documents containing the ith term; for example,

$$P\left[w, d\right] = \frac{M_{[w,d]}}{\sum_{i=1}^{N} M_{[w,i]}}. \quad (2)$$

This particular term weighting scheme has been very successful for many LSA studies [15], but other functions are possible.

(ii) The second is a term frequency and inverse document frequency weighting, known as a TF-IDF scheme, which assigns to word w a weight in document d, when N is the total number of documents in the dataset and M is the term frequency by documents matrix; for example,

$$A_{[w,d]} = \frac{M_{[w,d]}}{\sum_{i=1}^{N} M_{[i,d]}} \times \log\left(\frac{N}{\sum_{i=1}^{N} M_{[w,i]}}\right). \quad (3)$$

This particular term weighting scheme has been very successful for many LSA studies [8].

To reflect the major associative patterns in matrix A and ignore the smaller less important influences, a reduced-rank approximation of matrix A is computed using the truncated Singular Value Decomposition [16]. Note that the SVD of the original weighted matrix can be written as

$$A_{[w,d]} = T_{[w,n]} \times S_{[n,n]} \times D^{T}_{[n,d]}, \quad (4)$$

where $A_{[w,d]}$ is the words-by-documents matrix; T is a $[w \times n]$ orthogonal matrix whose values represent the left singular vectors of A; D is a $[d \times n]$ orthogonal matrix whose values represent the right singular vectors of A; and S is a $[n \times n]$ diagonal matrix which contains the singular values of A in descending order. Note that n is the smaller value between the total number of words w and the total number of documents d.

To obtain the truncated SVD denoted by A', it is necessary to restrict SVD matrices to their first $k <$ min (terms, documents) dimensions, as revealed by

$$A'_{[w,k]} = T_{[w,k]} \times S_{[k,k]} \times D^{T}_{[k,d]}. \quad (5)$$

Choosing the appropriate number of dimensions k is an open research problem. It has been proposed that the optimum value of k is in a range from 50 to 500 dimensions, depending on the size of the dataset. As described in [17], if the number of dimensions is too small, significant semantic content will remain uncaptured, and if the dimension is too large, random noise in word usage will be remodeled. Note that the truncated SVD represents both terms and documents as vectors in k-dimensional space.

Finally, for information retrieval purposes, the k-dimensional semantic space is used. Thus, the terms of a user query are folded into the k-dimensional semantic space to identify a point in the space. This can be accomplished by parsing a query into a vector denoted by q whose nonzero values correspond to the term weights of all unique valid words of the user query. Then, the query folding process denoted by q' can be represented as

$$q'_{[k]} = q^{T}_{[w]} \times T_{[w,k]} \times S^{-1}_{[k,k]}. \quad (6)$$

Then, this q' vector can be compared with any or all documents/terms vectors of the k-dimensional semantic space. To

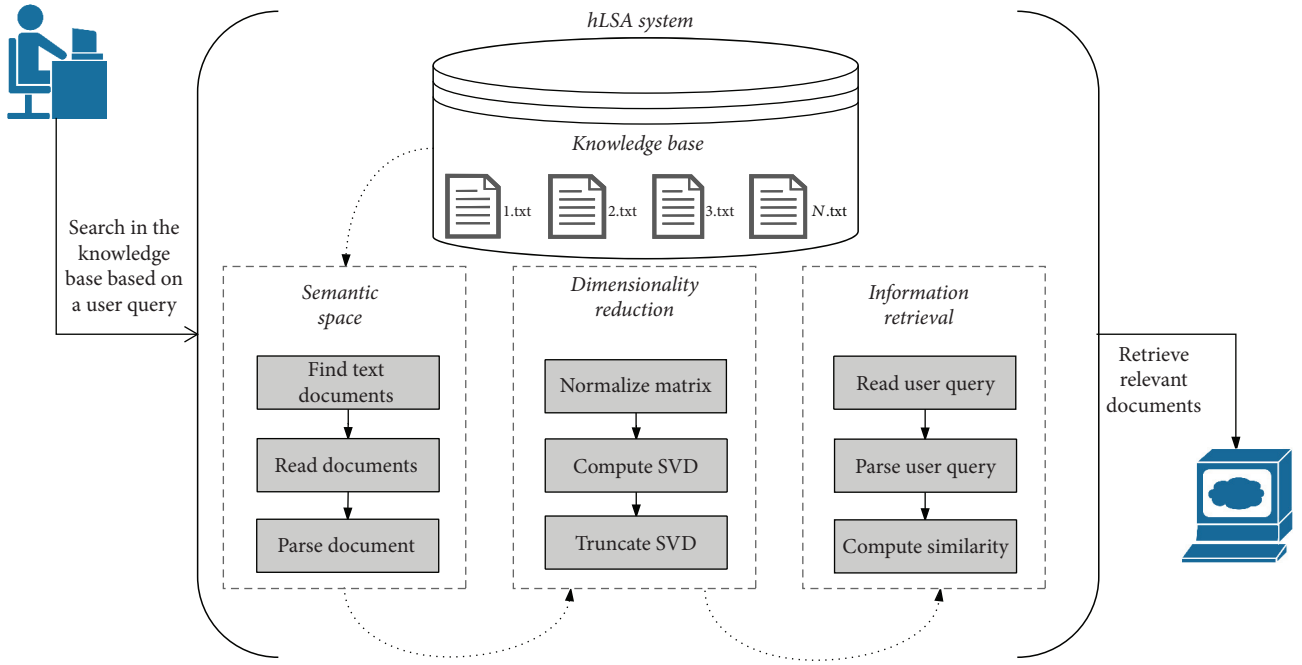


Figure 1: The hLSA system presents three stages: semantic space, dimensionality reduction, and information retrieval. Also, the hLSA system works with a user query, as well as a knowledge base, and presents the relevant documents.

compare vectors, the dot product or cosine between points is used. For example,

$$\cos_{(v,q')} = \frac{\vec{v} \times \vec{q'}}{\left|\vec{v}\right| \times \left|\vec{q'}\right|}, \tag{7}$$

where v is a vector representation of the k-dimensional space. LSA proposes the retrieval of information in two ways: (1) by establishing a minimum value of similarity, for example, all the similarities greater than 0.90, and (2) by obtaining the top-ranked values of similarity, for example, the top 10 and the top 5.

3. Heterogeneous Latent Semantic Analysis System

The hLSA system has several technical challenges: (1) how to construct the *semantic space* using the multi-CPU architecture to speed up the text processing; (2) how to *reduce the dimensionality* of the term-by-document matrix using the GPU architecture to accelerate the matrices processing; and (3) how to *retrieve information* from the semantic space using GPU mechanisms to speed up the matrix and text computations. Figure 1 presents the proposed hLSA system for constructing, reducing, and retrieving relevant documents using heterogeneous architectures.

Notably, one of the features of the hLSA system is the use of GPU architecture, which is able to execute SIMD (Single Instruction, Multiple Data) operations, such as matrix and vector multiplications, very efficiently with high parallelism. In addition, we take advantage of the multi-CPU architecture, which is able to execute SIMD operations, such as *map* and *reduce* functions. On the one hand, the hLSA system utilizes the GPU architecture to solve matrix operations, especially in the stage of *dimensionality reduction* and *information retrieval*. On the other hand, the hLSA system utilizes CPU and multi-CPU architectures to solve text processing, in the *semantic space stage*. Thus, with the use of these mechanisms the hLSA system is able to achieve accelerated performance.

3.1. Semantic Space Stage. The input for constructing the semantic space is a knowledge base; this is represented by a dataset of raw texts, which is generally stored in the disk drive of a personal computer. Therefore, the hLSA system first needs to find the text documents. This execution has no major computational cost and therefore is executed by the CPU in a sequential manner. As a result, a document list is generated which is loaded into the main memory of the CPU.

From the document list obtained, the hLSA system begins to read each document text and starts the preprocessing, for instance, eliminating the blank spaces at the beginning and the end and, also, eliminating special characters such as *"?!(),;:.`* and ignoring common words known as stopwords, such as *"the"; "for"; and "you."* As a result, the hLSA system generates a list of preprocessed texts, which is stored in the main CPU memory. This procedure is performed on CPU runtime in a sequential manner.

Then, the hLSA system starts to generate the count matrix of term frequency by documents denoted as A, where rows represent the terms and columns represent the documents.

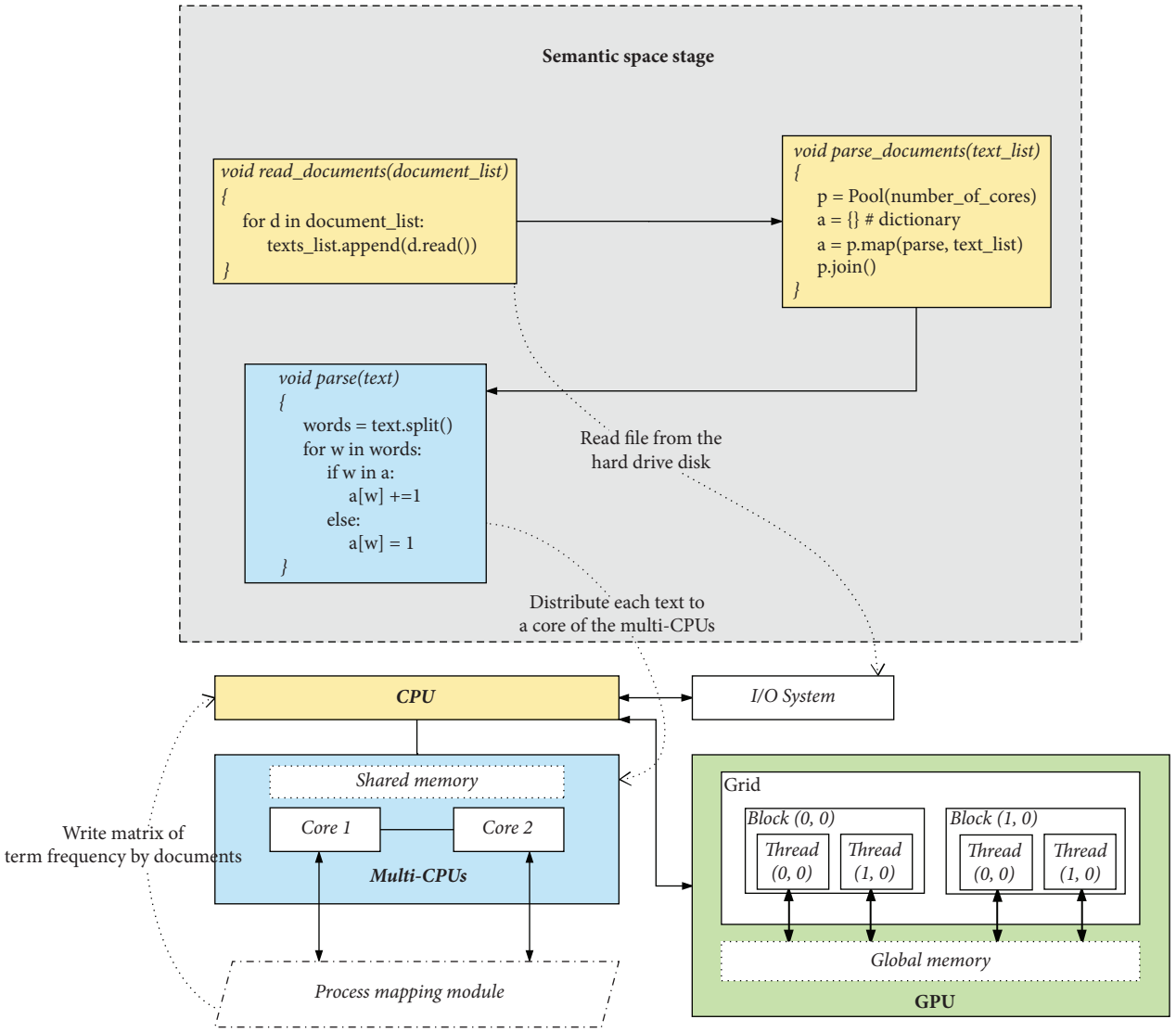


Figure 2: The architecture used in the semantic space stage of the hLSA system. The blocks in light yellow represent the principal procedures executed by the CPU, the blocks in light blue represent the procedures executed by multi-CPUs, and the blocks in light green represent the procedures executed by GPU, respectively. The multi-CPUs include shared memory models for multiprocessing programming.

In order to do this, the hLSA system analyzes the list of preprocessed documents.

Therefore, the hLSA system takes each preprocessed text of the list and splits the content when a blank space appears. For example, the text "This is an example" is split at each blank space. As a result, a list of words: ["This"; "is"; "an"; "example"] is generated.

Next, the hLSA system iterates the new list of words and for each time the word appears in a document the hLSA system adds one to the corresponding cell value of matrix A. This procedure has a high computational cost.

Thus, we implement a parallel multiprocessing model using a multi-CPU architecture. Each CPU core is in charge of processing an element of the preprocessed text of the list as shown in Figure 2. As a result of the semantic space stage, the hLSA system generates in parallel and sequential procedures matrix A, where each cell value corresponds to the frequency that a word appears in document of the knowledge base.

3.2. Dimensionality Reduction Stage. The hLSA system uses the CUDA programming model in order to process the dimensionality reduction of matrix A. As a consequence, the hLSA system uses two dimensions (x, y) of the CUDA architecture; the x dimension is associated with the rows of matrices and the y dimension with the columns of matrices. Each block of the CUDA architecture executes 32 concurrent threads and the total number of blocks executed depends on the total size of the matrix to be processed.

The hLSA system needs to normalize matrix A. Therefore, two term weighting schemes are used to normalize matrix A: the Log Entropy and the Term Frequency-Inverse Document Frequency schemes. In order to compute the term weighting

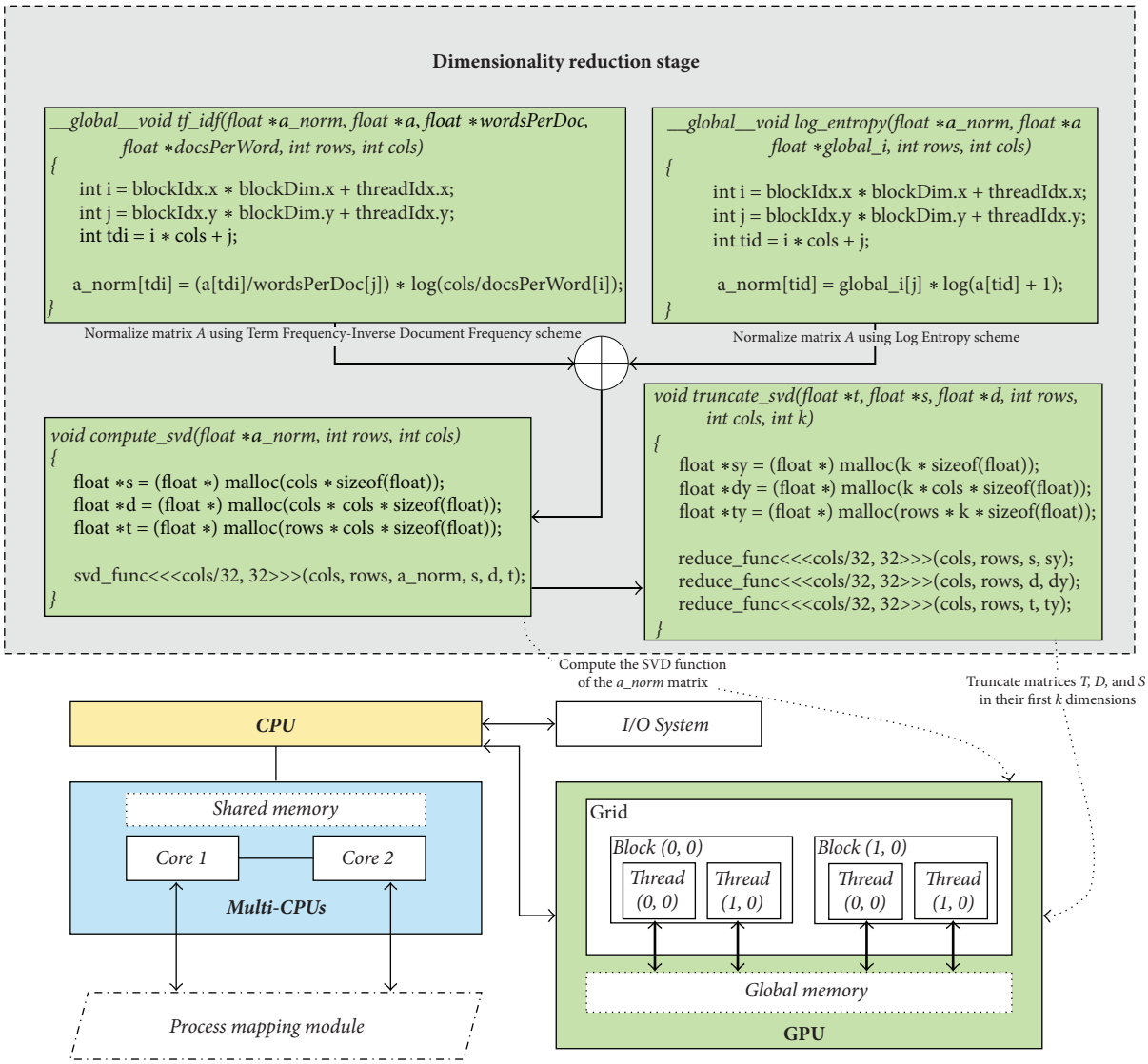


Figure 3: The architecture used in the dimensionality reduction stage of the hLSA system. The blocks in light yellow represent the principal procedures executed by the CPU, the blocks in light blue represent the procedures executed by multi-CPUs, and the blocks in light green represent the procedures executed by GPU, respectively. The GPU executions include CUDA kernels and CUDA functions using floating-point numbers.

schemes, the hLSA system loads matrix A into the GPU global memory and, then, executes the CUDA kernel functions and the resulting matrix is saved into the GPU global memory as *A_norm*.

The hLSA system uses the SVD algorithm, which decomposes matrix *A_norm* into three matrices. Matrix T is an orthogonal matrix of left singular vectors. Matrix S is a diagonal matrix and contains the singular values ordered by magnitude. Matrix D^T is an orthogonal matrix of right singular vectors. Therefore, the hLSA system needs to reserve space in the GPU global memory for the resultant matrices T, S, and D^T. Then, it executes the *compute_svd* CUDA function. At the end of the execution the hLSA system frees the GPU memory space of matrix *A_norm*.

Finally, the hLSA system truncates the resultant matrices of the SVD function into their first k dimensions. Therefore, it executes the *truncate_svd* CUDA function. As a result, the hLSA system generates three matrices denoted by *Sy*, *Ty*, and *Dy*. At the end of the execution the matrices are copied into the main memory of the CPU and then the GPU global memory is freed. The dimensionality reduction stage of the hLSA system is shown in Figure 3.

3.3. Information Retrieval Stage. The hLSA system uses a user query in order to retrieve the relevant information from the semantic space. Therefore, the hLSA system creates a one-dimensional vector that is denoted by q, with a size equal to the total number of terms obtained in the semantic

space. Then, it analyzes the query; namely, if a word of the query corresponds to a word in the semantic space, the hLSA system increments by one the corresponding cell value in the vector q. This execution has no major computational cost and therefore is executed by the CPU in a sequential manner.

Then the hLSA system folds the vector q into the semantic space by using the query folding equation (6). Therefore, the hLSA system executes the *query_folding* CUDA function using the matrices *Sy*, *Ty*, and *Dy* loaded in the GPU global memory. As a result, the hLSA system generates a vector q', whose values correspond to the terms weights of all the unique words of the query. Then, the hLSA system compares the vector q' with all document vectors inside the semantic space in order to find their similarity.

To compute the similarity, the hLSA systems uses the cosine similarity equation (7). Hence, it executes the *cosine_similarity* CUDA function using the vector q' and the document vectors of matrix *Dy*. Ultimately, the hLSA system presents the most relevant documents based on the similarity value between the vector q' and each document vector. To present the results, the hLSA system establishes a threshold of similarity, for example, all the similarities greater than 0.90, and/or the hLSA systems obtain the top-ranked values of similarity, for example, the top 20, the top 10, and the top 5. The information retrieval stage of the hLSA system is shown in Figure 4.

4. Experiments

The objective of this section is to evaluate the performance of the proposed hLSA system. The design of the experiments is detailed in (1) Definition of the Knowledge Base, (2) Definition of Dataset Size, (3) Definition of Stopword Document, (4) Definition of k Dimensions, (5) Definition of Use Cases, and (6) Evaluation Methods.

(1) Definition of Knowledge Base. The experiments conducted with the hLSA system use documents of the Open Access Subset of PubMed Central (PMC) as the knowledge base of the experiments. PMC is an online digital database of freely available full-text biomedical literature. Extensively used in the Text REtrieval Conference (TREC) (the Text REtrieval Conference (TREC), cosponsored by the National Institute of Standards and Technology (NIST) and the US Department of Defense, can be found at http://trec.nist.gov). The text of each article in the open access subset is represented as an NXML file, which is an XML format used by NML Journal Archiving and Interchange Tag Set (the Journal Archiving and Interchange Tag Set defines elements and attributes that describe the content and metadata of journal articles, including research and nonresearch articles, letters, editorials, and book and product reviews).

The documents are named and identified by a unique number (PMCID) defined by an <*article-id*> element. Also, each NXML file has XML elements such as <*article-title*>, <*doi*>, < *abstract*>, <*fig*>, and <*publisher-name*>,. For experimental purposes, we only use the abstract of the document. This information is extracted from each NXML and copied to a new text file (.txt). Each text file is stored in a working directory with a file name equal to the PMCID identifier.

(2) Definition of Dataset Size. The experiments conducted with the hLSA system use a total of five thousand documents of the PMC database. In order to observe the variation of the total time of execution of the hLSA system with large-scale datasets, we partition the five-thousand documents into twelve subsets of different sizes ranging in size from five hundred to five thousand documents. In Table 1, we present a summary of each subset. The first column represents the total number of documents. The second column represents the total number of words in a dataset after preprocessing the raw texts. The third column presents the total number of unique words in the dataset. The fourth column represents the total number of elements in the matrix of term-by-document and the last column represents the size in megabytes of the term-by-document matrix.

(3) Definition of Stopword Document. The experiments conducted with the hLSA system uses a document file of stopwords to exclude certain single words like determinants, that is, "the," "a," "an," and "another"; coordinating conjunctions, that is, "for," "but," "or," and "so"; prepositions, that is, "in," "under," and "towards"; pronouns forms, that is, "me," "my," "myself," and "you"; verb forms, that is, "am," "is," "was," and "do"; among other words. In total, one thousand nine hundred eighty-two words were used.

(4) Definition of k Dimensions. The experiments conducted with the hLSA uses several values of k to find the appropriate value with which the hLSA system is more accurate. In total, twenty values of k were used ranging from twenty-five to five hundred dimensions in increments of twenty-five.

(5) Definition of Use Cases. The experiments in the hLSA system were conducted with three use cases: (a) bipolar disorders, (b) lupus disease, and (c) topiramate weight-loss.

(a) Bipolar Disorders. Bipolar disorders could be presented in patients in their middle age with obese disorders, who struggle with their weight and eat when feeling depressed, excessively anxious, agitated, and irritable and when having suicidal thoughts and/or difficulty sleeping.

(b) Lupus Disease. Lupus disease could be presented in patients with alopecia, rash around their nose and checks, delicate nonpalpable purpura on their calves, and swelling and tenderness of their wrists and ankles and with normocytic anemia, thrombocytopenia, and being positive for protein and RBC casts.

(c) Topiramate Weight-Loss. Some research reports have shown that the anticonvulsant topiramate causes weight-loss in various patient groups. Patients have lost a mean of 7.75 kg in six months and at twelve months patients have lost a mean of 9.61 kg.

(6) Evaluation Methods. We have defined the size of the subsets, the number of k dimensions, and the use cases for our

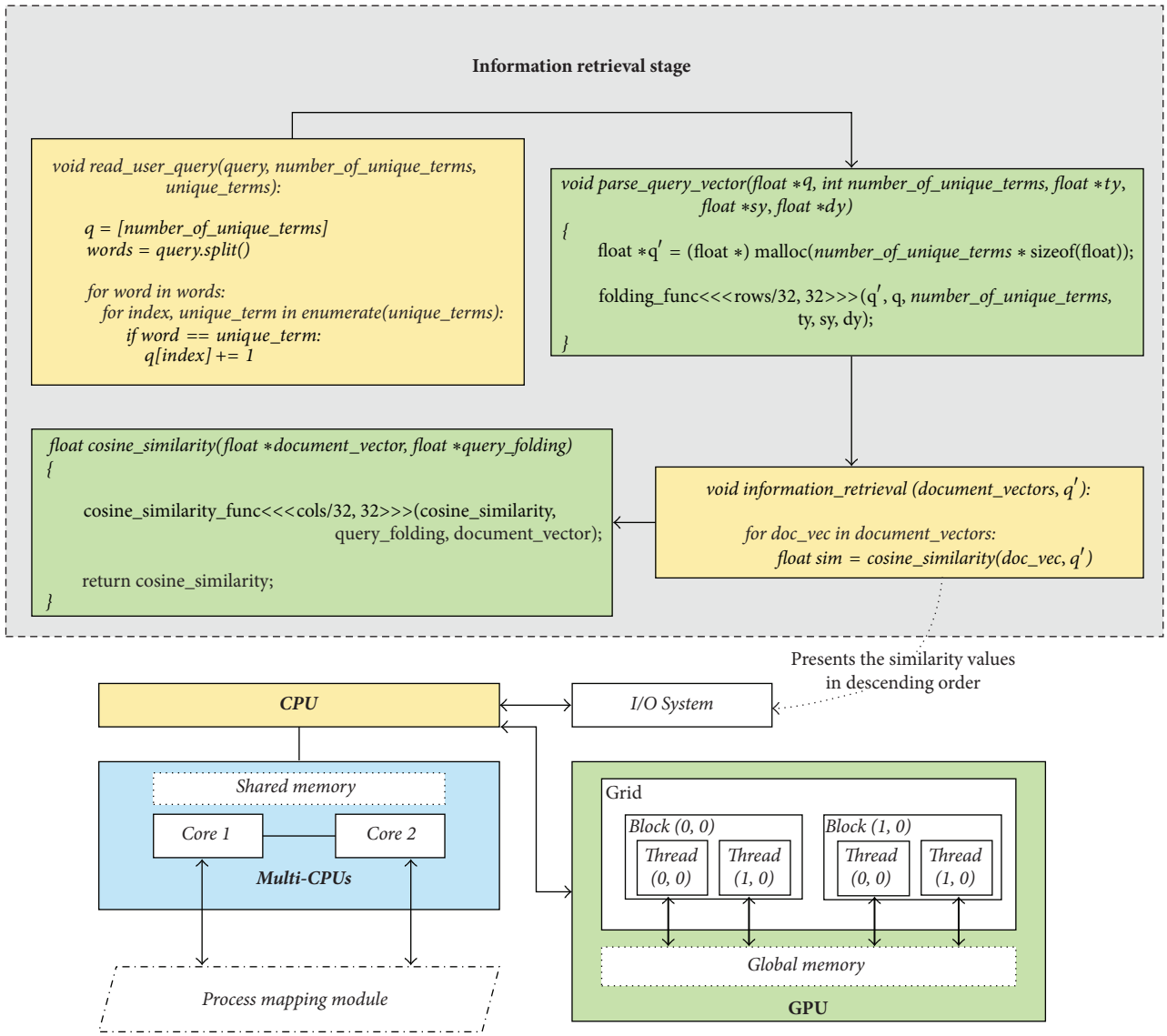


Figure 4: The architecture used in the information retrieval stage of the hLSA system. The blocks in light yellow represent the principal procedures executed by the CPU, the blocks in light blue represent the procedures executed by multi-CPUs, and the blocks in light green represent the procedures executed by GPU, respectively. The GPU executions include CUDA kernels and CUDA functions using floating-point numbers.

experiments. Hence, to evaluate our hLSA system, we have used two methods: (a) we have built an LSA system using CPU sequential-only architecture, in order to specifically evaluate the acceleration reached by the hLSA system. Thus, we have compared the execution time of the hLSA and LSA (CPU sequential-only) system. (b) We have evaluated the accuracy and recall of the information retrieval stage.

(a) Time Execution. We have executed the hLSA system and LSA systems a total of four hundred eighty times to complete all the experiments with the different subsets, k dimensions, weighting schemes, and use cases. Therefore, in order to compare the execution time of the hLSA system versus the LSA system, we have collected the execution time of each procedure involved in each stage of both systems. Hence, we have presented in Section 4 the mean values of these executions.

(b) Similarity Accuracy. We have executed the hLSA system and LSA systems a total of four hundred eighty times to complete all the experiments with the different subsets, k dimensions, weighting schemes, and use cases. Therefore, in order to evaluate the information retrieval process of the hLSA system, we have presented the top 10 ranked

Table 1: Subsets used in the experiments for the hLSA system.

Number of documents	Number of words	Number of unique words	Number of elements in matrix	Size of matrix (Mb)
500	13856	6819	3,409,500	27
750	18194	9039	6,779,250	54
1000	22107	11026	11,026,000	88
1250	25264	12587	15,733,750	126
1500	28251	14066	21,099,000	169
1750	31282	15596	27,293,000	218
2000	34024	16990	33,980,000	272
2500	39239	19650	49,125,000	393
3000	44114	22039	66,117,000	529
3500	48622	24284	84,994,000	680
4000	53003	26373	105,492,000	844
5000	61466	30273	151,365,000	1211

documents, and we have compared the similarity values generated by the hLSA system in each experiment with the subset of five thousand documents.

5. Results

In this section, we present the results of the experiments with the hLSA system. First, we show the execution time for the semantic space stage. Second, we present the execution time for the dimensionality reduction stage. Third, we show the execution time for the information retrieval stage. Fourth, we present the execution time for the full hLSA system. Finally, the results to evaluate the similarity accuracy and recall of the information retrieval stage are presented.

The hLSA system has been executed on a Linux OpenSUSE 13.2 (3.16.6-2 kernel) system with Core i7 Intel processors at 2.50 GHz with 16 GB DDR3L of main memory and 4 cores with 8 threads for each core and a NVIDIA GeForce GTX 970M. The GPU has 10 SMs with 128 CUDA cores for each SM, a total of 1280 CUDA cores. The GPU has a total amount of global memory of 3064 Mbytes and a clock speed of 2.5 GHz. Also, the maximum number of threads per multiprocessor is 2048, and the maximum number of threads per block is 1028. The GPU maximum dimension size of thread block is "1024, 1024, and 64" in the (x, y, z) dimension.

(1) Runtime Results: Semantic Space Stage. The procedures executed in the semantic space stage are to read the stopwords file, find documents in the hard disk drive, read documents from the hard disk drive, parse documents into the main CPU memory, and build the matrix of terms frequency by documents. The hLSA system uses the benefits of multi-CPU architecture to parse the documents into the main CPU memory. Table 2 present the execution time of procedures executed in CPU sequential-only. In Table 3, we compare the execution time of the *Document Parser* procedure using CPU sequential-only and multi-CPU architecture with four, eight, and sixteen processors. Additionally, in Figure 5, we present the overall execution time using CPU and multi-CPU

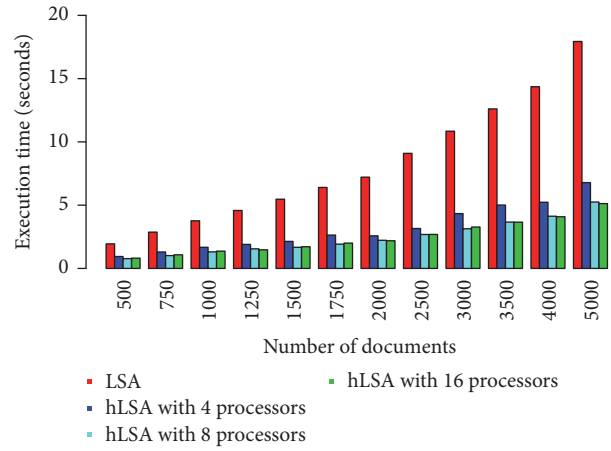


Figure 5: Results of the overall execution time in seconds of the semantic space stage using CPU sequential-only architecture and multi-CPU architecture with four, eight, and sixteen processors.

architectures. The execution times were obtained as mean values of the four hundred eighty executions and are presented in milliseconds.

The procedures with high execution time in the LSA system are *Read Documents* and *Document Parser*. On the other hand, the procedure with the highest execution time in hLSA system is *Read Documents*. This is due to the fact that the *Read Documents* procedure has to access the drive disk in order to read the text documents, and the time to access the drive disk depends on the hardware. Therefore, there is a bottleneck which is limited by the hardware of disk I/O system, which goes beyond the scope of our work.

Moreover, we have found a substantial improvement in the *Document Parser* procedure in the hLSA system using multi-CPU architecture. Acceleration of 2.50 times is seen with five hundred documents and maximum acceleration of 2.90 times with five-thousand documents using four processors. When using eight processors, acceleration of 3.00 times

Table 2: Results of the *CPU sequential-only* procedures in milliseconds of the semantic space stage in the hLSA system.

Number of documents	Read stopwords (ms)	Find documents (ms)	Read documents (ms)	Build matrix (ms)
500	27.016	186.764	4.044	41.646
750	27.016	187.885	5.929	61.891
1000	27.016	195.235	7.845	82.295
1250	27.016	196.450	9.752	105.363
1500	27.016	189.951	12.333	121.197
1750	27.016	191.454	13.816	152.200
2000	27.016	198.723	16.475	164.590
2500	27.016	194.696	19.134	208.550
3000	27.016	192.528	23.034	250.311
3500	27.016	191.431	26.621	297.541
4000	27.016	198.942	34.243	339.805
5000	27.016	189.929	37.598	434.999

Table 3: Results of the *Document Parser* procedure in milliseconds of the semantic space stage in the LSA system using CPU sequential-only architecture and in the hLSA system using multi-CPU architecture with four, eight, and sixteen processors.

Number of documents	Document Parser Procedure			
	LSA (ms)	hLSA with 4 processors (ms)	hLSA with 8 processors (ms)	hLSA with 16 processors (ms)
500	1681.982	676.587	504.683	549.639
750	2587.080	1003.574	720.285	780.755
1000	3453.187	1351.907	984.871	1037.802
1250	4241.641	1549.046	1205.027	1128.528
1500	5121.449	1770.975	1303.507	1347.592
1750	6016.346	2236.582	1522.095	1596.852
2000	6806.595	2158.235	1789.595	1766.482
2500	8645.503	2667.631	2212.338	2215.629
3000	10351.710	3799.512	2594.976	2740.206
3500	12066.547	4420.248	3077.626	3065.884
4000	13762.060	4582.366	3495.762	3442.618
5000	17247.301	6029.288	4477.206	4352.604

is seen with five-hundred documents and maximum acceleration of 3.85 times with five thousand documents. Finally, for sixteen processors, acceleration of 3.30 times is seen with five-hundred documents and maximum acceleration of 3.96 times with five thousand documents. Thus, a better overall performance is found with sixteen processors and greater acceleration when the document corpus increases.

(2) Runtime Results: Dimensionality Reduction Stage. The weighting schemes Log Entropy and Term Frequency-Inverse Document Frequency have a high computational cost in the LSA system, as shown in Table 4. The hLSA system reaches acceleration of thirty times for the smaller datasets (five hundred to two thousand five hundred) and acceleration of eight hundred fifty times for the larger datasets (three thousand to five thousand) using the TF-IDF scheme. Also, using the Log Entropy scheme the hLSA system has reached acceleration of sixty-two times for the smaller datasets and acceleration of one thousand fifty-two times for the bigger datasets.

The computational complexity of Log Entropy is $0(d)$ and that of TD-IDF schemes is $O(dw)$, where w is the total number of individual words and d is the total number of documents in the corpus. As shown in the results, the TF-IDF scheme in the hLSA system has reduced the execution time from more than 160 seconds to less than 0.20 seconds, and the Log Entropy scheme in the hLSA system has reduced the execution time from more than 321 seconds to less than 0.20 seconds.

Meanwhile, an exact SVD has a computational complexity of $O(dw^2)$, and this is expensive for large matrices. In the hLSA system, we implement the GPU-SVD function, which

Table 4: Results of the execution time in the hLSA system using GPU-accelerated procedures and in the LSA system using sequential-only procedures for the Log Entropy scheme and the TF-IDF scheme.

Number of documents	TF-IDF		Log Entropy	
	hLSA (ms)	LSA (ms)	hLSA (ms)	LSA (ms)
500	117.814	3676.445	118.735	7401.700
750	118.319	7315.378	121.697	14300.521
1000	118.622	11845.949	121.143	23370.072
1250	123.446	17255.605	124.997	33131.246
1500	127.632	22982.827	132.696	45446.027
1750	130.057	29501.134	129.801	57330.141
2000	134.766	36193.939	135.845	71768.188
2500	143.089	54128.704	154.173	103304.719
3000	148.864	70769.040	154.600	139719.203
3500	161.621	92035.205	161.590	179290.609
4000	167.414	114055.179	175.994	233267.328
5000	189.126	160822.907	206.725	320797.375

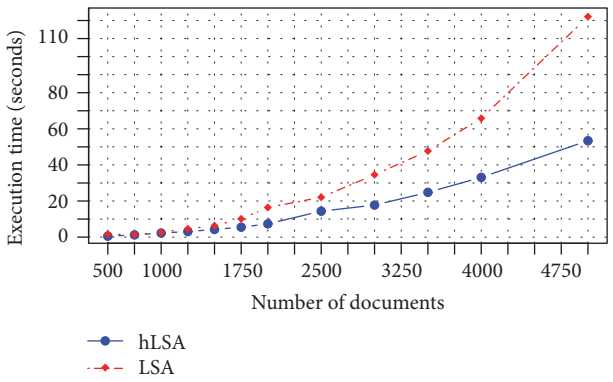


Figure 6: Results of the execution time of the SVD procedure applied in the hLSA system using GPU-accelerated procedures and in the LSA system using CPU sequential-only procedures.

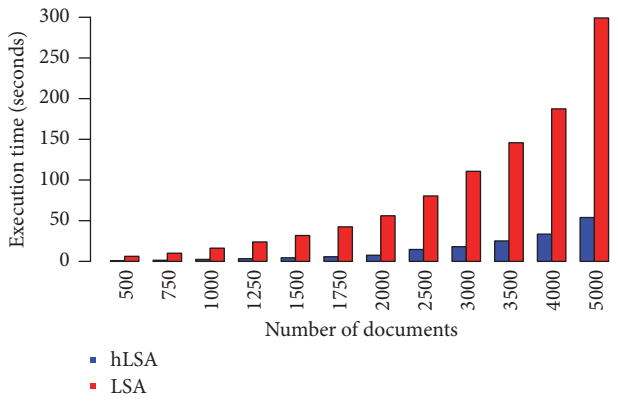


Figure 7: Results of a complete execution of dimensionality reduction stage with k-dimension equal to 300 and TF-IDF weighting scheme applied in the hLSA system using GPU-accelerated procedures and the LSA system using CPU sequential-only procedures.

gives us acceleration of up to three times compared to the LSA system. In Figure 6, we present a comparison of the SVD execution time between the hLSA system and the LSA system.

In the smallest dataset, we measure an execution time of 1.60 seconds in the LSA system and, in the largest dataset, we measure an execution time of 122.10 seconds. On the other hand, in the hLSA system, we measure an execution time of 0.50 seconds in the smallest dataset and an execution time of 53.40 seconds in the largest dataset. Hence, we reach acceleration of two to three times in most cases.

Consequently, the SVD procedure with the NVIDIA Visual Profiler was analyzed. We found that SVD generates about 77 percent of all the time-processing and launches around six thousand seven hundred thirty kernel instances of SVD in CUDA architecture. In addition, the profiler showed an optimization problem in memory access for computing the SVD procedure. For shared memory, the SVD utilizes a low bandwidth of 39.193 GB/s for 273,163 total load/store transactions, and, for device memory, the SVD utilizes a low bandwidth of 42.814 GB/s for 1,193,617 read/write transactions.

Meanwhile, the computational cost for the truncated SVD procedure takes less than one second in the LSA system using the biggest dataset. However, in the hLSA system, we have accelerated up to nine times for the smaller datasets and up to four times for the larger datasets. In Table 5, we present the results of an execution with the dataset of two thousand documents. In this execution, our results present acceleration of four times.

We have presented the results of procedures: Log Entropy, TF-IDF, SVD, and Truncated SVD. As shown in results, the dimensionality reduction stage has the most computational cost in our experiments. Therefore, we now present in Figure 7 the results of a complete execution of dimensionality reduction stage with $k = 300$ and the TF-IDF weighting scheme. We obtain for the smaller datasets acceleration from

Table 5: Results of execution time in a truncated SVD procedure applied in the hLSA system using GPU-accelerated procedures and the LSA system using CPU sequential-only procedures for 2000 documents.

Number of k	hLSA				LSA			
	Matrix S (ms)	Matrix D (ms)	Matrix T (ms)	Total (ms)	Matrix S (ms)	Matrix D (ms)	Matrix T (ms)	Total (ms)
25	0.158	7.241	59.635	67.033	0.094	11.151	254.472	265.717
50	0.134	6.540	59.228	65.902	0.113	12.756	251.627	264.496
75	0.144	6.870	59.576	66.590	0.127	10.842	250.042	261.011
100	0.149	4.216	60.443	64.808	0.118	10.594	251.530	262.242
125	0.118	5.064	60.644	65.826	0.115	11.048	262.973	274.136
150	0.176	4.657	61.183	66.016	0.127	10.387	261.292	271.806
175	0.113	6.631	61.259	68.002	0.120	10.626	257.150	267.896
200	0.162	4.195	62.297	66.653	0.135	10.148	265.645	275.928
225	0.174	4.204	63.519	67.897	0.140	10.298	266.845	277.283
250	0.177	4.941	63.100	68.219	0.142	10.090	256.517	266.749
275	0.177	4.599	65.915	70.691	0.140	10.917	257.991	269.048
300	0.174	4.829	64.671	69.675	0.147	10.869	269.890	280.906
325	0.174	6.413	64.655	71.242	0.150	10.774	254.650	265.574
350	0.178	4.269	67.372	71.818	0.149	10.570	253.184	263.903
375	0.157	4.843	68.000	72.999	0.194	10.630	252.340	263.164
400	0.209	5.808	66.520	72.537	0.181	10.776	252.570	263.527
425	0.209	6.443	66.587	73.239	0.174	12.384	265.330	277.888
450	0.243	5.847	69.124	75.213	0.162	10.868	258.973	270.003
475	0.198	4.866	70.410	75.474	0.169	10.796	265.007	275.972
500	0.201	5.765	70.373	76.338	0.172	10.656	256.289	267.117

Table 6: Results of the execution time of query parser procedure applied in the hLSA system with the three use cases: *bipolar disorders, lupus disease,* and *topiramate weight-loss*.

Number of documents	Bipolar disorders (sec)	Lupus disease (sec)	Topiramate weight-loss (sec)
500	7.81	6.22	27.13
750	10.28	8.20	35.99
1000	12.62	9.96	44.06
1250	14.22	11.39	50.22
1500	15.92	12.88	56.38
1750	17.71	14.11	62.29
2000	19.34	15.38	68.33
2500	22.22	17.81	78.71
3000	25.05	20.08	88.35
3500	27.51	22.09	97.49
4000	30.11	23.93	107.81
5000	34.46	27.50	121.37

six to ten times, and for the larger datasets acceleration from five to eight times.

(3) Runtime Results: Information Retrieval Stage. We have implemented the hLSA system not just for indexing documents, but also for information retrieval. Thus, we now present the results of the execution time in information retrieval procedures, which are as follows: query parser, query folding, and cosine similarity.

We obtain the process time for the query parser procedure, which has been implemented in CPU sequential-only for both systems hLSA and LSA. We present, in Table 6, the results of the query parse procedure on each dataset using the three use cases.

Table 7: Results of execution time for the query folding and cosine similarity procedures are applied in the hLSA system using GPU-accelerated procedures and the LSA system using sequential-only procedures for 5000 documents and topiramate weight-loss use case.

Number of k	hLSA		LSA	
	Query folding (ms)	Cosine similarity (ms)	Query folding (ms)	Cosine similarity (ms)
175	1.989	585.023	26.348	245.986
200	3.484	620.742	2.401	246.752
225	3.979	621.718	14.623	247.087
250	3.929	614.037	6.367	248.554
275	4.065	595.509	19.193	256.337
300	4.539	623.648	7.603	255.993
325	4.900	595.137	4.495	259.016
350	4.805	589.494	30.019	262.982

Table 8: Results of the total execution time in the hLSA system versus the LSA system. Results include all procedures using 5000 documents, $k = 300$, Log Entropy, TF-IDF, and four, eight, and sixteen multi-CPUs.

	Bipolar disorders		Lupus disease		Topiramate weight-loss	
	Log Entropy (sec)	TF-IDF (sec)	Log Entropy (sec)	TF-IDF (sec)	Log Entropy (sec)	TF-IDF (sec)
hLSA	96.29	110.16	87.42	86.40	181.93	180.25
LSA	770.50	610.82	756.33	590.01	848.97	689.12
Acceleration	8.00	5.54	8.65	6.83	4.67	3.82

The process with the highest computational cost in terms of the information retrieval stage is the query parser procedure. For the biggest dataset with the use case of bipolar disorders it took around 34 seconds, with the use case of lupus disease it took around 27 seconds, and with the use case of topiramate weight-loss it took more than 120 seconds. This is due to the number of unique words in the queries. The first use case has an average of 31 unique words, the second use case has an average of 22 unique words, and the last use case has an average of 136 unique words.

Additionally, we now present the result times for the query folding procedure and cosine similarity procedure. We present the best results using the use case of topiramate weight-loss. These procedures do not have a high computational cost; in most cases, the execution time is less than one second for all datasets in the hLSA and LSA system. Table 7 shows several results of query folding and cosine similarity procedures.

(4) Runtime Results: hLSA versus LSA. We present in Table 8 the overall execution time for the hLSA system versus the LSA system with five thousand documents, k-dimension equal to three hundred, three use cases, and two weighting schemes. Also, for use case of bipolar disorders, we present results with four processors; for use case of lupus disease, we present results with eight processors; and, for use case of topiramate weight-loss, we present results with sixteen processors. We reach overall acceleration of five to eight times.

(5) Information Retrieval Results. To evaluate the accuracy, we compare the documents retrieved by the hLSA system based on a text query related to each use case versus the relevant documents defined by the experts. The experts define that the most relevant documents for use case of bipolar disorders are the articles with identifiers 1087494, 1434505, and 2031887; for use case of lupus disease the most relevant documents are the articles with identifiers 1065341, 1459118, and 1526641; and for use case of topiramate weight-loss the most relevant document is the article with identifier 1087494. We present the results of the similarities for each relevant document covering all the documents of the knowledge base with two weighting schemes and twenty values of k, for use case of bipolar disorders in Figure 8, for use case of lupus disease in Figure 9, and for use case of topiramate weight-loss in Figure 10.

As shown in the results, the best similarities found in the experiments with the hLSA system for use case of bipolar disorders are found with values of $k = 50$ and accuracy = 0.88, for use case of lupus disease are found with values of $k = 25$ and accuracy = 0.56, and for use case of topiramate weight-loss are found with values of $k = 25$ and accuracy = 0.98. Anyhow, the similarities values are normalized in the range from one hundred fifty to three hundred.

Moreover, we present the results of the top ten documents with use case of bipolar disorders in Table 9; with use case of lupus disease, we show results in Table 10, and, for use case of topiramate weight-loss, we present the results in Table 11.

Table 9: Results of the information retrieval process in the hLSA system with use case of bipolar disorders, using Log Entropy and TF-IDF weighting scheme with 500 documents and k = 75, 100, 125, and 150; 2000 documents and k = 175, 200, 225, and 250; 4000 documents and k = 275, 300, 325, and 350.

	Tf-Idf		Log Entropy			Tf-Idf		Log Entropy	
	File	Similarity	File	Similarity		File	Similarity	File	Similarity
				500 documents					
		k = 75					k = 100		
1	**1434505.txt**	0.62	3662141.txt	0.53	1	**1434505.txt**	0.59	**1087494.txt**	0.53
2	3714066.txt	0.51	3416914.txt	0.47	2	3714066.txt	0.48	3425832.txt	0.44
3	3181786.txt	0.39	**1087494.txt**	0.47	3	3821851.txt	0.37	3650299.txt	0.42
4	3103169.txt	0.38	3395358.txt	0.42	4	3181786.txt	0.35	3662141.txt	0.41
5	3821851.txt	0.35	3574453.txt	0.42	5	3662141.txt	0.34	2776598.txt	0.39
6	3425832.txt	0.34	2776598.txt	0.41	6	3585785.txt	0.31	3078090.txt	0.37
7	3662141.txt	0.32	3078747.txt	0.40	7	2933487.txt	0.31	**2031887.txt**	0.36
8	2933487.txt	0.32	3425832.txt	0.39	8	3425832.txt	0.30	3483824.txt	0.33
9	3585785.txt	0.30	3483824.txt	0.39	9	**1087494.txt**	0.28	3443410.txt	0.33
10	3574453.txt	0.29	3650299.txt	0.38	10	3785904.txt	0.28	3395358.txt	0.32
		k = 125					k = 150		
1	**1434505.txt**	0.55	**1087494.txt**	0.53	1	**1434505.txt**	0.49	**1087494.txt**	0.57
2	3714066.txt	0.46	3425832.txt	0.45	2	3714066.txt	0.40	3650299.txt	0.45
3	2933487.txt	0.33	3650299.txt	0.41	3	2933487.txt	0.34	3425832.txt	0.43
4	3821851.txt	0.31	3662141.txt	0.38	4	3483824.txt	0.33	3483824.txt	0.37
5	**1087494.txt**	0.28	3483824.txt	0.34	5	3416914.txt	0.33	3443410.txt	0.33
6	1779284.txt	0.28	3078090.txt	0.34	6	3662141.txt	0.33	3078680.txt	0.33
7	3181786.txt	0.27	3443410.txt	0.32	7	3078747.txt	0.32	3416914.txt	0.32
8	3662141.txt	0.26	**2031887.txt**	0.30	8	1779284.txt	0.31	**1434505.txt**	0.31
9	3585785.txt	0.26	3078680.txt	0.29	9	**1087494.txt**	0.31	3662141.txt	0.29
10	3785904.txt	0.26	3416914.txt	0.28	10	**2031887.txt**	0.30	3180447.txt	0.27
				2000 documents					
		k = 175					k = 200		
1	**1434505.txt**	0.53	2928181.txt	0.47	1	**1434505.txt**	0.52	3662141.txt	0.42
2	2928181.txt	0.50	2758213.txt	0.45	2	3714066.txt	0.44	2758213.txt	0.40
3	2758213.txt	0.47	3662141.txt	0.43	3	2758213.txt	0.41	2928181.txt	0.38
4	3714066.txt	0.46	3650299.txt	0.42	4	2928181.txt	0.41	3574453.txt	0.37
5	**2031887.txt**	0.39	3574453.txt	0.41	5	**2031887.txt**	0.39	**2031887.txt**	0.36
6	3833508.txt	0.37	**1087494.txt**	0.40	6	3615012.txt	0.35	**1087494.txt**	0.35
7	3615012.txt	0.36	**1434505.txt**	0.39	7	3833508.txt	0.33	3650299.txt	0.34
8	3759464.txt	0.33	**2031887.txt**	0.38	8	3759464.txt	0.31	3488815.txt	0.32
9	3662141.txt	0.31	3514449.txt	0.34	9	3662141.txt	0.31	3514449.txt	0.32
10	2426811.txt	0.31	3615012.txt	0.34	10	3641510.txt	0.30	3615012.txt	0.32
		k = 225					k = 250		
1	**1434505.txt**	0.50	3662141.txt	0.38	1	**1434505.txt**	0.48	2758213.txt	0.37
2	3714066.txt	0.43	2758213.txt	0.37	2	3714066.txt	0.41	3662141.txt	0.33
3	**2031887.txt**	0.38	3650299.txt	0.34	3	**2031887.txt**	0.37	3650299.txt	0.33
4	2928181.txt	0.36	2928181.txt	0.34	4	2928181.txt	0.34	**1087494.txt**	0.32
5	2758213.txt	0.36	3641510.txt	0.32	5	3615012.txt	0.32	**2031887.txt**	0.32
6	3615012.txt	0.33	**2031887.txt**	0.32	6	2758213.txt	0.30	3641510.txt	0.31
7	3833508.txt	0.32	**1087494.txt**	0.32	7	3759464.txt	0.29	2928181.txt	0.30
8	3759464.txt	0.31	3615012.txt	0.30	8	3641510.txt	0.28	3724308.txt	0.29
9	3641510.txt	0.29	3574453.txt	0.29	9	3833508.txt	0.27	3615012.txt	0.28
10	2426811.txt	0.27	**1434505.txt**	0.28	10	3662141.txt	0.25	**1434505.txt**	0.28
				4000 Documents					
		k = 275					k = 300		
1	**1434505.txt**	0.42	3662141.txt	0.42	1	**2031887.txt**	0.41	3662141.txt	0.42
2	**2031887.txt**	0.41	**1087494.txt**	0.35	2	**1434505.txt**	0.40	**1434505.txt**	0.34

Table 9: Continued.

	Tf-Idf		Log Entropy			Tf-Idf		Log Entropy	
	File	Similarity	File	Similarity		File	Similarity	File	Similarity
3	3615012.txt	0.40	2758213.txt	0.32	3	3615012.txt	0.39	**1087494.txt**	0.34
4	2928181.txt	0.39	**1434505.txt**	0.32	4	3714066.txt	0.36	2758213.txt	0.33
5	3714066.txt	0.37	2928181.txt	0.32	5	3290984.txt	0.36	2928181.txt	0.31
6	3290984.txt	0.35	3574453.txt	0.31	6	2928181.txt	0.36	2990827.txt	0.31
7	2758213.txt	0.35	3488815.txt	0.30	7	3426905.txt	0.33	3135225.txt	0.30
8	3442741.txt	0.33	3135225.txt	0.30	8	3641510.txt	0.33	3650299.txt	0.29
9	3426905.txt	0.33	**2031887.txt**	0.30	9	2758213.txt	0.32	3488815.txt	0.29
10	3641510.txt	0.32	2990827.txt	0.30	10	3442741.txt	0.32	**2031887.txt**	0.29
		$k = 325$					$k = 350$		
1	**2031887.txt**	0.39	3662141.txt	0.39	1	**2031887.txt**	0.38	3662141.txt	0.37
2	**1434505.txt**	0.39	**1087494.txt**	0.34	2	**1434505.txt**	0.38	**1087494.txt**	0.34
3	3615012.txt	0.36	**1434505.txt**	0.34	3	3615012.txt	0.35	**1434505.txt**	0.33
4	3714066.txt	0.34	2928181.txt	0.31	4	3290984.txt	0.34	2990827.txt	0.31
5	3290984.txt	0.34	2990827.txt	0.29	5	3426905.txt	0.33	2928181.txt	0.30
6	3426905.txt	0.33	2758213.txt	0.29	6	3714066.txt	0.33	3135225.txt	0.28
7	2758213.txt	0.32	3514449.txt	0.29	7	2758213.txt	0.30	**2031887.txt**	0.28
8	2928181.txt	0.31	**2031887.txt**	0.28	8	3641510.txt	0.30	2758213.txt	0.27
9	3641510.txt	0.31	3833508.txt	0.28	9	2928181.txt	0.29	3650299.txt	0.27
10	2579333.txt	0.29	3615012.txt	0.28	10	2579333.txt	0.28	3615012.txt	0.27

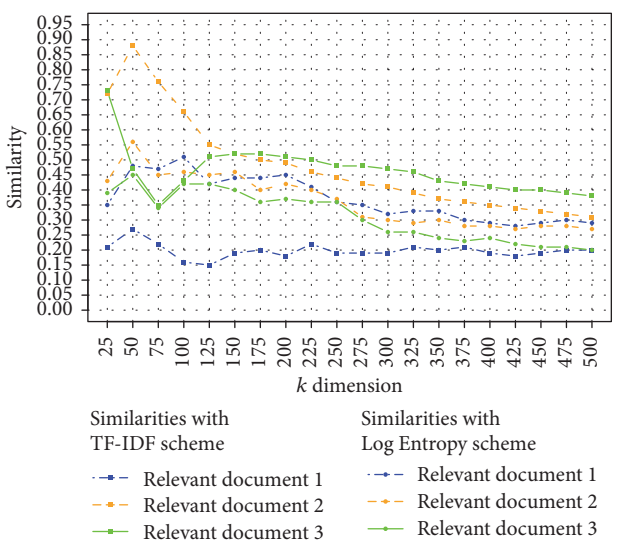


Figure 8: Results of the similarity for relevant document 1: 1087494, relevant document 2: 1434505, and relevant document 3: 2031887 in the use case of bipolar disorder using the subset of 5000 documents, twenty values of k, and two weighting schemes (TF-IDF and Log Entropy).

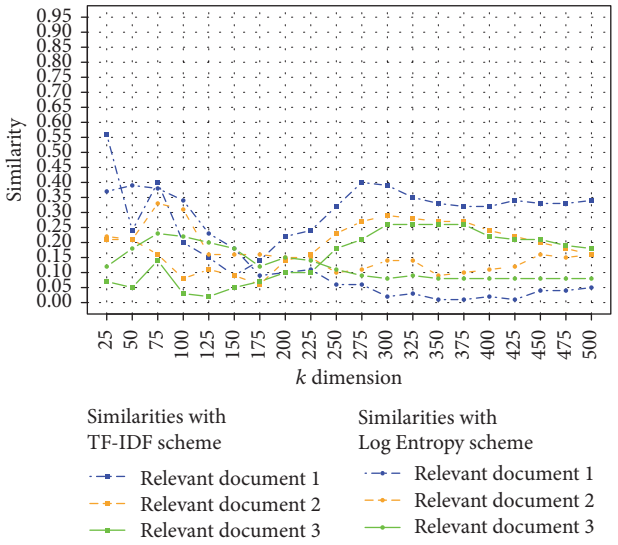


Figure 9: Results of the similarity for relevant document 1: 1065341, relevant document 2: 1459118, and relevant document 3: 1526641 in the use case of lupus disease using the subset of 5000 documents, twenty values of k, and two weighting schemes (TF-IDF and Log Entropy).

As shown in the results, in several of the experiments the identifiers for the relevant documents defined by the experts are retrieved in the top ten.

6. Conclusions

The paper has introduced a heterogeneous system based on the Latent Semantic Analysis method. The proposed system

Table 10: Results of the information retrieval process in the hLSA system with use case of lupus disease, using Log Entropy and TF-IDF weighting scheme with 1000 documents and k = 50, 75, 100, and 125; 3000 documents and k = 250, 275, 300, and 325; 5000 documents and k = 350, 375, 400, and 425.

	Tf-Idf		Log Entropy			Tf-Idf		Log Entropy	
	File	Similarity	File	Similarity		File	Similarity	File	Similarity
				1000 documents					
		k = 50					k = 75		
1	3654263.txt	0.64	3754886.txt	0.55	1	**1065341.txt**	0.85	3420478.txt	0.39
2	2974973.txt	0.63	2106780.txt	0.47	2	**1526641.txt**	0.73	2106780.txt	0.39
3	3271631.txt	0.61	2892131.txt	0.47	3	2816238.txt	0.72	2836250.txt	0.38
4	3283793.txt	0.60	2836250.txt	0.46	4	3423037.txt	0.68	3538549.txt	0.37
5	2933487.txt	0.59	3339798.txt	0.45	5	2141996.txt	0.59	3754886.txt	0.36
6	2229596.txt	0.59	3517290.txt	0.45	6	**1459118.txt**	0.55	3207056.txt	0.34
7	3501694.txt	0.58	3420478.txt	0.43	7	2138500.txt	0.48	3517290.txt	0.34
8	2816238.txt	0.58	3412204.txt	0.43	8	2832080.txt	0.47	3002958.txt	0.33
9	2684704.txt	0.58	3538549.txt	0.42	9	2133386.txt	0.44	3336207.txt	0.33
10	2108140.txt	0.58	3754383.txt	0.42	10	2974973.txt	0.44	1661595.txt	0.33
		k = 100					k = 125		
1	**1065341.txt**	0.77	2106780.txt	0.36	1	**1065341.txt**	0.72	3420478.txt	0.33
2	**1526641.txt**	0.61	3340108.txt	0.35	2	2816238.txt	0.57	3002958.txt	0.29
3	2816238.txt	0.60	3654263.txt	0.33	3	2141996.txt	0.51	3340108.txt	0.29
4	2141996.txt	0.56	3420478.txt	0.33	4	3423037.txt	0.49	3517290.txt	0.28
5	3423037.txt	0.53	3517290.txt	0.32	5	**1526641.txt**	0.48	2865479.txt	0.28
6	**1459118.txt**	0.48	3002958.txt	0.32	6	2974973.txt	0.44	2836250.txt	0.28
7	2974973.txt	0.45	2816647.txt	0.29	7	**1459118.txt**	0.43	2106780.txt	0.27
8	2138500.txt	0.43	3754886.txt	0.29	8	3514429.txt	0.40	3371498.txt	0.27
9	3514429.txt	0.40	2836250.txt	0.28	9	2138500.txt	0.36	2825732.txt	0.26
10	2119723.txt	0.40	3334796.txt	0.27	10	2119723.txt	0.33	2930170.txt	0.26
				3000 documents					
		k = 250					k = 275		
1	**1065341.txt**	0.49	3507913.txt	0.39	1	**1065341.txt**	0.50	3507913.txt	0.37
2	2247119.txt	0.47	2247119.txt	0.36	2	2247119.txt	0.46	2975002.txt	0.33
3	3842271.txt	0.38	2975002.txt	0.33	3	3842271.txt	0.37	3272557.txt	0.32
4	3260316.txt	0.38	3272557.txt	0.32	4	3260316.txt	0.36	2247119.txt	0.32
5	3512896.txt	0.37	270621.txt	0.30	5	3512896.txt	0.36	270621.txt	0.32
6	270621.txt	0.36	2724402.txt	0.30	6	270621.txt	0.34	2820672.txt	0.30
7	2141996.txt	0.34	3514429.txt	0.29	7	2141996.txt	0.34	3368184.txt	0.29
8	2873406.txt	0.33	3368184.txt	0.29	8	2974973.txt	0.31	2995072.txt	0.28
9	2974973.txt	0.33	3372565.txt	0.29	9	2133386.txt	0.30	3512896.txt	0.28
10	3526860.txt	0.32	2820672.txt	0.28	10	3526860.txt	0.30	1784585.txt	0.27
		k = 300					k = 325		
1	**1065341.txt**	0.50	3507913.txt	0.35	1	**1065341.txt**	0.48	3507913.txt	0.35
2	2247119.txt	0.46	3272557.txt	0.31	2	2247119.txt	0.46	2975002.txt	0.30
3	3842271.txt	0.34	2975002.txt	0.31	3	3842271.txt	0.33	2820672.txt	0.30
4	270621.txt	0.33	2820672.txt	0.31	4	3260316.txt	0.31	3368184.txt	0.30
5	3512896.txt	0.32	2247119.txt	0.30	5	270621.txt	0.31	2247119.txt	0.30
6	3260316.txt	0.31	2995072.txt	0.29	6	3489096.txt	0.31	3272557.txt	0.30
7	2974973.txt	0.31	270621.txt	0.29	7	2873406.txt	0.31	270621.txt	0.28
8	**1459118.txt**	0.30	3368184.txt	0.29	8	**1459118.txt**	0.29	2995072.txt	0.28
9	2873406.txt	0.30	3512896.txt	0.27	9	2133386.txt	0.29	3512896.txt	0.28
10	3526860.txt	0.30	3586474.txt	0.26	10	3512896.txt	0.29	3586474.txt	0.25
				5000 documents					
		k = 350					k = 375		
1	2247119.txt	0.47	3507913.txt	0.35	1	2247119.txt	0.47	2247119.txt	0.33
2	3842271.txt	0.36	2247119.txt	0.33	2	3514209.txt	0.34	270621.txt	0.31

Table 10: Continued.

	Tf-Idf		Log Entropy			Tf-Idf		Log Entropy	
	File	Similarity	File	Similarity		File	Similarity	File	Similarity
3	3160874.txt	0.35	3272557.txt	0.31	3	2801992.txt	0.33	3603166.txt	0.31
4	270621.txt	0.34	3603166.txt	0.31	4	3842271.txt	0.33	3586474.txt	0.30
5	**1065341.txt**	0.33	270621.txt	0.30	5	**1065341.txt**	0.32	3272557.txt	0.30
6	3514209.txt	0.33	2975002.txt	0.30	6	3160874.txt	0.32	2975002.txt	0.30
7	2873406.txt	0.32	2820672.txt	0.29	7	2873406.txt	0.32	2820672.txt	0.29
8	3510383.txt	0.30	3284243.txt	0.28	8	3284243.txt	0.31	3507913.txt	0.29
9	3284243.txt	0.30	3435747.txt	0.27	9	270621.txt	0.31	3284243.txt	0.29
10	2801992.txt	0.29	3586474.txt	0.27	10	3489096.txt	0.30	3435747.txt	0.28
		$k = 400$					$k = 425$		
1	2247119.txt	0.47	2247119.txt	0.33	1	2247119.txt	0.47	3272557.txt	0.31
2	3842271.txt	0.33	270621.txt	0.31	2	**1065341.txt**	0.34	270621.txt	0.31
3	3514209.txt	0.32	3272557.txt	0.30	3	3514209.txt	0.33	2247119.txt	0.30
4	2873406.txt	0.32	3507913.txt	0.29	4	3842271.txt	0.32	3507913.txt	0.29
5	2801992.txt	0.32	2975002.txt	0.28	5	2873406.txt	0.32	2975002.txt	0.28
6	**1065341.txt**	0.32	3435747.txt	0.28	6	2801992.txt	0.30	3435747.txt	0.28
7	3160874.txt	0.31	3586474.txt	0.28	7	3160874.txt	0.30	3603166.txt	0.27
8	270621.txt	0.30	2820672.txt	0.28	8	3501694.txt	0.30	2820672.txt	0.27
9	3489096.txt	0.30	3284243.txt	0.28	9	270621.txt	0.29	3368184.txt	0.26
10	3284243.txt	0.29	3603166.txt	0.27	10	2778347.txt	0.29	3586474.txt	0.26

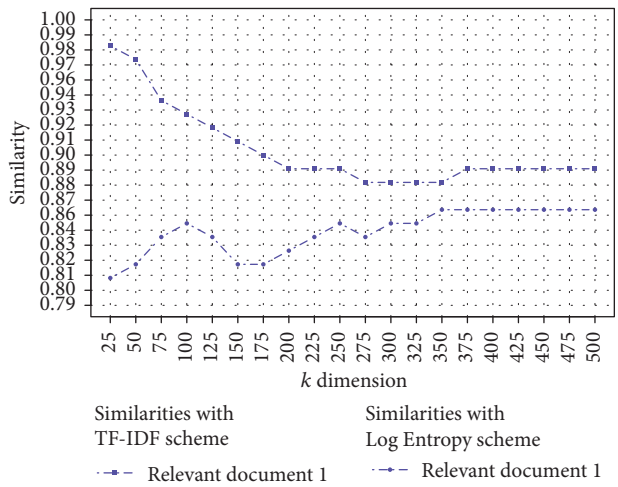


Figure 10: Results of the similarity for relevant document 1: 1087494 in the use case of topiramate weight-loss using the subset of 5000 documents, twenty values of k, and two weighting schemes (TF-IDF and Log Entropy).

allows the indexing of text documents from a knowledge base faster than the standard LSA versions. In addition, the hLSA system allows the retrieval of relevant documents based on a query. The results we have found are feasible and promising.

The hLSA system is divided into three main stages: semantic space, dimensionality reduction, and information retrieval. Acceleration of four times is shown in the semantic space stage, acceleration of eight times is shown in a dimensionality reduction stage, and in the information retrieval stage we do not find acceleration. However, the hLSA system is limited by the main and global memory of GPU and CPU. Therefore, in future work, we propose adding a multi-GPU functionality to increase the size of matrices to be processed.

We have succeeded with our main motivation of improving the execution time of Latent Semantic Analysis method through the use of heterogeneous architectures such as GPUs and multi-CPU. As shown in experimental tests, we have achieved overall acceleration of around eight times faster than standard LSA systems for five thousand documents.

We have retrieved documents using twelve subsets of different sizes from five hundred to five thousand documents of the Open Access Subset of PubMed Central by using two weighting schemes: the Log Entropy and the Term Frequency-Inverse Document Frequency and twenty different values of k ranging from 25 to 250 to prove the appropriate value with which the hLSA system is more accurate. Hence, the hLSA has achieved an accuracy of 88% with the use case of bipolar disorders, an accuracy of 56% with the use case of lupus disease, and an accuracy of 98% with the use case of topiramate weight-loss. Moreover, we can infer that in our experiments the Log Entropy scheme has a higher value of similarity in two out of three use cases over the TF-IDF scheme.

Table 11: Results of the information retrieval process in the hLSA system with use case of topiramate weight-loss, using Log Entropy and TF-IDF weighting scheme with 1500 documents and $k = 25$, 50, 75, and 100; 3500 documents and $k = 150$, 175, 200, and 225; 4000 documents and $k = 425$, 450, 475, and 500.

	Tf-Idf		Log Entropy			Tf-Idf		Log Entropy	
	File	Similarity	File	Similarity		File	Similarity	File	Similarity
				1500 documents					
		$k = 25$					$k = 50$		
1	**1087494.txt**	0.97	**1087494.txt**	0.92	1	**1087494.txt**	0.96	**1087494.txt**	0.89
2	2838810.txt	0.94	3552983.txt	0.82	2	2362538.txt	0.86	3128260.txt	0.74
3	3720211.txt	0.94	3055095.txt	0.82	3	2063236.txt	0.85	3771300.txt	0.73
4	2246781.txt	0.94	3880815.txt	0.82	4	3845698.txt	0.85	3055095.txt	0.71
5	3337849.txt	0.93	3771300.txt	0.82	5	2719193.txt	0.85	3148993.txt	0.69
6	2918360.txt	0.93	3284015.txt	0.79	6	2246781.txt	0.84	2275478.txt	0.68
7	3519703.txt	0.93	3128260.txt	0.78	7	3800929.txt	0.84	2941806.txt	0.64
8	2719193.txt	0.93	3800929.txt	0.75	8	2360336.txt	0.83	3284015.txt	0.64
9	3191588.txt	0.93	1896169.txt	0.74	9	545962.txt	0.82	3540966.txt	0.62
10	3845698.txt	0.93	3617973.txt	0.74	10	3095993.txt	0.82	3005214.txt	0.62
		$k = 75$					$k = 100$		
1	**1087494.txt**	0.94	**1087494.txt**	0.90	1	**1087494.txt**	0.93	**1087494.txt**	0.90
2	2362538.txt	0.83	3128260.txt	0.65	2	2246781.txt	0.77	3128260.txt	0.61
3	2063236.txt	0.81	3055095.txt	0.59	3	2063236.txt	0.77	3055095.txt	0.60
4	2719193.txt	0.80	3148993.txt	0.56	4	3128260.txt	0.75	3650299.txt	0.54
5	2246781.txt	0.80	3098785.txt	0.56	5	2719193.txt	0.75	3771300.txt	0.52
6	3128260.txt	0.80	3650299.txt	0.55	6	2362538.txt	0.75	3443410.txt	0.49
7	3800929.txt	0.78	3617973.txt	0.55	7	2275478.txt	0.74	2275478.txt	0.48
8	545962.txt	0.78	3540966.txt	0.54	8	2360336.txt	0.73	3540966.txt	0.48
9	3720211.txt	0.77	2275478.txt	0.54	9	3720211.txt	0.72	1434505.txt	0.47
10	3112026.txt	0.77	3771300.txt	0.54	10	3800929.txt	0.72	3337849.txt	0.47
				3500 documents					
		$k = 150$					$k = 175$		
1	**1087494.txt**	0.90	**1087494.txt**	0.84	1	**1087494.txt**	0.90	**1087494.txt**	0.84
2	2001661.txt	0.72	3128260.txt	0.48	2	2001661.txt	0.69	3443410.txt	0.47
3	3857881.txt	0.70	3443410.txt	0.48	3	3857881.txt	0.65	3128260.txt	0.45
4	3855068.txt	0.68	3650299.txt	0.45	4	2275478.txt	0.64	3650299.txt	0.41
5	2275478.txt	0.67	3055095.txt	0.42	5	3855068.txt	0.64	3420720.txt	0.39
6	3720211.txt	0.67	2988027.txt	0.39	6	3720211.txt	0.64	3771300.txt	0.38
7	2246781.txt	0.66	1434505.txt	0.39	7	3065878.txt	0.64	3055095.txt	0.38
8	3065878.txt	0.66	3420720.txt	0.39	8	3196922.txt	0.63	3286846.txt	0.38
9	2063236.txt	0.65	3771300.txt	0.39	9	2246781.txt	0.61	3022769.txt	0.38
10	2887190.txt	0.65	3751340.txt	0.37	10	2063236.txt	0.61	1434505.txt	0.38
		$k = 200$					$k = 225$		
1	**1087494.txt**	0.89	**1087494.txt**	0.85	1	**1087494.txt**	0.89	**1087494.txt**	0.86
2	2001661.txt	0.64	3443410.txt	0.48	2	2001661.txt	0.59	3443410.txt	0.48
3	3065878.txt	0.60	3650299.txt	0.41	3	3128260.txt	0.56	1434505.txt	0.41
4	2275478.txt	0.60	3128260.txt	0.40	4	3720211.txt	0.56	3650299.txt	0.40
5	3128260.txt	0.59	2988027.txt	0.39	5	2275478.txt	0.55	3286846.txt	0.39
6	3855068.txt	0.58	3286846.txt	0.39	6	3855068.txt	0.55	3128260.txt	0.39
7	3720211.txt	0.57	3540966.txt	0.37	7	3065878.txt	0.54	3540966.txt	0.38
8	3196922.txt	0.57	1434505.txt	0.36	8	2887190.txt	0.53	2988027.txt	0.36
9	2887190.txt	0.57	3420720.txt	0.36	9	2063236.txt	0.53	3592507.txt	0.34
10	2063236.txt	0.57	3065878.txt	0.35	10	3448379.txt	0.51	3065878.txt	0.33
				4000 documents					
		$k = 425$					$k = 450$		
1	**1087494.txt**	0.89	**1087494.txt**	0.86	1	**1087494.txt**	0.89	**1087494.txt**	0.86
2	2350122.txt	0.47	3443410.txt	0.36	2	2350122.txt	0.44	3650299.txt	0.36

Table 11: Continued.

	Tf-Idf		Log Entropy			Tf-Idf		Log Entropy	
	File	Similarity	File	Similarity		File	Similarity	File	Similarity
3	2362551.txt	0.44	1434505.txt	0.36	3	2362551.txt	0.42	3443410.txt	0.35
4	2001661.txt	0.42	3650299.txt	0.35	4	3065878.txt	0.40	1434505.txt	0.35
5	3065878.txt	0.41	3286846.txt	0.31	5	3530724.txt	0.40	3128260.txt	0.31
6	1832214.txt	0.40	3128260.txt	0.30	6	3650299.txt	0.40	2350122.txt	0.30
7	3530724.txt	0.39	2350122.txt	0.29	7	1832214.txt	0.39	3286846.txt	0.30
8	3592507.txt	0.39	3540966.txt	0.27	8	2001661.txt	0.39	2750184.txt	0.25
9	3857881.txt	0.39	2750184.txt	0.26	9	3592507.txt	0.39	3540966.txt	0.24
10	1831765.txt	0.39	293436.txt	0.24	10	1831765.txt	0.37	293436.txt	0.23
		$k = 475$					$k = 500$		
1	**1087494.txt**	0.89	**1087494.txt**	0.87	1	**1087494.txt**	0.89	**1087494.txt**	0.87
2	2350122.txt	0.44	3650299.txt	0.37	2	2350122.txt	0.45	3650299.txt	0.37
3	3650299.txt	0.40	3443410.txt	0.34	3	3650299.txt	0.42	3443410.txt	0.35
4	2362551.txt	0.40	1434505.txt	0.33	4	3530724.txt	0.40	1434505.txt	0.34
5	3530724.txt	0.39	3128260.txt	0.31	5	2362551.txt	0.38	3128260.txt	0.30
6	3065878.txt	0.38	3286846.txt	0.30	6	3065878.txt	0.37	3286846.txt	0.29
7	1832214.txt	0.38	2350122.txt	0.30	7	2001661.txt	0.36	2350122.txt	0.29
8	2001661.txt	0.37	2750184.txt	0.25	8	3592507.txt	0.36	2750184.txt	0.24
9	3592507.txt	0.36	293436.txt	0.23	9	1832214.txt	0.36	3181688.txt	0.23
10	1831765.txt	0.36	2988027.txt	0.23	10	3286846.txt	0.35	293436.txt	0.23

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Universidad Politécnica Salesiana (UPS) through the research group of Cloud Computing, Smart Cities & High Performance Computing (GIHP4C) and the Sciences Research Council (CONACyT) through the Research Project no. 262756.

References

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the Association for Information Science and Technology*, vol. 41, no. 6, pp. 391–407, 1990.

[2] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge," *Psychological Review*, vol. 104, no. 2, pp. 211–240, 1997.

[3] T. J. Connolly, V. D. Veksler, and W. D. Gray, "Predicting Interest: Another Use for Latent Semantic Analysis," in *Proceedings of the Eighth International Conference on Cognitive Modeling*, 2009.

[4] B. Li and K. Wang, "Computer aided diagnosis semantic model for the report of medical image via LDA and LSA," in *Proceedings of the 2011 IEEE International Symposium on IT in Medicine and Education, ITME 2011*, pp. 699–703, chn, December 2011.

[5] A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards, "Detecting cyberbullying: Query terms and techniques," in *Proceedings of the 3rd Annual ACM Web Science Conference, WebSci 2013*, pp. 195–204, fra, May 2013.

[6] P. Zhang, Y. Zhang, T. Thomas, and S. Emmanuel, "Moving people tracking with detection by latent semantic analysis for visual surveillance applications," *Multimedia Tools and Applications*, vol. 68, no. 3, pp. 991–1021, 2014.

[7] J. Wang, J. Peng, and O. Liu, "A classification approach for less popular webpages based on latent semantic analysis and rough set model," *Expert Systems with Applications*, vol. 42, no. 1, pp. 642–648, 2015.

[8] W. Zhang, T. Yoshida, and X. Tang, "TFIDF, LSI and multi-word in information retrieval and text categorization," in *Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics, SMC 2008*, pp. 108–113, sgp, October 2008.

[9] S. T. Dumais, "Latent Semantic Analysis," *Annual Review of Information Science and Technology*, vol. 38, pp. 189–230, 2004.

[10] J. R. Herrera-Morales and L. I. Barbosa-Santillán, "Analysis of Medical Publications with Latent Semantic Analysis Method," in *Proceedings of the The Third International Conference on Advances in Information Mining and Managament*, pp. 1–6, 2013.

[11] B. Rosario, "Latent semantic indexing: An overview," *Techn rep INFOSYS*, vol. 240, pp. 1–16, 2000.

[12] J. M. Cavanagh, T. E. Potok, and X. Cui, "Parallel latent semantic analysis using a graphics processing unit," in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pp. 2505–2510, ACM Press, NY, USA, 2009.

[13] S. Vigna, "Distributed, Large-Scale Latent Semantic Analysis by Index Interpolation," in *Proceedings of the 3rd International ICST Conference on Scalable Information Systems*, Vico Equense, Italy, June 2008.

[14] G. A. León-Paredes and L. I. Barbosa-Santillán, "A Heterogeneous System Based on Latent Semantic Analysis (hLSA)," in *Proceedings of GPU Technology Conference*, 2017, http://on-demand.gputechconf.com/gtc/2017/posters/images/1920x1607/GTC_2017_Algorithms_AL_11_P7175_WEB.png.

[15] B. Pincombe, "Comparison of human and latent semantic analysis (lsa) judgements of pairwise document similarities for a news corpus," Tech. rep., Defence Science and Technology Organisation Salisbury (Australia) Info Sciences Lab, 2004.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, Md, USA, 4th edition, 2013.

[17] S. C. Deerwester, S. T. Dumais, G. W. Furnas et al., "Computer information retrieval using latent semantic structure," *US Patent 4,839,853*, 1989.

Hindawi
Scientific Programming
Volume 2017, Article ID 1329281, 6 pages
https://doi.org/10.1155/2017/1329281

Research Article

Sentiment Analysis in Spanish for Improvement of Products and Services: A Deep Learning Approach

Mario Andrés Paredes-Valverde,[1] Ricardo Colomo-Palacios,[2] María del Pilar Salas-Zárate,[1] and Rafael Valencia-García[1]

[1]*Departamento de Informática y Sistemas, Universidad de Murcia, 30100 Murcia, Spain*
[2]*Computer Science Department, Østfold University College, Holden, Norway*

Correspondence should be addressed to María del Pilar Salas-Zárate; mariapilar.salas@um.es

Received 16 June 2017; Accepted 27 August 2017; Published 26 October 2017

Academic Editor: Jezreel Mejia-Miranda

Sentiment analysis is an important area that allows knowing public opinion of the users about several aspects. This information helps organizations to know customer satisfaction. Social networks such as Twitter are important information channels because information in real time can be obtained and processed from them. In this sense, we propose a deep-learning-based approach that allows companies and organizations to detect opportunities for improving the quality of their products or services through sentiment analysis. This approach is based on convolutional neural network (CNN) and word2vec. To determine the effectiveness of this approach for classifying tweets, we conducted experiments with different sizes of a Twitter corpus composed of 100000 tweets. We obtained encouraging results with a precision of 88.7%, a recall of 88.7%, and an F-measure of 88.7% considering the complete dataset.

1. Introduction

Nowadays, there is a lot of online opinions. This information is important for users because it helps them to make decisions about buying a product, voting in a political election, and choosing a travel destination, among other subjects. This information is also important for organizations since it helps them to know the general opinion about their products, the sales forecast, and the customer satisfaction in real time. Based on this information, companies can identify opportunities for improving the quality of their products or services.

A good example that demonstrates the importance of the opinions is a t-shirt of Zara clothing store which received negative opinions because it looked like the clothes used in the Holocaust. In these situations, companies must act quickly and solve the problem to avoid these opinions affecting their reputation. In this sense, to know the public opinion in real time is very important. Twitter is a social network, where users share information on almost everything in real time. Therefore, companies consider this social network as a rich source of information that allows knowing the general opinion about their products and services, among others [1]. However, analyzing and processing all these opinions require much time and effort for the humans. On these grounds, a technology that processes automatically this information has arisen. This technology is known as sentiment analysis or opinion mining.

Sentiment analysis has been defined by several authors. However the definition most used in the research community is the proposed by Liu [2], who defined it as follows: "Sentiment analysis is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes."

In the last years, several approaches have been proposed for sentiment analysis. Most of these approaches are based on two main techniques, semantic orientation and machine learning. Although good results were obtained for both

techniques, several works in the literature have demonstrated that machine learning obtained better results. However, in more recent years a new technique known as deep learning has captured the attention of researchers because it has significantly outperformed traditional methods [3, 4]. Most of the deep-learning-based approaches for sentiment analysis are based on the English language. Hence, we propose a deep-learning-based approach for sentiment analysis of tweets in Spanish. Spanish is the third language most used on the Internet (http://www.internetworldstats.com/stats7.htm). Therefore, we consider that new approaches for sentiment analysis in the Spanish language are necessary.

The remainder of the paper is structured as follows. Section 2 presents a review of the literature about sentiment analysis and deep learning. Section 3 described the proposed approach. The experiments and results are presented in Section 4. Finally, Section 5 presents conclusions and future work.

2. Related Works

In the literature, several authors have proposed approaches for the sentiment analysis. These works have used two main techniques, semantic orientation and machine learning. With respect to the first technique, approaches use sentiment lexicons to determine the polarity. SentiWordNet is the most used lexicon in the literature [5, 6]. This lexicon is based on WordNet and it contains multiple senses of a word. Also, it provides a positive, objective, and negative value for each sense. Several works using this technique have obtained promising results; however, some other works have not obtained good results due to two main reasons: (1) sentiment lexicons mainly are based on English, which forces researchers to translate the English lexicons to the target language and (2) a word can have different senses depending on the domain where they are used.

Regarding the machine learning approach, authors use classification algorithms such as Support Vector Machines (SVM) [7–11], Bayesian Networks (BayesNet) [12], and decision trees (J48) [10], among others. For this technique, two data sets are necessary, a training set and an evaluation set. The training set is used for the algorithm to learn from features of the domain. Meanwhile, the evaluation set is used to validate the built model from the training set. The performance of the machine learning technique depends on the effectiveness of the selected method for feature extraction. Among the most used methods are bag of words [13], TF-IDF [14], *n*-grams (unigrams, bigrams, and trigrams) [11, 15], features based on POS tagging [16], and features based on dependency rules [17].

However, most recent works are based on deep learning techniques. For instance, Dos Santos and Gatti [18] proposed an approach to sentiment analysis of short texts. The approach is based on convolutional neural network, which is applied on two corpora, movies reviews (Stanford Sentiment Tree-bank) and Twitter messages (Stanford Twitter Sentiment corpus). Araque et al. [19] introduced an approach based on deep learning for sentiment classification. The authors used a word embeddings model and a machine learning algorithm. To evaluate the performance of the proposed approach, the authors used six corpora publicly available of Twitter and movies reviews. Hu et al. [20] proposed a framework based on neural network for sentiment analysis. This framework is composed of two main phases. Firstly, feature vectors are obtained through linguistic and domain knowledge. Secondly, a Deep Neural Network is designed. Also, the authors evaluated their approach on three datasets (electronic products, movies reviews, and hotels reviews). Tang et al. [21] built a supervised learning framework. The authors combined sentiment features and features related to emoticons, negation, punctuation, cluster, and *n*-grams. Then, they trained a classifier by using a benchmark corpus provided in SemEval 2013. Ruder et al. [22] proposed an approach to aspect-based sentiment analysis. The authors used a convolutional neural network (CNN) for aspect extraction and sentiment analysis. The proposal was evaluated in several domains such as restaurants, hotels, laptops, phones, and cameras. Severyn and Moschitti [4] introduced a deep learning model which is applied to two tasks of SemEval 2015, namely, message-level and phrase-level of Twitter sentiment analysis. Sun et al. [23] proposed a sentiment analysis approach for Chinese microblog with a Deep Neural Network model. The proposed method extracted features to obtain semantic and information of words. Finally, three models, SVM, Naïve Bayes, and Deep Neural Network, are selected to prove the effectiveness of the method. Finally, Poria et al. [24] presented an approach to aspect extraction for sentiment analysis by using a deep learning technique. Also, the authors obtained a set of linguistic patterns to combine them with neural networks.

On the other hand, the approaches for the sentiment analysis are mainly focused on the analysis of the opinions of blogs, forums, and travel and sales websites. However, recently a more special interest has arisen on social networks such as Twitter because a lot of information from different topics can be extracted for its analysis. Among the most studied domains in the sentiment analysis area are movies, technological products, tourism, and health. Finally, regarding the language, most of them are based on English language and only one is based on the Chinese language.

Next section describes the deep-learning-based approach for sentiment analysis proposed in this work. More specifically, this section describes the architecture of our proposal as well as the relations among all its components.

3. Approach

The sentiment classification approach presented in this work is divided into three main modules: (1) preprocessing module, (2) word embeddings, and (3) CNN model. Figure 1 shows the workflow of the system. Firstly, the tokenization and normalization of the text are carried out. Secondly, word2vec is used to obtain the feature vectors. The last step consists in training a convolutional neural network to classify tweets as positives or negatives. A detailed description of these modules is provided in the following sections.

Parece q tenías razón *@bufalo58* y tendré q cambiarme a iPhone, xq el servicio técnico de *@SamsumgChile* no va a reparar mi celu **#ChaoSamsumg**	*Mentions and replies* **Hashtags** URL

Box 1

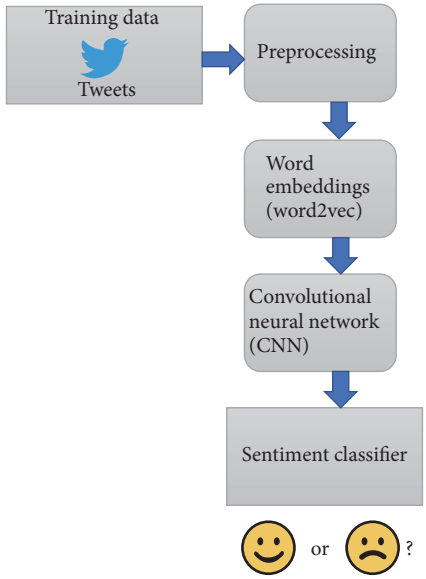


Figure 1: Our proposed approach.

Figure 2: Example of a tweet.

3.1. Preprocessing Module. The first step of the method proposed consists in the preprocessing of the tweets. Twitter is a social network where users use informal language due to the limitation of 140 characters. Therefore, there are several issues such as spelling errors, slang words, abbreviations, and replication of characters, among others, that must be addressed before detecting the polarity of a tweet. Figure 2 presents a tweet with some of these issues. In order to deal with this problem, we adopted the approach presented in [25] for the tweets processing.

The first phase of the preprocessing module consists in the tokenization process. In this process, the text is divided in tokens, which can be words or punctuation marks. To perform this process, the Twokenize (http://www.cs.cmu.edu/~ark/TweetNLP/) tool was used. This tool is oriented to Twitter and allows identifying items of Twitter like hashtags, mentions and replies, and URLs, among others.

The second phase of this module consists in the normalization of the text. Firstly, items identified by Twokenize are removed because they do not provide important information for the detection of polarity. Next, each item removed from tweets is described.

(1) Mentions and replies to users: these items are represented with @.

(2) URLs: all items start with http://

(3) Hashtags: in this case, the character # is only removed due to the rest of the text representing an important part to be analyzed.

For instance, let us consider the tweet presented in Figure 2: "Parece q tenías razón @bufalo58 y tendré q cambiarme a iPhone, xq el servicio técnico de @SamsungChile no va a reparar mi celu #ChaoSamsung—It looks like you were right @bufalo58 and I will have to switch to iPhone because the @SamsungChile technical service is not going to repair my cell phone #ChaoSamsung." In this step, Twokenize detects two mentions and one Hashtag. Then, the module removes the mentions ("@bufalo58" and "@SamsumgChile") and the character "#" of "#ChaoSamsumg" (see Box 1).

Secondly, hashtags (strings that contain one or more words) are split based on capital letters. Considering the example presented above, #ChaoSamsung is split into two words "Chao" and "Samsung."

Thirdly, abbreviations and shorthand notations are extended. To this aim, we used the NetLingo (http://www.netlingo.com) dictionary. For example, "que" instead of "q," "por que" instead of "xq," and "celular" instead of "celu." Finally, Hunspell (http://hunspell.github.io) dictionary is used to correct spelling errors.

3.2. Word Embeddings. In this approach, we use word2vec for learning word embeddings. This tool implements the continuous bag-of-words model (CBOW) and skip-gram model for computing vector representations of words [26]. Word embeddings represent an important part in CNN architecture due to the fact that it allows obtaining syntactic and semantic information from the tweets, which is very important for sentiment classification.

3.3. CNN Model. We use a deep convolutional neural network for classification of tweets into positive and negative classes. The CNN (convolutional neural network) architecture requires concatenated word vectors of the text as input. Regarding the implementation of this model, Tensorflow (https://www.tensorflow.org) was used.

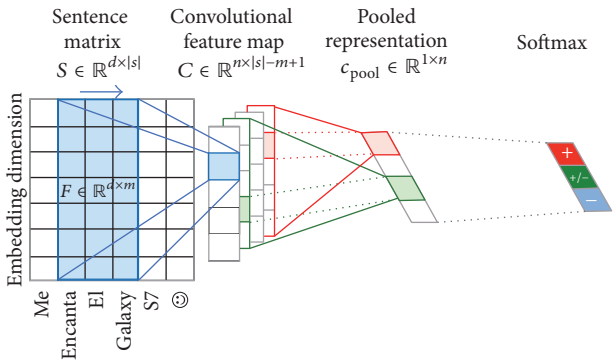


Figure 3: Learning model for sentiment classification [4].

Next, Figure 3 shows the architecture of a convolutional neural network used for sentiment classification [4].

4. Experiments

4.1. Data. The main objective of this approach is to detect important information about products and services that allows companies and organizations to improve them. Therefore, our approach requires a corpus related to products and services. Although several corpora have been provided in the literature, there is a lack of corpora for Spanish. In this sense, we have obtained a corpus from Twitter in Spanish. The process for collecting this corpus is described below.

(1) Tweets were collected by using Twitter4J (http://twitter4j.org/) library. To obtain relevant tweets, a set of keywords related to technological products were defined.

(2) Duplicated tweets, retweets, tweets in other languages, and tweets that contain only URLs were removed.

(3) We obtained a total of 70000 positive tweets and 63000 negative tweets.

(4) Finally, we selected only 50000 positive tweets and 50000 negative tweets, which were manually analyzed to obtain those relevant to our study.

This corpus is not available publicly because according to the Twitter privacy policy it is not possible to share the content of the tweets. Next, two examples from the corpus collected are presented. Figure 4 shows a positive tweet "Una excelente característica del iPhone 7 #JumboMobile @tiendasjumboco es su resistencia al agua—An excellent feature of the iPhone 7 #JumboMobile @tiendasjumboco is its water resistance," while Figure 5 shows an example of a negative tweet "lo que quise dar a entender es que n me salio bueno ni el cargador ni el iPhone pq se me rompieron los dos—What I meant to say was, both the charger and iPhone were not good because the two broke."

Table 1 shows the distribution of our corpus. As can be seen, 40000 positive and negative tweets were used to train the classifier and 10000 tweets positive and negative were used to test the model built.

Figure 4: Positive tweet.

Figure 5: Negative tweet.

Table 1: Distribution of the corpus.

	Positive	Negative	Total
Train	40000	40000	80000
Test	10000	10000	20000

4.2. Evaluation and Results. Aiming to measure the performance of our proposed approach, we have used well-known metrics: precision, recall, and F-measure. Precision (1) represents the proportion of predicted positive cases that are real positives. On the other hand, recall (2) is the proportion of actual positive cases that were correctly predicted as such. F-measure (3) is the harmonic mean of precision and recall [27].

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}, \quad (1)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}, \quad (2)$$

$$F\text{-measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

Also, we used the macro precision (4), macro recall (5), and macro F-measure (6) metrics due to the fact that the polarity detection is a multiclass problem.

$$\text{Macro precision} = \frac{\sum_{i=1}^{|C|} \text{Precision}}{|C|}, \quad (4)$$

$$\text{Macro Recall} = \frac{\sum_{i=1}^{|C|} \text{Recall}}{|C|}, \quad (5)$$

$$\text{Macro } F\text{-measure} = \frac{\sum_{i=1}^{|C|} F\text{-measure}}{|C|}. \quad (6)$$

Table 2 shows that our approach obtained encouraging results with a precision of 88.5%, a recall of 88.8%, and an F-measure

Table 2: Results obtained with CNN.

	Precision	Recall	F-measure
Positive	0.885	0.888	0.887
Negative	0.888	0.884	0.886
Macro	**0.887**	**0.887**	**0.887**

Table 3: Results of traditional model and Deep model.

	CNN	SVM	NB
5000	0.785	0.675	0.680
10000	0.793	0.723	0.712
20000	0.843	0.805	0.802
40000	0.844	0.808	0.779
60000	0.852	0.811	0.771
80000	0.858	0.833	0.777
100000	0.887	0.837	0.779

of 88.7% for the positive class, and a precision of 88.8%, a recall of 88.4%, and an F-measure of 88.6% for the negative class.

4.3. Comparison with Traditional Learning Methods. In this work, different classification algorithms were compared with the same feature vector, namely, SVM, NB, and CNN (see Table 3). For a fair comparison, the default parameters were used for each algorithm without carrying out an additional tuning process. This analysis was carried out in order to study the effects of the proposed approach with a convolutional neural network. The algorithms were evaluated with several sizes of the corpus. Each subset is split into two datasets: (1) 80% of the data is used as a training set and (2) 20% of the data is used as a testing set.

As can be seen in Figure 6, traditional models show similar results. However, SVM provides better results than NB when the size of data increases. On the other hand, results also indicate that convolutional neural network obtained better results that traditional models (SVM and NB) with the different subsets of the Twitter corpus. These results confirm that deep learning techniques outperformed traditional methods of machine learning for sentiment analysis.

It is important to mention that we did not carry out a comparison of our results with those reported in related works because there is a lack of deep learning approaches for sentiment analysis in Spanish.

5. Conclusions and Future Work

In this work, we presented an approach for Twitter sentiment analysis. The main objective of this proposal was providing the basis to know customer satisfaction and identify opportunities for improvement of products and services. The proposal is based on a deep learning model to build a classifier for sentiment detection. Our approach obtained encouraging results, with a precision, recall, and F-measure of 88.7%. The results also show that CNN outperformed traditional models such as SVM and NB.

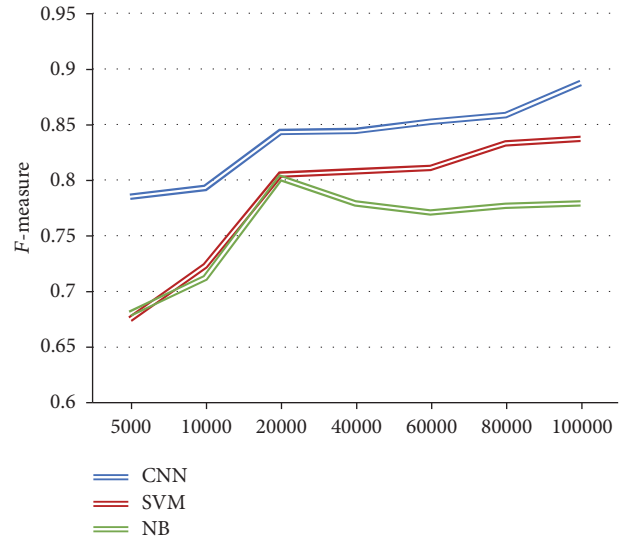


Figure 6: Comparison of SVM, NB, and CNN for different sizes of subsets for the Twitter dataset.

As future work, we are considering exploring other neural network models such as Recursive Neural Tensor Networks (RNTN), Recurrent Neural Networks (RNN), and Long Short Term Memory (LSTM). Also, we plan to evaluate other word embedding features as those presented in [21]. Finally, we have considered applying our approach to other languages such as English, French, and Arabic.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Commission (FEDER/ERDF) through Project KBS4FIA (TIN2016-76323-R). María del Pilar Salas-Zárate and Mario Andrés Paredes-Valverde are supported by the National Council of Science and Technology (CONACYT), the Secretariat of Public Education (SEP), and the Mexican government.

References

[1] A. Das, S. Gollapudi, and K. Munagala, "Modeling opinion dynamics in social networks," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14)*, pp. 403–412, 2014.

[2] B. Liu, "Sentiment Analysis and Opinion Mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[3] M. Cliche, "BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs," Apr. 2017.

[4] A. Severyn and A. Moschitti, "UNITN: training deep convolutional neural network for twitter sentiment classification,"

in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval '15)*, pp. 464–469, 2015.

[5] I. Peñalver-Martinez, F. Garcia-Sanchez, R. Valencia-Garcia et al., "Feature-based opinion mining through ontologies," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5995–6008, 2014.

[6] A. Montejo-Ráez, E. Martínez-Cámara, M. T. Martín-Valdivia, and L. A. Ureña-López, "Ranked word net graph for sentiment polarity classification in Twitter," *Computer Speech & Language*, vol. 28, no. 1, pp. 93–107, 2014.

[7] P. Biyani, C. Caragea, P. Mitra et al., "Co-training over Domain-independent and Domain-dependent features for sentiment analysis of an online cancer support community," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13)*, pp. 413–417, ACM, Sydney, Australia, August 2013.

[8] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: an empirical comparison between SVM and ANN," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621–633, 2013.

[9] N. U. Pannala, C. P. Nawarathna, J. T. K. Jayakody, L. Rupasinghe, and K. Krishnadeva, "Supervised learning based approach to aspect based sentiment analysis," in *Proceedings of the 16th IEEE International Conference on Computer and Information Technology (CIT '16)*, pp. 662–666, 2016.

[10] M. Venugopalan and D. Gupta, "Exploring sentiment analysis on twitter data," in *Proceedings of the 8th International Conference on Contemporary Computing (IC3 '15)*, pp. 241–247, August 2015.

[11] M. del Pilar Salas-Zárate, M. A. Paredes-Valverde, J. Limon-Romero, D. Tlapa, and Y. Baez-Lopez, "Sentiment classification of Spanish reviews: an approach based on feature selection and machine learning methods," *Journal of Universal Computer Science*, vol. 22, no. 5, pp. 691–708, 2016.

[12] M. Del Pilar Salas-Zárate, E. López-López, R. Valencia-García, N. Aussenac-Gilles, Á. Almela, and G. Alor-Hernández, "A study on LIWC categories for opinion mining in Spanish reviews," *Journal of Information Science*, vol. 40, no. 6, pp. 749–760, 2014.

[13] N. F. F. Da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170–179, 2014.

[14] E. Fersini, E. Messina, and F. A. Pozzi, "Sentiment analysis: bayesian ensemble learning," *Decision Support Systems*, vol. 68, pp. 26–38, 2014.

[15] P. Smith and M. Lee, "Cross-discourse development of supervised sentiment analysis in the clinical domain," in *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pp. 79–83, 2012.

[16] I. Habernal, T. Ptáček, and J. Steinberger, "Reprint of "supervised sentiment analysis in Czech social media"," *Information Processing and Management*, vol. 51, no. 4, pp. 532–546, 2015.

[17] B. Agarwal, S. Poria, N. Mittal, A. Gelbukh, and A. Hussain, "Concept level sentiment analysis using dependency-based semantic parsing: a novel approach," *Cognitive Computation*, vol. 7, no. 4, pp. 487–499, 2015.

[18] C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of the 25th International Conference on Computational Linguistics (COLING '14)*, pp. 69–78, 2014.

[19] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017.

[20] Z. Hu, J. Hu, W. Ding, and X. Zheng, "Review Sentiment Analysis Based on Deep Learning," in *Proceedings of the 12th IEEE International Conference on E-Business Engineering (ICEBE '15)*, pp. 87–94, 2015.

[21] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Coooolll: A Deep Learning System for Twitter Sentiment Classification," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval '14)*, pp. 208–212, Dublin, Ireland, 2014.

[22] S. Ruder, P. Ghaffari, and J. G. Breslin, "INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval '16)*, pp. 330–336, San Diego, CA, USA, June 2016.

[23] X. Sun, C. Li, and F. Ren, "Sentiment analysis for Chinese microblog based on deep neural networks with convolutional extension features," *Neurocomputing*, vol. 210, pp. 227–236, 2016.

[24] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.

[25] M. d. Salas-Zárate, M. A. Paredes-Valverde, M. Á. Rodriguez-García, R. Valencia-García, and G. Alor-Hernández, "Automatic detection of satire in Twitter: A psycholinguistic-based approach," *Knowledge-Based Systems*, vol. 128, pp. 20–33, 2017.

[26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS '13)*, pp. 3111–3119, December 2013.

[27] M. D. P. Salas-Zárate, R. Valencia-García, A. Ruiz-Martínez, and R. Colomo-Palacios, "Feature-based opinion mining in financial news: an ontology-driven approach," *Journal of Information Science*, 2016.