Spring 5-12-2018

Automated Development of Semantic Data Models Using Scientific Publications

Martha O. Perez-Arriaga
University of New Mexico

Martha Ofelia Perez Arriaga

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Trilce Estrada-Piedra, Chairperson

Dr. Soraya Abad-Mota, Co-chairperson

Dr. Abdullah Mueen

Dr. Sarah Stith

AUTOMATED DEVELOPMENT OF
SEMANTIC DATA MODELS
USING SCIENTIFIC PUBLICATIONS

by

MARTHA O. PEREZ-ARRIAGA

M.S., Computer Science, University of New Mexico, 2008

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico
Albuquerque, New Mexico

May, 2018

Dedication

“*The highest education is that which does not merely give us information but makes our life in harmony with all existence*"

Rabindranath Tagore

I dedicate this work to the memory of my primary role models: my mother and grandmother, who always gave me a caring environment and stimulated my curiosity.

All through my childhood, I spent most of the time with my grandmother: Adela, who with proverbs and a kind smile revealed her honest, empathic, trustworthy, and wise persona to me.

My mother, Clara, was always there for any people in need, and her personality full of cheer and encourage was contagious. She loved and enjoyed life deeply, while being resourceful, ethical, tolerant, and responsible under any circumstances.

Acknowledgments

First and foremost, I am profoundly thankful to my advisor and co-advisor Dr. Trilce Estrada and Dr. Soraya Abad-Mota, who appeared in my life when I truly needed academic guidance. Their ideas and human values inspire me to keep working in science. In addition, I sincerely thank the other two professors in my dissertation committee, Dr. Abdullah Mueen and Dr. Sarah Stith.

In particular, the chair of this dissertation committee, Dr. Trilce Estrada, supported me with her valuable expertise and constant recommendations at each stage of the development of this investigation, from the search of this topic, to implementation, alignment of ideas with writing, and completion. The co-chair of this dissertation committee, Dr. Soraya Abad-Mota, always presented me relevant recommendations, positive encouragement, and constructive comments of every aspect of this work, from inception to conclusion. Dr. Abdullah Mueen provided me useful feedback and suggestions to improve my presentations and dissertation work, and Dr. Sarah Stith contributed with fruitful conversations about my dissertation research, writing, and job search.

I give special thanks to Dr. Thomas Preston Caudell for teaching me the fundamentals on research work and the wonderful world of Neural Networks, as well as for being an exceptional advisor during my master's degree. I express thanks to Dr. Dorian Arnold for his support and mentorship while searching for an advisor to complete my degree. I acknowledge the Visualization and Data Science Laboratories, whose members gave me friendship and useful feedback for research and presentations: Shan, Takeshi, Victor, Cameron, Jacob, Jeremy, Matt, Xin Yu, and Rudy. I thank Haleh, who collected the last dataset used in this work. I am grateful to my study partners for the comprehensive examination, particularly, Josh and Joe. I thank all my professors and classmates, especially those who helped me to learn either about Computer Science or about life.

I recognize the National Council of Science and Technology (Consejo Nacional de Ciencia y Tecnología) for funding resources at the beginning of my studies and for strengthening the scientific investigation in Mexico. I am appreciative of the collaboration with all the people I had the opportunity to work at the Sandia National Laboratories, where I developed more problem solving and programming skills for issues in the field of Bioinformatics and Data Management.

I appreciate two extraordinary women who appeared along my path during a search for tools to improve myself, Dr. Susan Cooper for beneficial observations on writing and presenting technical information; and Mable Orndorff-Plunkett for being a supportive mentor, helping with the search of an editor, and giving me excellent pointers for presentations as well.

I am deeply thankful to my family and friends, especially to my daughter and husband for their patience and understanding on my continuous search of knowledge, equality, and justice. Tania Saraí was the major motivation to start this particular journey and shared her time with my studies. Aaron Andrew gave me the encouragement, strength, and emotional support to complete this important chapter in our pilgrimage. I express gratitude to Edmundo because he always encouraged me to pursue my passion for science and challenged me to search for the truth of life. I am grateful to the rest of my family that even in the distance supported me unconditionally: Arturo, Ernesto, Guadalupe, Joaquín, Meg, Nicholas, Norma, Patricia, Robert, Sayward, Sergio, Trinidad, and Verónica. Finally, I have gratitude to my dear friends Anne, Claudia, Elisa, Mariela, Mercedes' daughter, and Mercedes' mom, who gave me cheerful moments and brightened my path with their presence.

AUTOMATED DEVELOPMENT OF SEMANTIC DATA MODELS USING SCIENTIFIC PUBLICATIONS

by

MARTHA O. PEREZ-ARRIAGA

M.S., Computer Science, University of New Mexico, 2008

Ph.D., Computer Science, University of New Mexico, 2018

Abstract

The traditional methods for analyzing information in digital documents have evolved with the ever-increasing volume of data. Some challenges in analyzing scientific publications include the lack of a unified vocabulary and a defined context, different standards and formats in presenting information, various types of data, and diverse areas of knowledge. These challenges hinder detecting, understanding, comparing, sharing, and querying information rapidly.

I design a dynamic conceptual data model with common elements in publications from any domain, such as context, metadata, and tables. To enhance the models, I use related definitions contained in ontologies and the Internet. Therefore, this dissertation generates semantically-enriched data models from digital publications based on the Semantic Web principles, which allow people and computers to work cooperatively. Finally, this work uses a vocabulary and ontologies to generate a structured characterization and organize the data models. This organization allows integration, sharing, management, and comparing and contrasting information from publications.

Specifically, this work develops automated methods for Information Extraction, Semantic Analytics, Information Modeling, and Information Retrieval. I research how to deal with the diversity of tabular and document layouts, and detect, extract, and organize information from tables embedded in digital publications using an adaptive method.

To understand relevant information in documents, I interpret and enrich tables, disambiguating concepts and finding unique definitions from a general ontology and the Internet with an unsupervised learning method, while keeping the provenance information for each publication. Furthermore, I apply Natural Language Processing methods to discover and extract semantic relationships from text with non-standard vocabulary and various writing styles.

Applying these methods to a variety of digital publications, I formally characterize semantic data models in a machine-readable format. These models allow us to create a network of publications to integrate, manage, and analyze information not only in a specific domain, but also among disciplines.

Contents

List of Figures

List of Tables

Glossary

AER	American Economic Review.
DOI	Digital Object Identifier.
ER	Entity-Relationship.
$IEEE$	Institute of Electrical and Electronics Engineers.
$HTML$	Hypertext Markup Language
$JSON$	JavaScript Object Notation.
$JSON - LD$	JavaScript Object Notation-Linked-data.
$K - NN$	K-Nearest Neighbor.
LSI	Latent Semantic Indexing.
LWR	Locally Weighted Regression.
NLP	Natural Language Processing.
$NELL$	Never Ending Learning Language.
$NoSQL$	Not Only Structured Query Language.
OBO	Open Biomedical Ontologies.

Glossary

OIE	Open Information Extraction.
OWL	Ontology Web Language.
POS	Part of Speech.
PDF	Portable Document Format.
PMC	PubMed Central.
RDF	Resource Description Framework.
SDM	Semantics Data Model.
SIO	Semanticscience Integrated Ontology.
SVM	Support Vector Machines.
TAO	TAble Organization.
TFIDF	Term Frequency Inverse Document Frequency.
URI	Universal Resource Identifier.
URL	Uniform Resource Locator.
XML	Extensible Mark-up Language.

Chapter 1

Introduction

The massive production of a variety of digital documents and the lack of a defined context, as well as a lack of standards to present information in some areas hinder the prompt analysis of this information. Current methods to analyze this information focus on specific formats and domains. This investigation develops data models enriched semantically with relevant information for rapid understanding and analysis, as well as for easy interoperability among publications in different domains.

Data Science and Big Data provide methods to discover knowledge immersed in an extensive quantity of information, and to make decisions despite issues related to the volume, variety, veracity, value, and velocity of production of information. In addition, the Database area organizes data to access and manage information systematically. This dissertation intersects these areas with Machine Learning to solve issues related to knowledge discovery on digital publications, representation of information, and semantic analytics.

1.1 Overview

The velocity of the production of scientific publications keeps growing at different rates [1]. By 2014, Google Scholar had about 160-165 million digital documents including journals, conferences, theses, and other reports [2]. Of these documents, about 100 million are publicly available for scholars and researchers, and pertain to different areas, such as engineering, medicine, and social science [3]. Currently, the academic search engine BASE [4] contains more than 124 million documents from 6,196 different sources.

Regardless of the method used, either quantitative or qualitative [5], researchers perform common tasks that include identifying information related to their interest; annotating definitions; discovering, validating, comparing, and evaluating information; and ultimately, gaining knowledge. Moreover, the volume, variety, and velocity of production of information hinder the use, detection, and querying of information in publications and, therefore, can impact with any interaction among disciplines.

To solve the previously mentioned issues and facilitate research tasks, I design a conceptual schema to build semantically enhanced models from scientific publications. In particular, I develop a framework to create a characterization of digital documents systematically, that is, a semantic data model. The representation of concrete information extracted from publications should contain semantic cues to facilitate detection, integration, management, sharing, comparison, and querying information from articles. In addition, this representation should be domain agnostic.

J. Peckham and F. Maryanski review different semantic data models and establish the common components in these models [6]:

> a) relationships between data objects that support the manner in which the user perceives the real-world enterprise, and b) semantics for the

> relationships that specify the acceptable states, transitions, and responses of the database system.

J. Peckham and F. Maryanski refer to *enterprise* as the piece of real world represented in a database. The semantic data model concept from the Database field can be applied to digital documents, which may contain a collection of semantic relationships between concepts. However, the concepts and relationships in the documents are not defined a priori and need to be identified dynamically.

In this framework, the main component of a semantic data model is a digital publication. Although scientific articles contain formal vocabulary, no unified vocabulary exists to describe a study even in the same domain. For instance, the concepts *glycemic control* and *diabetes treatment* can be used interchangeably in the same document. In addition, the scientific publications present information in different formats, for example, Hypertext Markup Language (HTML) and Portable Document Format (PDF), as well as, a variety of publishing standards and guidelines exists.

The Health Sciences define their standards for publishing research articles. For instance, the digital library PubMed Central [7] (PMC) assigns identifiers to accepted publications. Journals and conferences of other domains, such as the *American Economic Review* [8] (*AER*) and the Institute of Electrical and Electronics Engineers [9] (IEEE), define submission guidelines for publishing research and require specific layouts for camera-ready publications. The Semantic Web Science Association [10] defines rules to improve sharing and promoting semantic scholar articles. Besides its publishing guidelines, the Association for the Advancement of Artificial Intelligence [11] requests metadata for publications.

The importance of metadata has arisen in several areas because it allows us to identify information rapidly, such as the title and author of a publication. Metadata is especially useful for the research areas that produce large volumes of data, such as Genomics. The large size of genomic data makes it difficult to store all this information. Metadata management systems exist to preserve the most relevant data and to maintain data features of interest [12]. Although this type of system is useful, expertise on a specific domain is necessary to tailor a metadata management system and to fulfill researcher's needs. Formats including tags, for example, HTML, facilitate the inclusion of metadata. Although, PDF is not based on tags, it allows us to include basic metadata.

PDF includes internal specification and location of elements in a document; the main benefit of using this format is that it does not require additional software or hardware to read this type of document [13]. However, D. Shotton [14] mentions the difficulty of using information in PDF articles, describing the PDF format as contradictory to the qualities of the Web (i.e., static, difficult to link, non-machine readable and non-interactive). Despite these non-interactive qualities, PDF documents are widely used to disseminate science.

As stated by T. Bernes-Lee and his colleagues [15], "the power of the Web stems from the linking it makes possible." Methods to make PDF documents easier to navigate exist. For instance, the framework SALT [16] semantically annotates a LaTex document, and creates metadata and embedded hyperlinks in this format. Not every scientific writer, however, uses LaTex syntax to create articles. In addition, F. Ronzano and H. Saggion [17] convert PDF articles into a Web representation for easier navigation. On the other hand, D. Shotton semantically enhances online publications in HTML format with elements such as the Digital Object Identifier (DOI), hyperlinks, and semantic mark-up text with linked data to additional information [14] to make digital publications analysis easier and more shareable. These works advance the management of annotated references in scientific documents and provide easier navigation regardless of a document's format. Moreover, researchers still spend most of their time selecting, analyzing, and comparing relevant information within publications for their research interests.

To facilitate reading and analyzing digital publications, some works associate concepts with definitions from reliable and established databases, such associations are named *semantic annotations*. For instance, BioLit [18] helps associate concepts, that is, add hyperlinks, within the publications with definitions established in Life Sciences databases. In addition, L. Penev et al. use a markup language to include mandatory taxonomic definitions [19]. These measures enable researchers to select concepts within publications and to review definitions in the *Official Registry of Zoological Nomenclature* [20] and the *Encyclopedia of Life* [21]. Most of the existing tools to semantically annotate scientific publications focus on the Health and Life Sciences.

Summarization can be used to analyze information rapidly and to compare information among publications. H. Saggion and F. Ronzano [22] mention the importance of summarization to access scientific information. Using semantic annotations for concepts and context in a scientific document, this investigation develops a short summary, that is, a synthesis of a publication. A. Nenkova and K. McKeown [23] present a survey with several techniques to summarize text written in different areas, such as Web pages, emails, and scientific documents, highlighting the importance of including context and scoring sentences for summarization.

To compare and relate publications, some research has focused on reference analysis. The relationships among different publications cited by each other ensures the existence of some kind of relationship. For example, SimRank [24] uses mainly Bibliometrics, that is, reference counting, and title of publications to infer relationships among them. Citations are relevant, but they do not necessarily ensure finding semantic relationships; therefore, this framework complements these works using semantic information to compare and contrast scientific documents.

Furthermore, semantic interoperability refers to having integrated information systems, which are context responsive. A. Sheth points out the importance of semantic interoperability to achieve information systems that are easy to use, contain context to query processing, and hide syntax and structural heterogeneity [25]. The analysis of a dynamic production of publications require to include semantic cues from different sources while generating these documents. S. Peroni [26] introduces the semantic publishing and referencing ontologies to facilitate publishing and referencing scientific articles. Because this kind of publishing is not widely used yet, a need exists to integrate and channel semantic information in documents.

To include semantic interoperability in the models, I integrate information from diverse sources, that is, metadata, context, – tables –, and free-text using semantic relationships. Because of its simplicity in presenting important associations of information, a large proportion of this investigation studies table information. Altough T. Green points out the importance of establishing standards for tables [27], it is not feasible to have a unique standard among scientific areas. Therefore, this study analyzes tables and their diversity of presentation, formats, and content.

In this investigation, I use a semantic data model in two ways. First, I use it as a representation for understanding the content of tables embedded in a publication, which includes semantic annotations for concepts in tables and contextual relationships among concepts within a publication. Second, because a semantic data model represents the blueprint of a publication with concrete information derived from tables and metadata to identify a publication, I use a semantic data model as a synthesis of each publication.

The Semantic Web principles provide rules for presenting meaningful information in which people and computers can cooperate to have a global Web using machine-readable data [28]. Although the principles apply to the Web, they are important for digital publications regardless of their environment and format. Moreover, to achieve interoperability, that is, relate information of publications among scientific disciplines, the Semantic Web principles can help generate a machine-readable representation for digital documents. In particular, I use a vocabulary, integrated ontologies, and a machine-readable format. Lastly, I represent this information in an organized approach.

Therefore, the semantic data model allows us to keep minimal information and still assist in managing and analyzing large datasets of publications. This model with a flexible organization can be used in a database or other data management systems to create a network of models. A collection of models enables query processing, information sharing, integrating information, comparing and contrasting documents, and interoperability to collaborate with other scientists in a specific field or among disciplines. Because this investigation can integrate concrete and available information, it can potentially assist in reproducing research in diverse scientific areas.

This investigation is motivated by the importance of sharing and comparing information, as well as collaborating across scientific disciplines. The assumptions to create semantic data models from scientific documents include that they contain tables with concrete information, and text using specific and formal natural language. Though scientific publications are the main source used in composing semantic data models, these methods can be adapted and applied to other kinds of documents.

1.2 Thesis and Summary of Contributions

The velocity of production of a large quantity of scientific publications, which lack 1) a unified vocabulary, 2) a standard format in tables, and 3) semantic interoperability, especially those in PDF format, impedes the rapid analysis and retrieval of specific information from publications. On the other hand, it is not feasible to require a single vocabulary and tabular format across all scientific areas.

The challenges previously mentioned have prompted the thesis statement for this investigation. It is feasible to create semantic data models without knowing a priori their components: entities and relationtionships and to discover these components on scientific publications in any domain using automated methods. This investigation aims to facilitate the retrieval, integration, comparison, sharing, and interoperability among publications.

To accomplish this investigation, I develop a conceptual design to create a semantically enriched data model per publication. In particular, a framework helps demonstrate how to generate data models systematically using metadata, text, context, and – tables – embedded in publications, related ontologies, and a vocabulary. These models contain specific information and facilitate analysis, management, and contrasting information from publications.

1.2.1 Contributions

The information contained in this dissertation contributes to the field of Computer Science by introducing novel methods for addressing common challenges in the fields of Information Extraction, Semantic and Data Analytics, and Information Modeling on digital publications. In addition, this work is related to other research areas, including, but not limited to, Databases, Data Management, Semantic Web, Semantic Similarity, Model Generation, Software Engineering, and Knowledge Discovery. From the first three parts of this investigation, my main contributions follow:

1. To extract table information from digital publications, I develop a process in [29] with

   – an adaptive method for detecting and extracting table information from various layouts in PDF documents,

  - an annotated representation of the information extracted from tables, and
  - a Web-based prototype system for using the methods for the organization of tables.

2. For Information Extraction and Semantic Analytics of digital publications, I develop a process in [30] with

   - a method for the interpretation of tables, including processes for searching for unambiguous entities and extracting semantic relationships;
   - a method for integrating information from different sources (i.e., metadata, tables, text, ontologies, Internet); and
   - a machine-readable characterization of each digital publication as a synthesis.

3. For Information Modeling and Data Analytics of a set of digital publications, I develop

   - a conceptual design for dynamic generation of semantic data models, and
   - an approach for generating a semantic network of publications.

1.2.2 Development of Methods for Semantic Data Models

To accomplish the goal of generating semantic data models from publications, I perform tables and text analyses. Figure 1.1 depicts the development of this framework and guides the description of its main components. In the first part, **Discovery of Table Cells**, I develop heuristics and supervised pattern learning for automatic table detection and extraction from different layouts of tables and documents. I evaluate these methods and compare their results to a baseline. I perform a set of experiments on different layouts of documents to measure the number of tables detected and the number of data cells correctly extracted from tables. Finally, I develop a prototype system to submit digital documents and produce a document containing the tables extracted and organized for each publication.

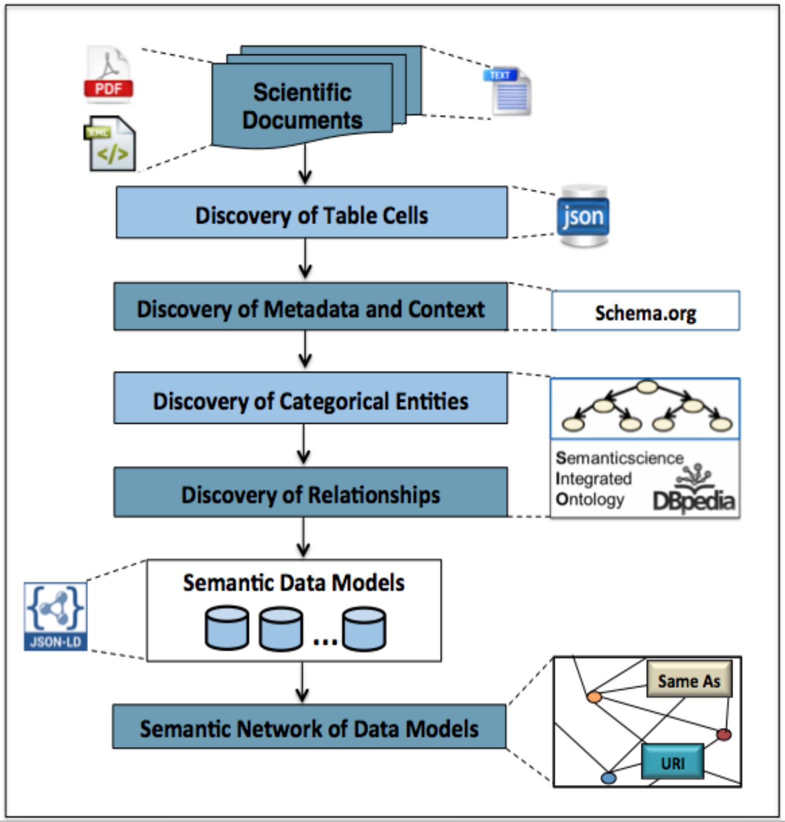

PDF
XML
Scientific Documents
TEXT
Discovery of Table Cells
json
Discovery of Metadata and Context
Schema.org
Discovery of Categorical Entities
Semanticscience Integrated Ontology
DBpedia
Discovery of Relationships
JSON-LD
Semantic Data Models
...
Same As
URI
Semantic Network of Data Models

Figure 1.1: The Semantic Data Modeling Framework

The second part of this framework includes **Discovery of Metadata and Context, Categorical Entities, and Relationships** (see Figure 1.1). To understand the content of tables or to interpret tables, I present methods to find context and entities (i.e., concepts) in tables. The entities are related to contextual relationships within a publication and to definitions taken from an ontology. I create a synthesis of a publication composed of a publication's metadata, semantic relationships, and tables' entities with annotations, using statistical and unsupervised learning methods. I evaluate the methods for table interpretation, including entity recognition, annotation, and disambiguation. Moreover, I evaluate the extraction of semantic relationships, as well as the quality of the annotations and semantic relationships.

Finally, this framework includes the generation of **Semantic Data Models** and **Semantic Network of Data Models** (see Figure 1.1). In this part, I formally define the components to create a semantic data model for each publication. Each model is used as a synthesis of a publication, facilitating to collect and manage digital documents. A collection of models enable us to integrate information from different sources and domains. I present an approach to build a network using semantic data models derived from a set of publications with different topics. This network allows us to compare and contrast semantic relationships among publications.

The remainder of this dissertation is divided into five components. Chapter 2 presents a literature review of related work for the different stages of the development of this framework. Chapter 3 shows the process of identifying and extracting table information from digital publications. To understand the content of tables, Chapter 4 presents methods to detect metadata and context, perform table interpretation, and extract entities and semantic relationships to produce a synthesis for each publication. Chapter 5 shows a generic schema of the components of a semantic data model, an approach to create a network of models, and a comparison of semantic relationships among publications. Chapter 6 contains conclusions and recommendations derived from this dissertation, as well as suggestions for future work. In addition, Appendix A contains the conceptual design of a semantic data model, Appendix B contains definitions of semantic relationships from the Semanticscience Integrated Ontology, and Appendix C contains the context to represent a semantic data model. Lastly, I include the bibliography used for this work.

Chapter 2

Literature Review

This chapter reviews methods used to obtain the main elements to compose the data models. The first part of this review presents related work to analyze tables in digital documents. Second, I present important work to extract semantic relationships. Third, I briefly describe methods used to integrate information and design semantic data models, as well as tools used to annotate scientific publications and summarize information.

2.1 Table Analysis

To begin, I present general definitions of table identification, document converters, and related work to perform table analysis. Because tables within digital publications present simple, yet important associations using structured cells, several doctoral works have studied tables and their functions [31], [32]. X. Wang presents a thorough work on the analysis of tables [32], where she distinguishes the importance of the physical and logical elements inside tables. Her doctoral work describes the main table functions: to understand the information, to search for information, and to interpret or compare information. M. Hurst [31] describes in more detail the model representation of tables, including graphical, physical, functional, structural, and semantic components.

J. Hu and her colleagues [33] define the table identification problem in two parts: *table detection* and *table recognition*. *Table detection* refers to the identification of the existence of tables embedded in documents, while *table recognition* refers to identifying the actual structure of a table [33]. Table identification is also known as *table processing* [34]. Table processing analyzes the physical and logical structures of a table. The literature offers two detailed reviews for table processing [34], [35].

Commonly, documents containing tables exist in formats HTML, XML, text, and PDF. S. Balakrishnan et al. [36] detect high quality Web tables with simple heuristic rules and a Machine Learning classifier. Two multi-kernel Support Vector Machines (SVM) classifiers with three Gaussian kernels identify vertical and horizontal tables. The features for the classifiers are the number of rows and columns, the mean/variance of the string length in a cell, the ratio of data cells, the ratio of cells with `<th>` tags, and the number of distinct tokens in a table.

The lack of tags to identify tables in PDF documents poses more challenges extracting tables from this type of documents. Therefore, extracting information from PDF documents is a difficult task. To facilitate table identification, recent works use PDF converters [37], [38], [39].

T. Hassan [40] uses the PDF converter jpedal, TableSeer [38] uses PDFBox, Y. Liu et al. [39] use PDFlib TET, and TAO [29] uses PDFMiner [41]. Although these tools are powerful, they may suffer from text sequence errors [42]. Text sequence errors occur when the converter tool shows the text in a different order than the original PDF document. Several converters, however, provide coordinates for processing the right order.

Some approaches to identify tables embedded in PDF documents include heuristics [43] and statistical methods [39]. TableSeer [38] uses the PDFBox converter and a page box-cutting method to detect tables. The method is based on heuristic rules to detect metadata in the document and table elements. The TableSeer algorithm relies on table structure, font size, and keyword matching to identify table candidates, examples of these words are *table* and *form*. TableSeer also relies on white spaces to detect structure and tables, assuming the tables have the same font size and ignoring boxes with different font sizes. TableSeer also extracts metadata information from the document and offers an algorithm to index and search table information. Although TableSeer is functional and complete, at times its output makes it difficult to reconstruct the table content in its original position. It yields XML to represent tables found in a document.

J. Fang et al. [44] use PDF parsing, page layout analysis to detect multiple-columns in a document, and an analysis of lines and whitespaces to detect separation between tables. This approach has been used to detect tables in two million e-books by Founder Corporation, and showed better recall than TableSeer when tested on 70 documents selected from the TableSeer dataset. This commercial work, however, is unavailable for comparison.

PDF-TREX [43] presents a heuristic approach to table recognition and extraction from PDF documents. The PDF-TREX algorithm distinguishes between text and table areas in the document. The heuristics work in a bottom-up way to align and group *"content elements, exploiting spatial relationships among them."* However, this heuristics extract tables only in single-column documents.

To overcome the limitations of these works, such as relying on keywords to detect tables, identifying tables only in one-column documents, and a lack of a representation to recover an original table's format, I develop novel methods using a combination of heuristics and statistical methods in TAO [29]. This approach 1) relies on the PDF converter PDFMiner [41], 2) uses a supervised method to learn the alignment between columns and rows to identify tables, 3) supports the identification of tables in single and multiple-columns, and 4) uses a comprehensive representation of a table's content.

Besides identifying a table, it is crucial to understand and interpret a table's content. *Table interpretation* is the process of understanding the information contained in a table. M. Gobel and his colleagues [45] describe *table interpretation* in more detail:

> Rediscovering the meaning of the tabular structure. This includes: (a) functional analysis: determining the function of cells and their abstract logical relationships; (b) semantic interpretation: understanding the semantics of the table in terms of the entities represented in the table, their attributes, and the mutual relationships between such entities.

To interpret tables in digital documents, some work studies certain formats [46], [47], [48], [49], for instance, HTML and PDF. For the former, M. Cafarella and his colleagues develop Webtables [46] to classify and interpret tables from a large number of Web documents. Their work detects metadata for tables, that is, headers in a large scale analysis of tables. If tables lack a header, this method finda a synthetic header by using a database to match the contents to a known label. From this analysis, Cafarella et al. create a database with attribute co-occurrence statistics, including the attributes and their frequencies.

Webtables have evolved throgh the years. Another paper related to Webtables is when P. Venetis and his colleagues [50] annotate table content to improve table search. They assign a category to metadata related to a column, using a maximum likelihood model and the information contained in the column. They create two databases from the Web, the first database is built from text patterns, for instance, using the words “such as”. This database has the form isA(instance, class). The second database uses TextRunner [51], a method to extract relationships of the form (argument1, predicate, argument2). The isA database has a score to avoid giving more weight to those more frequent labels. The databases cover many domains, but can be noisy. Therefore, a class label (entity) is assigned to a column of a set of cells, and a relationship label is assigned to represent a binary relationship (property) between a column class and the set of cells in a column.

The main differences between Webtables and this framework are that the latter identifies and annotates tables from PDF documents, and preserves information of each table’s source document. Webtables understand tables in a more general way, while this framework finds precise information for each table. For example, identifying another publication related to a concept to understand a table and its publication. Because a table presents important information related to its source document, this framework also considers the document’s context and text to interpret a table, and consequently to understand its source document. In addition, this framework preserves information about the table’s source document and its provenance.

Using table interpretation to extract and understand tables in PDF is less common than for HTML. Xonto [48] extracts syntactic and semantic information from PDF documents. Xonto uses the Description Logic Programs+ ontology representation language with descriptors for objects and classes to represent the extracted information from tables semantically. Even though Xonto uses lemmas and part of speech tags to identify entities, it lacks an entity disambiguation process.

Texus [49] is a method to identify, extract, and understand tables. Texus claims to be format independent because a text or HTML document can be readily converted to PDF. R. Rastan, H. Paik, and J. Shepherd convert PDF documents to a model; find tables, cells, rows, and columns; and perform functional and structural analyses. Their analyses help to understand tables. Functional analysis finds each cell’s role (i.e., headers or data), where cells can belong to box head, stub, stub head, and table body, and the cells' function include data, access, or attribute. The headers can function either as attributes or access cells. Structural analysis finds logical relationships between cells.

To understand tables and overcome previous research's weaknesses, such as lack of a semantic analysis and the ability to link provenance information to table interpretation, I propose an integral approach as found in [30]. This approach uses tabular structures and unstructured text in a publication to automatically find context, unique entities, and metadata to identify a particular publication.

The most direct method to disambiguate concepts is to use a simple source of knowledge, i.e., a dictionary [52] such as WordNet [53]. This approach suffices when the dictionary contains all the entities of interest and when each entity is free of ambiguity. Given the variety of concepts and topics appearing in publications, these conditions are rarely observed. P. Ferragina and U. Scaiella [54] use sources of knowledge containing structured and unstructured information, such as Wikipedia, which feature more information than a dictionary.

The Semantic Web [55] represents entities and relationships known as *triples.* DBpedia alone contains 4.7 billion triples [56], including the areas of people, books, and scientific publications. However, most triples from scientific publications are missing in the Semantic Web. To understand tables, V. Mulwad and colleagues find annotations for entities [47] using knowledge sources that follow the Semantic Web principles. Therefore, the Semantic Web databases can complement scientific publications and help to disambiguate concepts.

This framework uses mainly DBpedia [56] to find a description of each entity. DBpedia contains semantic information that follows the Semantic Web principles and it is 1) a curated ontology originated from Wikipedia and 2) a structured ontology with specific entities' properties. In addition, this framework develops a method using the unsupervised method *Latent Semantic Indexing* [57] and the publication's context to discover explanations of entities on the Internet. This framework, therefore, allows us to disambiguate concepts and use them to support the extraction of semantic relationships.

2.2 Semantic Relationships

For the second part of this review, I focus on related work to discover semantic relationships. V. Storey [58] presents the importance of semantic relationships for data management, a comprehensive analysis of semantic relationships, as well as an analyzer of relationships. This investigation is motivated by A. Sheth and his colleagues' work [59], where they describe with great detail the importance of semantic relationships and ontologies to achieve semantic interoperability.

Several approaches are used to discover semantic relationships, from finding fixed patterns [60] to using unsupervised methods [61], [51]. D. Hristovski et al. find relationships using fixed patterns [60]. M. Ruiz-Casado, E. Alfonseca, and P. Castells [62] find them using entity sense disambiguation, pattern extraction, pattern generalization, and identification of new relationships. A. Moro and R. Navigli [63] use a distributed soft kernel k-medoids method.

Three important works to extract semantic relationships include the Never Ending Learning Language (NELL) [64], PATTY [65], and Open Information Extraction (OIE) methods [61], [66], [51]. NELL [64] creates a knowledge base of categories and relationships. This approach contains a set of subsystems to extract patterns and novel relationships from the Web, to classify noun phrases, and to infer new relationships from their knowledge base. The approach PATTY [65], based on frequent itemset mining, detects relationships from the Web using syntactic, ontological, and lexical features.

The Open Information Extraction methods successfully find new relationships with no previous knowledge of them [61], [51]. However, A. Fader, S. Soderland, and O. Etzioni state that *"it can find incoherent and uninformative extractions."* To improve OIE, Fader et al. develop Reverb [66], which finds relationships and arguments using part of speech tags, noun phrase sentences, and syntactic and lexical constrains. Reverb uses a corpus built offline with 500 million Web sentences, enabling Reverb to match arguments in a sentence heuristically. In addition, Reverb uses a logistic regression classifier with a training set of $1,000$ sentences from the Web and Wikipedia to assign a confidence for each relationship. An evaluation showed that Reverb performs better than another OIE method called Textrunner [51].

The second generation of OIE [67] includes Reverb and R2A2. The latter performs better than Reverb does, and both are highly scalable. Besides a linguistic and statistical analysis to extract relationships, R2A2 includes an argument learner. This consists of classifiers using features–such as specific patterns, punctuation, and part of speech tags–to identify arguments.

Initially, this framework [30] uses Reverb to detect semantic relationships within a publication. The high ranked relationships from Reverb may miss important entities, however. Therefore, I take advantage of the affluence of information in tabular structures to find the main components of relationships, and develop a method to retrieve relationships with relevant entities from a publication. To find semantic relationships from text, I use a combination of approaches: a) the statistical method Latent Semantic Indexing, b) Natural Language Processing, and c) pattern matching. Similarly to OIE methods, this framework uses unsupervised learning and heuristics; however, it only focuses on relationships containing entities of interest.

In general, related work to extract semantic relationships examines mainly Web and text formats, lacks methods to systematically match a source document with its relationships, and lacks a representation to organize relationships per document. To facilitate researchers' work, this framework finds important relationships from publications and organizes them in an integrated document, preserving the original publication's information and provenance for further consultation.

2.3 Semantic Data Integration

The last part of this section includes related work to integrate information, to represent semantic information, to provide semantic annotations, and to summarize and compare information from digital publications.

Data integration refers to map heterogeneous sources of information to use and query information as a single unit. The variety of schemas, formats, and domains representing data makes it difficult to integrate information. A. Doan, A. Halevy, and Z. Ives [68] mention two general architectures for data integration: warehousing and virtual integration. *Warehousing* refers to loading individual sources into a physical database–or a warehouse. In contrast, *virtual integration* refers to using the individual sources in their original form and accessing them as necessary for querying information. Both architectures use a mediated schema that contains relevant properties of the sources to integrate their data. This framework integrates information using an architecture similar to warehousing, without necessarily using database representation. In addition, this approach's mediated schema contains a conceptual design of entities and relationships, which are not established in advance.

Semantic mapping refers to matching information from one schema to another or between ontologies. This mapping recovers the same information into a unique representation. C. Hian and his colleagues [69] present a framework to mediate the possible conflicts between different sources, composed of the application domain;

the relationships between attributes in the sources, and the semantic types in the domain; and the relationships one-to-one from context to an individual source.

This framework differs from semantic mapping because I do not know *a priori* the information to integrate. Instead, I gather common elements within publications, and generate a characterization using general ontologies and a vocabulary to prepare each publication for further integration and mapping. In particular, I develop a conceptual design to organize relevant concepts and relationships that can be integrated with other documents having the same context. In addition, this framework is based in *linked data* [70], a general method to create typed links between data from heterogeneous systems to interoperate at the data level, using the Universal Resource Identifier to represent data.

Semantic data models include two main components: 1) relationships between entities and 2) well-defined semantics for the relationships [6]. R. Hull and R. King [71] describe in detail how semantic data models allow to a) separate concepts in physical and logical forms, b) minimize overload of relationships, and c) represent abstract concepts and relationships. These models have been used to include semantics in a database schema, which may be difficult to represent.

F. Steimann [72] presents a method to model concepts in an object-oriented approach, and the importance of roles in polymorfism. For instance, a person can be an employer or employee. Then, depending on the role, the model can perform a specific task. The models represent data objects and relationships known *a priori*. For example, the *World Traveler Database* modeling shown in [71] represents defined entities and relationships among them. In contrast, I develop a conceptual design using discovered entities from tabular layouts embedded in a digital document, as well as relationships from 1) tables, 2) text within a document, 3) the general ontology DBpedia, and 4) the Internet.

The importance of annotating scientific documents has increased in different areas, especially in the Health Sciences. Tools to annotate scientific publications, such as Reflect [73], Biolit [18], and BioC [74], are mainly useful in the biological and medical domains. BioC [74] generates a format in XML to provide relationships and annotations from scientific publications. Similarly to BioC, my approach uses a design to define a document with annotations and relations. BioC offers data classes for developers to generate annotations, while this framework finds the annotations without programming involved by a researcher. BioLit [18] reviews a publication's full text and finds a specific ontology and database terms, the terms are generated as metadata in an XML format. BioLit also provides context information for the metadata.

Because the goal is to create interdisciplinary semantic data models, the model generation is based on a general ontology–DBpedia–to represent the knowledge. In addition, I use the knowledge on the Internet to enrich the publications and learn from different domains. For the definitions of relationships, I use the Semanticscience Integrated Ontology [75] to maintain structure and represent the knowledge derived from the scientific publications. To represent the publication's metadata, I use the general vocabulary schema.org [76].

J. Piskorski and R. Yangarber [77] present a review for Information Extraction approaches. They state that *"the task of Information Extraction is to identify a pre-defined set of concepts in a specific domain"*; while true, it takes time identifying non semantically defined concepts and domain in documents. This framework presents a novel approach to preparing documents for Information Extraction, and dynamically generates and organizes a semantic data model with context and concepts.

Summarized information provides a short representation of the content of publications and can be used to analyze these documents faster. Automatic approaches have advanced the summarization research area. M. Allahyari and colleagues [78] present a recent survey of text summarization methods and point out the importance of ontologies in summarization. A. Nenkova and colleages [79] present a thorough description of methods used for this endeavor in different genres, such as medical, emails, journals, and news. S. Teufel and M. Moens [80] stress the difference between summarizing scientific versus news documents, the former being less repetitive and more concise in their arguments. I also notice that space restriction in an article can make the narrative even more succinct.

Some approaches to summarize publications use rethoric analysis and identify the section of each relevant sentence. For instance, to summarize publications, S. Teufel and M. Moens analyze rethorical status that is based on analyzing how organization, contribution, and citations appear in scientific writing, as well as relevance in sentences. A recent work by E. Baralis et al. [81] uses frequent itemsets to summarize documents with sentences from the documents. PaperViz [82] allows researchers to manage references and metadata, and to search documents in a collection of publications. DeepDive is a system using machine reading to integrate information and facilitate building knowledge bases [83]. This framework generates a short summary, synthesizing a publication based on ontologies and integrating information using semantic relationships with relevant concepts, which can also be used to build knowledge bases.

Novel methods for semantic publishing have emerged. For instance, S. Peroni [26] introduces the semantic publishing and referencing ontologies to create metadata and include semantic annotations for publishing and referencing scientific articles. PDFX [84] converts a scientific document from PDF to the Extensible Mark-up Language (XML) with annotations to identify different elements in a publication, including author, title, body, references, and tables. In addition, F. Ronzano and H. Saggion [17] provide a Web representation of each PDF article, which contains reference annotations, as well as relationships of the form subject, verb, and object, using abstract and sentences in a publication.

To relate scientific publications to each other, several works focus on studying references to discover relationships. SimRank [24] uses references and titles of publications to infer relationships. A. Nenkova and colleages [79] mention the importance of using citations to discover relationships with related, previous, and future work. This framework complements the previously mentioned work because it uses a synthesis of semantically enhanced information, such as context and semantic relationships to compare and contrast scientific documents, even if documents have no citations in common. The conceptual schema of this framework generates semantic data models systematically from digital documents, and allows us to create networks of publications to compare and contrast specific information. S. Uddin et al. [85] mention the importance of networks to co-relate authors of scientific publications. Instead, this approach detects important nodes, i.e., entities, and their relationships derived from different publications to relate them to each other.

Chapter 3

Table Detection and Information Extraction

Digital documents communicate scientific research and enable the exchange of information in a variety of layouts and formats. In addition, the number of digital documents has been increasing rapidly. To take advantage of the knowledge embedded in an ever growing information source better, effective tools for automatic extraction of relevant information are needed. This information is embedded in data resources within a document, such as tables.

Tables exist as crucial elements for presenting information in scientific publications. Publications use tables to represent and report specific findings of the research. Current methods used to extract data from tables in PDF documents lack both precision in detecting information found in tables in diverse layouts and a detailed representation of the extracted table information to rebuild a table as the one presented in its original source.

I investigate methods for detecting, extracting and organizing information from tables within PDF documents. To demonstrate the methods, I develop a prototype system, *TAble Organization (TAO)*, to support information extracted from tables with diverse tabular and document layouts. In addition, the methods demonstrate how an adaptive learning method can overcome limitations of related work.

3.1 Discovery of Table Information

The vast amount of information on the Internet includes scientific documents. These documents communicate findings of research studies and exchange information in different areas. However, the large production of such documents makes it difficult to analyze them at the same speed they are produced. Scientific documents growth has increased about 8-9 percent per year in the past six decades [86]. In addition, the myriad topics and ways to present the information of digital documents hinder the analysis and management of their contents.

Tables are found in most sections of a document, such as introduction, methods, results, and discussion. A study of table classification by S. Kim et al. showed that approximately 75 percent of the information from tables, in different domains and journals, appear in the results section [87]. Therefore, tables offer concrete information that can help to extract and acquire knowledge.

Despite the benefits of tables, the process of detecting and extracting information from tables is not straightforward due to diverse factors, including the variety of content, design, and layout in tables. In addition, tables embedded within publications are mixed with other elements, such as free text and figures. There is a lack of standard formats for presenting tables in scientific publications [27]. Moreover, depending on information, authors design tables using different numbers of columns and rows for each table.

To solve these issues, I develop methods to detect and extract table information in digital documents automatically. The methods also organize information of tables within the documents. The output produces a format that facilitates information exchange and is easy to process automatically.

Most of the scientific publications adhere to strict formatting standards and undergo a review before publishing them. Therefore, these publications contain high quality elements, such as tables, the main subject of this chapter. Even though the primary target for this investigation refers to scientific publications, any document containing tables benefits from the methods developed in this work.

The main contributions are: 1) an adaptive method to detect and extract table's information, 2) an annotated representation of the information extracted from tables for further structural, functional, and semantic analyses, and 3) a Web-based prototype of the system at http://integra.cs.unm.edu.

The remainder of this chapter includes Section 3.2 Table Detection and Extraction to show the methods for document conversion, detection of table candidates, and extraction of table information. Section 3.3 Evaluation shows the experiments, datasets, baseline, results, and discussion.

3.2 Table Identification

A table is a set of structured data cells that represent associations. In this section, I present the first process of this framework: **Discovery of Table Cells**, as described in Figure 3.1, where it receives a document and discovers information contained in table cells.

J. Hu and his colleagues divide the problem of table identification into two parts: *Table detection* is the process of identifying the existence of tables from a document, and *table recognition* of identifying and extracting the cells contained in a table [33].

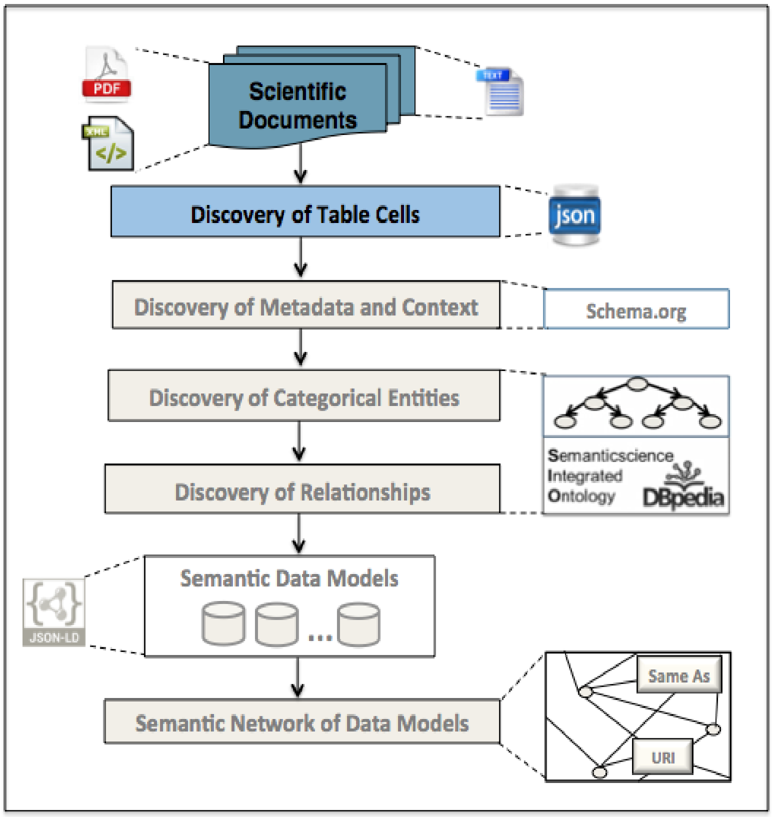

PDF
XML
Scientific Documents
TEXT
Discovery of Table Cells
json
Discovery of Metadata and Context
Schema.org
Discovery of Categorical Entities
Semanticscience Integrated Ontology
DBpedia
Discovery of Relationships
JSON-LD
Semantic Data Models
...
Semantic Network of Data Models
Same As
URI

Figure 3.1: The Table Identification Process

I present an approach capable of detecting and organizing tables within PDF and XML documents. This table identification process consists of three main modules: 1) document conversion, 2) table detection, and 3) table extraction.

I consequently refer to the table identification process as the TAble Organization (TAO) process (see Figure 3.2, based on [29]). The document conversion module relies on PDFMiner [41] to convert a PDF document into XML. PDFMiner's output includes separate XML tags for every character and every space. The table detection module parses this large XML version using a combination of layout heuristics to detect table candidates within the document. The table extraction module uses table candidates and a supervised learning method to find a table's content. Finally, TAO generates its output using a JavaScript Object Notation (JSON) document, which contains table information extracted from the document in PDF. As follows, I describe in more detail the three main modules of TAO.

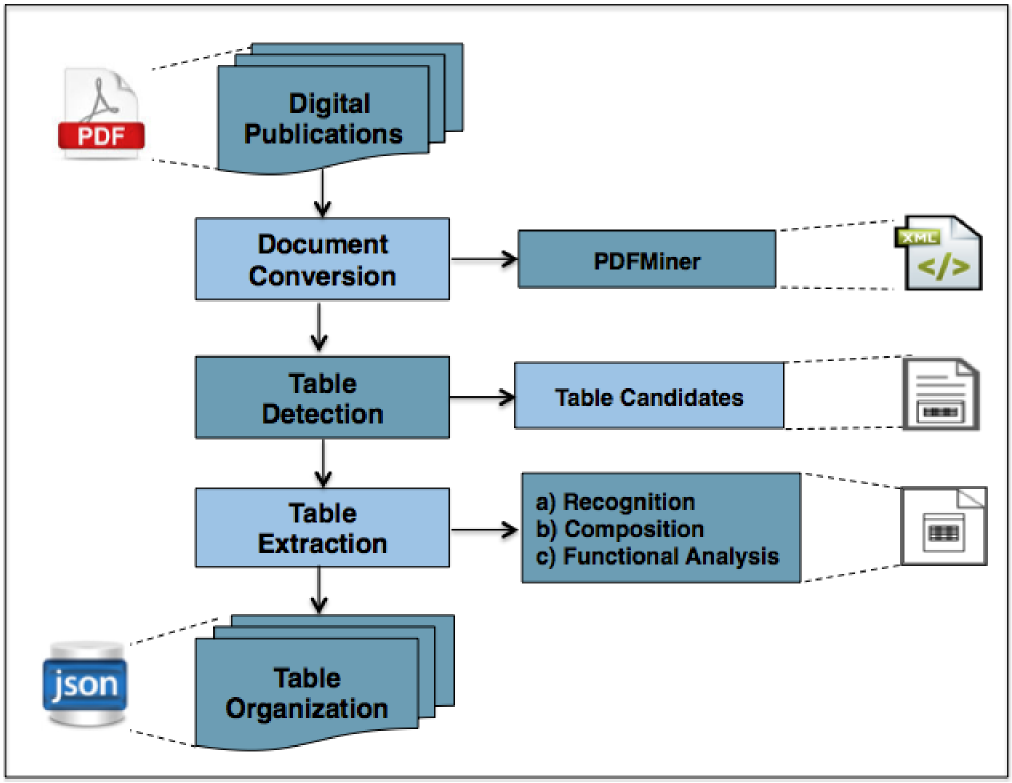


Figure 3.2: The TAble Organization (TAO) Process

3.2.1 Document Conversion

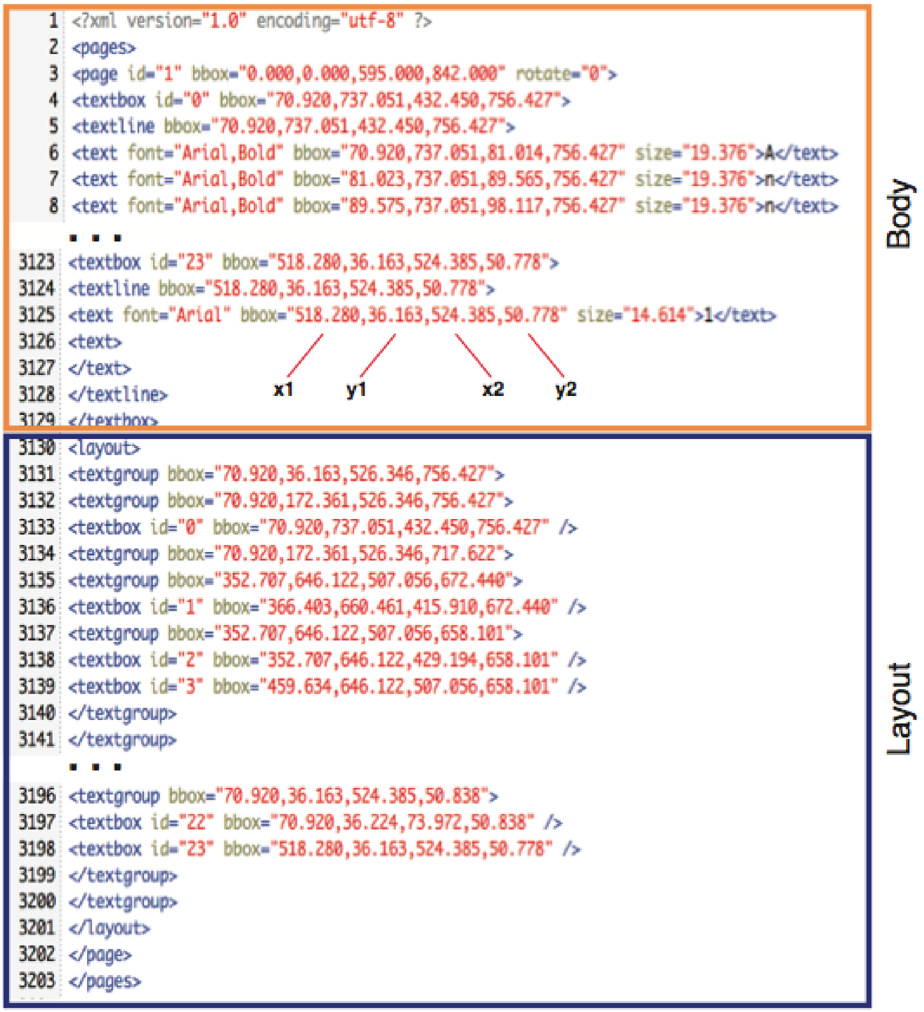
I perform document conversion using PDFMiner [41], a versatile extraction tool that converts PDF documents to text, HTML, XML, and tagged PDF formats. PDFMiner outputs the exact coordinates of the text, and other attributes as font style and size for each character into an XML file. After evaluatintg other state of the art PDF converters [88, 89], I select using PDFMiner because it is widely adopted and maintained. In addition, it provides a detailed layout of elements (i.e., figures, lines, and text), as well as a hierarchy of grouped elements based on their spatial relationships in the document.

PDFMiner converts a PDF file into an XML representation and generates a *body* and a *layout* for each page of the PDF document (see Figure 3.3). The layout contains a summary of the information structure, and the body contains the actual information of the document.

The body contains three main elements formed by *text boxes*, *text lines* and *text*. Text boxes include an identifier (id) and the top-left (x_1, y_1) and bottom-right coordinates (x_2, y_2) of the bounding box (bbox), see line 3123 in Figure 3.3. Text boxes are composed of text lines. Each text line includes its corresponding bounding box (bbox) coordinates, as seen on line 3124, as well as the text elements found within the limits of a text line. Text lines are composed of *text* tags. Each text tag contains a single character. Characters grouped in text lines form one or more words. PDFMiner also provides coordinates for each character, see line 3125 for the character '1'. Additional attributes associated to characters represent text font and text size.

The layout includes two elements: text groups and text boxes. A *text box* from the layout can also be found in the body, the only difference is that, in the layout, *text boxes* are organized into *text groups*. For example, see lines 4 and 3133 in Figure 3.3, the *text box* with id=0 is the same in both places. However, the one on line 3133 belongs to a *text group* represented on line 3132.

It is worth to note that the example in Figure 3.3 corresponds to a single page document containing one table. Nevertheless, the output produced by PDFMiner contains 3203 lines. For an IEEE research paper with 10 pages, the number of lines produced by PDFMiner ranges from [40,000-70,000] lines, depending on the



```
<?xml version="1.0" encoding="utf-8" ?>
<pages>
<page id="1" bbox="0.000,0.000,595.000,842.000" rotate="0">
<textbox id="0" bbox="70.920,737.051,432.450,756.427">
<textline bbox="70.920,737.051,432.450,756.427">
<text font="Arial,Bold" bbox="70.920,737.051,81.014,756.427" size="19.376">A</text>
<text font="Arial,Bold" bbox="81.023,737.051,89.565,756.427" size="19.376">n</text>
<text font="Arial,Bold" bbox="89.575,737.051,98.117,756.427" size="19.376">n</text>
. . .
<textbox id="23" bbox="518.280,36.163,524.385,50.778">
<textline bbox="518.280,36.163,524.385,50.778">
<text font="Arial" bbox="518.280,36.163,524.385,50.778" size="14.614">1</text>
<text>
</text>
</textline>
</textbox>
<layout>
<textgroup bbox="70.920,36.163,526.346,756.427">
<textgroup bbox="70.920,172.361,526.346,756.427">
<textbox id="0" bbox="70.920,737.051,432.450,756.427" />
<textgroup bbox="70.920,172.361,526.346,717.622">
<textgroup bbox="352.707,646.122,507.056,672.440">
<textbox id="1" bbox="366.403,660.461,415.910,672.440" />
<textgroup bbox="352.707,646.122,507.056,658.101">
<textbox id="2" bbox="352.707,646.122,429.194,658.101" />
<textbox id="3" bbox="459.634,646.122,507.056,658.101" />
</textgroup>
</textgroup>
. . .
<textgroup bbox="70.920,36.163,524.385,50.838">
<textbox id="22" bbox="70.920,36.224,73.972,50.838" />
<textbox id="23" bbox="518.280,36.163,524.385,50.778" />
</textgroup>
</textgroup>
</layout>
</page>
</pages>
```

Figure 3.3: PDFMiner's Conversion from a PDF Document

type of content. PDFMiner's output is comprehensive and long, and as such, it is practically impossible to parse and organize manually and it is not trivial to do so in an automatic way. In the following section, I use TAO to parse this XML output and to generate cleanly extracted and annotated table structures.

3.2.2 Table Detection

The table detection module refers to identifying tables within the XML output provided by PDFMiner. As shown in Figure 3.3, PDFMiner's output consists of XML elements described by their bounding coordinates that form boxes, and content including other XML elements or individual characters. This format does not discriminate between text in normal paragraphs and text in tables or figures, which is represented by *group boxes* and *text boxes*. Thus, TAO performs a structural analysis to identify sets of *text boxes* that are probable table candidates. The table detection process includes four steps: 1) comprehensive identification of *text boxes* within *text groups*, 2) distance calculation among *text boxes*, 3) identification of structural relationships, and 4) generation of table candidates. A description of each one of these steps follows.

As mentioned previously, *text boxes* are contained within *text groups* (see Section 3.2.1). For each *text group* I identify the set T containing all *text boxes*. The table detection process follows: first, I create a list containing all the *text box* identifiers (*id*) on each page. For each pair (a,b) of *text boxes* $\in T$, I compute their Manhattan distance using their top-left (x_1, y_1) and bottom-right (x_2, y_2) coordinates to fill in a distance matrix of coordinates (M_{diff}). The analysis of text boxes creates a matrix of differences. Each element in the matrix contains the distance of the top-left coordinates between a and b for the first dimension $(M_{diff}[a, b, 1] = (a.x_1 - b.x_1, a.y_1 - b.y_1))$ and the distance of the bottom-right coordinates between a and b for the second dimension $(M_{diff}[a, b, 2] = (a.x_2 - b.x_2, a.y_2 - b.y_2)$.

For instance, figure 3.4 shows six *text boxes* (a, b, c, d, e, f). The matrix M_{diff} stores the Manhattan distance for each pair of text boxes. The example depicts distance calculation for *text boxes* (a,b) and (d,e). For simplicity, the figure shows only the second dimension representing distances between the bottom-right coordinates

(x_2, y_2), thus the indexing [a,b,2] and [d,e,2] on top of the arrows. A similar concept is used to save distances for the top-left coordinates, where the indexing would be [a,b,1] and [d,e,1].

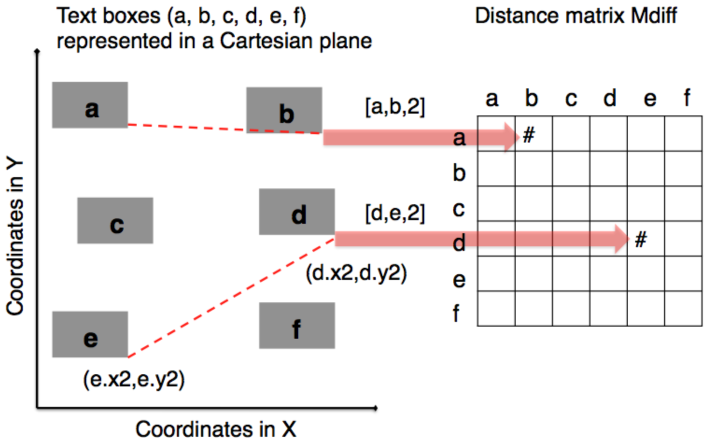


Figure 3.4: Identification of Text Boxes and Distance Calculation

I use the distance matrix to calculate alignment and proximity between *text box* pairs. This information allows us to determine global structural relationships, that is, rows and columns, among multiple *text boxes*. To identify columns, two text boxes have an alignment if their distance is at most $t = \pm 23$ points to either side (i.e., about 8 mm). This distance represents two characters plus a minimal separation in between columns with text in Helvetica or Arial font of 10pts, which are two of the most used fonts in research papers, technical reports, and magazines. For rows, the distances are smaller since tables in documents with a portrait format grow more frequently vertically than horizontally, resulting in compact rows with small inter-row distances. Thus, the vertical distance threshold is $t/10$ for the distance above a *text box*, and $t/2$ for distances below a *text box*.

Once all the distances are calculated in M_{diff} and set up the thresholds, I evaluate *text box* pairs. Pairs whose proximity is less than or equal to the thresholds are aligned, forming either a row or a column. Then I join all the text boxes in the same row or column. The aligned text boxes are considered to be table candidates. Figure 3.4 shows six *text boxes* (a,b,c,d,e,f), each one displaying their thresholds as vertical or horizontal black lines. Boxes whose distance is within the threshold limits are grouped either vertically or horizontally; forming in this way a preliminary structural organization of the table containing two columns and three rows.

3.2.3 Table Extraction

The table extraction module of *TAO* refers to recognizing the actual tables and locating each cell that belong to a particular table. I locate cells using the intersection between a row and a column, and store separately the tables found on each page of the document. The text contained in each cell of the table, if present, is extracted and saved in a semantic data structure, including additional information that describes the text in a cell, such as text font and size. The table extraction module is divided in two components: table recognition and table composition.

Table Recognition

In this component, I use the XML output produced by PDFMiner and the table candidates obtained from the Table Detection module. PDFMiner produces an output divided in two main sections: the body and the layout (see Figure 3.3). For each page on a document, the body section contains the information organized into a hierarchy of *text boxes*; *Text boxes* contain *text lines*, and *text lines* contain *text*, which in turn represent each single character and its coordinates in the original PDF document.

In order to extract words located in cells of the table I follow a two-phase process;

first I identify cells and reconstruct the text within each cell. To detect cells in the table, I use *text lines*. The logic behind this refinement is that *text boxes* provide us with a coarse representation of possible table layouts, while *text lines* provide a more accurate way to detect specific table elements, such as words in the table cells. Thus, I also calculate a distance matrix M_{d2} storing the differences between pairs of *text lines*.

Because *text lines* contain the feature *font size*, I use it to get the threshold of the maximum alignment on a column. To get this threshold, I select training data from 300 different tables, using the feature font size. The maximum left alignment of text in a cell, on the same column, is used as the target value. Similarly, I train data to find the maximum alignment between rows of tables.

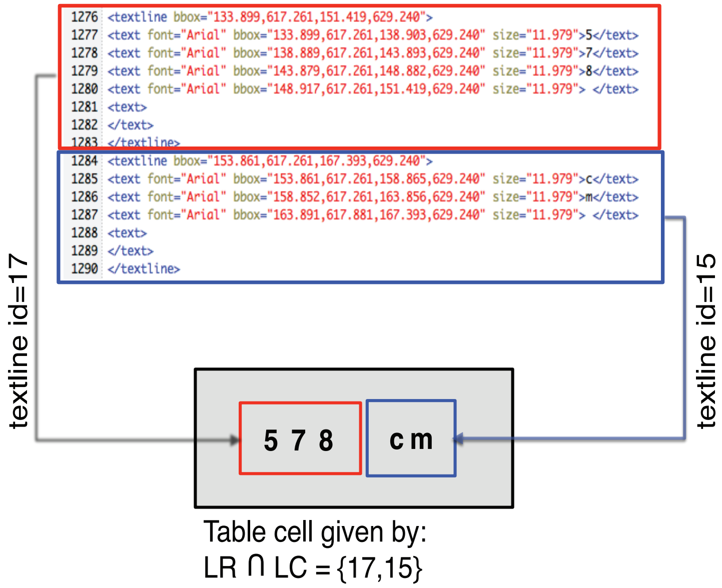
To take advantage of the characteristics of tables' composition, I select among Machine Learning approaches, a pattern recognition method. First, I compare the locally weighted regression (LWR) and k-nearest neighbor (K-NN) regression methods using the prepared training data to find the thresholds for maximum alignment among columns and rows. The K-NN regression method produces better results than LWR for these settings. The K-NN regression performs satisfactorily with four neighbors, an Eucledian distance, and a uniform weight approach. Then, I use K-NN regression to find the threshold t for maximum alignment to detect columns. In general the separation between two rows is not larger than the font size. Therefore, for rows it is more feasible to use the font size as a threshold due to the minimal variation of alignment for rows on tables.

Using the thresholds t and $fontsize$, the method identifies the *id*s of *text lines* aligned in the same column or row. In particular, this approach determines that, for a given pair of *text lines* (p, q) in M_{d2}, if $p.x_1 - q.x_2 < t$ and $p.x_2 - q.x_2 < t$ then (p, q) are in the same column, and their *id*s compose the list of columns LC. Similarly, if $p.y_1 - q.y_1 < fontsize$ and $p.y_2 - q.y_2 < fontsize$ then (p, q) are in the same row, and their *id*s compose the list of rows LR.

To reconstruct text within cells, or find words, the list of columns LC and the list of rows LR, obtained from the previous step, help identify cells. A cell (r, c) in the table is defined as the intersection of the *text line* identifiers found in a particular row r and a particular column c, which is represented as $cell[r, c] \leftarrow LR(r) \cap LC(c)$. This process produces a file containing all the cells, and its associated *text line* identifiers in a table grouped by column and row. The *text lines* contain *text* that represent particular characters in the document. Thus, using the grouped *text lines* and their associated *text* tags, this method can extract all the words in a particular cell.

Text extraction proceeds as follows: for each *text line* in a particular cell, this process identifies its children *text* tags and extract a) the particular character, b) text box id, c) text line id, d) *bbox* coordinates, e) text font, and f) text size (see line 3125 in Figure 3.3). PDFMiner may generate *text line* identifiers unsorted. For example, finding *text line id*s 15 and 17 in a particular cell does not mean that coordinates of 15 precede coordinates of 17 in the original Cartesian plane. Therefore, the order of identifiers does not respect the position of text in a table. I solve this issue organizing table elements with their coordinates instead of their *id*s. Figure 3.5 shows this text reconstruction procedure for a cell with two text lines, the first one has id=17 and contains four *text* tags with three numerical values and one space, while the second, with id=15 contains three *text* tags, two of them are characters and one is a space. This process is able to produce a sorted and aligned text fragment for the table cell, which is composed with the text line elements on a row and a column ($LR \cap LC$).

I keep cells grouped by row and preserve the number of column, coordinates and other attributes for each cell. If appropriate, these elements allow us to rebuild a table as found in its source document. Finally, the content of cells identified in each table within a document are organized in a file.



```
<textline bbox="133.899,617.261,151.419,629.240">
<text font="Arial" bbox="133.899,617.261,138.903,629.240" size="11.979">5</text>
<text font="Arial" bbox="138.889,617.261,143.893,629.240" size="11.979">7</text>
<text font="Arial" bbox="143.879,617.261,148.882,629.240" size="11.979">8</text>
<text font="Arial" bbox="148.917,617.261,151.419,629.240" size="11.979"> </text>
<text>
</text>
</textline>
<textline bbox="153.861,617.261,167.393,629.240">
<text font="Arial" bbox="153.861,617.261,158.865,629.240" size="11.979">c</text>
<text font="Arial" bbox="158.852,617.261,163.856,629.240" size="11.979">m</text>
<text font="Arial" bbox="163.891,617.881,167.393,629.240" size="11.979"> </text>
<text>
</text>
</textline>
```

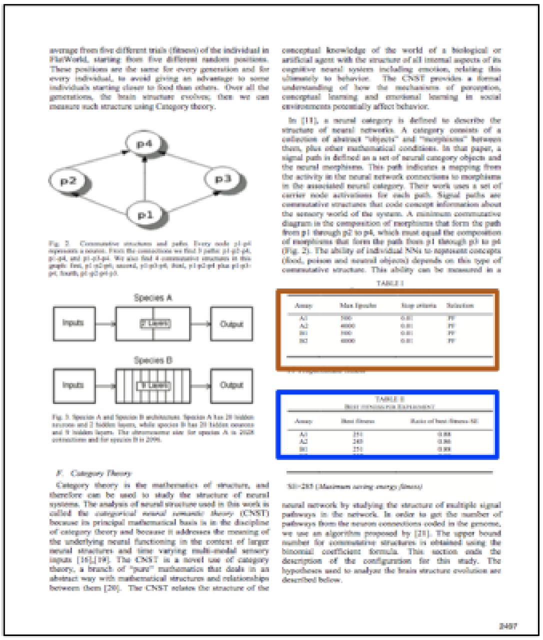
Figure 3.5: Text Reconstruction for Table Cells

Table Composition and Functional Analysis

The output of this component is a file describing tables found on each page of a document. Up to this point, multiple tables that appear on the same page are identified as a unique table. To further refine table separation, I perform Table Composition.

Table Composition deals with separating the tables in a document, it also classifies tables cells as **header** or **data**. A header refers to a label that groups table information (i.e., metadata of rows or columns), and data refers to the actual content or body of a table.

The composition process retrieves all cells in a particular page. It sorts coordinates of text cells grouped by row and calculates row separation. At times, there exist more than one table on a page and they need to be separated (see Figure 3.6). To calculate row separation between two tables, I previously selected training data from 300 different documents with two consecutive tables, using the font size as the main feature, and the row separation between tables as a target.



average from five different trials (fitness) of the individual in FlatWorld, starting from five different random positions. These positions are the same for every generation and for every individual, to avoid giving an advantage to some individuals starting closer to food than others. Over all the generations, the brain structure evolves; then we can measure such structure using Category theory.

Fig. 2. Commutative structures and paths. Every node p1-p4 represents a neuron. From the connections we find 3 paths: p1-p2-p4, p1-p4, and p1-p3-p4. We also find 4 commutative structures in this graph: first, p1-p2-p4, second, p1-p3-p4, third, p1-p2-p4 plus p1-p3-p4, fourth, p1-p2-p4-p3.

Fig. 3. Species A and Species B architecture. Species A has 20 hidden neurons and 2 hidden layers, while species B has 20 hidden neurons and 9 hidden layers. The chromosome size for species A is 2028 connections and for species B is 2996.

F. Category Theory

Category theory is the mathematics of structure, and therefore can be used to study the structure of neural systems. The analysis of neural structure used in this work is called the *categorical neural semantic theory* (CNST) because its principal mathematical basis is in the discipline of category theory and because it addresses the meaning of the underlying neural functioning in the context of larger neural structures and time varying multi-modal sensory inputs [16],[19]. The CNST is a novel use of category theory, a branch of "pure" mathematics that deals in an abstract way with mathematical structures and relationships between them [20]. The CNST relates the structure of the conceptual knowledge of the world of a biological or artificial agent with the structure of all internal aspects of its cognitive neural system including emotion, relating this ultimately to behavior. The CNST provides a formal understanding of how the mechanisms of perception, conceptual learning and emotional learning in social environments potentially affect behavior.

In [11], a neural category is defined to describe the structure of neural networks. A category consists of a collection of abstract "objects" and "morphisms" between them, plus other mathematical conditions. In that paper, a signal path is defined as a set of neural category objects and the neural morphisms. This path indicates a mapping from the activity in the neural network connections to morphisms in the associated neural category. Their work uses a set of carrier node activations for each path. Signal paths are commutative structures that code concept information about the sensory world of the system. A minimum commutative diagram is the composition of morphisms that form the path from p1 through p2 to p4, which must equal the composition of morphisms that form the path from p1 through p3 to p4 (Fig. 2). The ability of individual NNs to represent concepts (food, poison and neutral objects) depends on this type of commutative structure. This ability can be measured in a neural network by studying the structure of multiple signal pathways in the network. In order to get the number of pathways from the neuron connections coded in the genome, we use an algorithm proposed by [21]. The upper bound number for commutative structures is obtained using the binomial coefficient formula. This section ends the description of the configuration for this study. The hypotheses used to analyze the brain structure evolution are described below.

TABLE I

Array	Max Epochs	Stop criteria	Selection
A1	500	0.01	PF
A2	4000	0.01	PF
B1	500	0.01	PF
B2	4000	0.01	PF

PF: Proportionate fitness

TABLE II
BEST FITNESS FOR EXPERIMENT

Array	Best fitness	Ratio of best fitness SE
A1	251	0.88
A2	240	0.86
B1	251	0.88

SE=285 (*Maximum saving energy fitness*)

2497

Figure 3.6: Separation of Two Consecutive Tables

I use this training data with the K-NN regression method to find the separation threshold t_{table} between tables. Again, the pattern recognition K-NN regression method finds the separation between tables. In particular, the K-NN regression uses the following settings: four neighbors, Eucledian distance, and uniform weight. If the distance between two rows is greater than t_{table}, this methos classifies and assigns an identifier for a new table.

Once each table is a separate entity, cells undergo a functional analysis to classify them into data or header, the *font style* of the text in the cell serves for this purpose. In particular, bold font styles are an evident way to classify text as headers. Another alternative is to use *font size*, where larger font sizes are assigned to metadata and smaller ones to data. This heuristic is likely to fail for customized documents, but is general enough to capture relevant cell differences in research publications. Thus, following this heuristic, a cell is classified as a header (i.e., cell that groups some cells under it). For instance, given two table cells $[text : Age, font : ArialBold]$ and $[text : 15\ years, font : Arial]$, the text *Age* represents a header and *15 years* represents data. Using pattern matching, I also classify the cell content by data type. For simplicity, I classify numeric or string of characters. In this way, I recover information from a table and its metadata for each cell (i.e., content, column number, coordinates, font, size, data type, header or data label). TAO yields an annotated document in JSON format.

I extend TAO [29] to identify tables from non-PDF documents. To extract tables content from well-formed XML documents, I use the tags `<table-wrap>` and `<table>` that indicate the existence of tables. I use Xpath [90] to detect tags and find a table within a document. Then, the table cells are organized by rows in JSON. In addition, I determine if a cell is a header or data using the tags `<thead>` and `<tbody>`. For simplicity, I detect string and numeric data types using regular expressions.

Finally, the main elements of table extraction are included in a file: number of page where a table was found, and tables content. The tables content includes extracted text of table cells grouped by row. For further analysis, I also store provenance information associated with the cells, that is, identifiers of *text boxes* and *text lines* from where the text was identified, as well as particular text attributes including: coordinates of the text, text size, text font, and data type. To complement TAO, the original column of each cell is included.

Figure 3.7 shows a summarized view of applying TAO's methods to an example of a digital publication [91]; detecting a table embedded in a document, extracting information from table cells, and organizing cell's content and functions into a file. This organization also describes that the particular table is the first one in page one. Each element in the representation provided by TAO contains identifiers of their

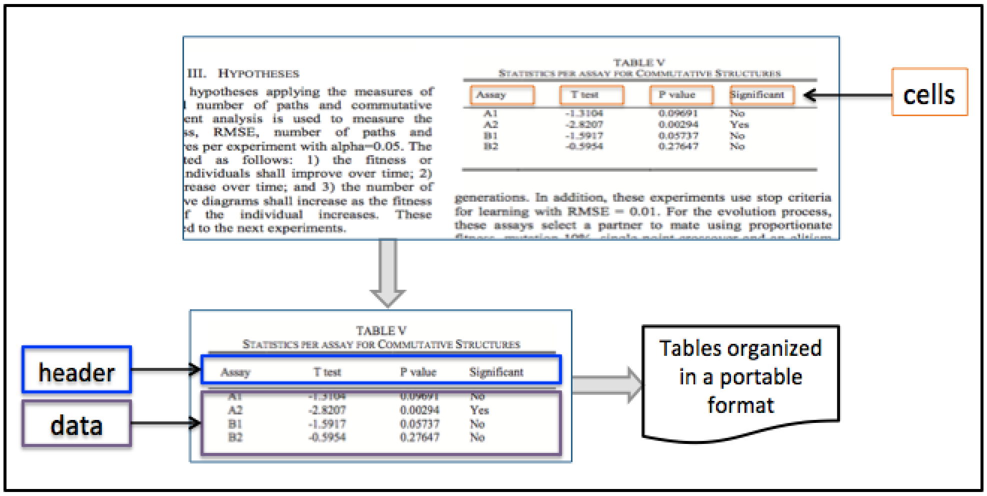


TABLE V
STATISTICS PER ASSAY FOR COMMUTATIVE STRUCTURES
cells
header
data
Tables organized in a portable format

Figure 3.7: TAO's Methods Applied to a Digital Publication

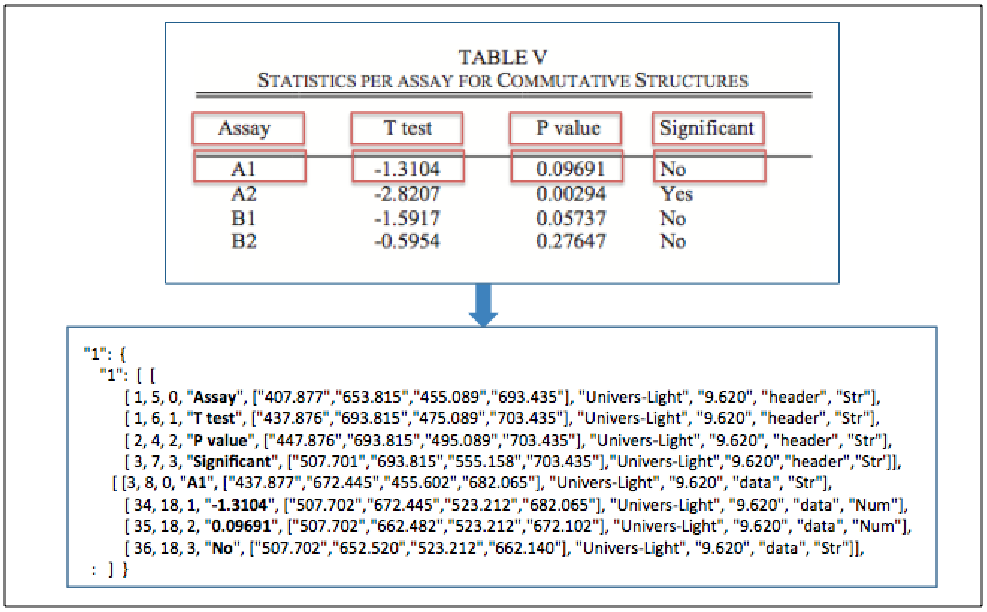


TABLE V
STATISTICS PER ASSAY FOR COMMUTATIVE STRUCTURES

Assay	T test	P value	Significant
A1	-1.3104	0.09691	No
A2	-2.8207	0.00294	Yes
B1	-1.5917	0.05737	No
B2	-0.5954	0.27647	No

```
"1": {
  "1": [ [
      [ 1, 5, 0, "Assay", ["407.877","653.815","455.089","693.435"], "Univers-Light", "9.620", "header", "Str"],
      [ 1, 6, 1, "T test", ["437.876","693.815","475.089","703.435"], "Univers-Light", "9.620", "header", "Str"],
      [ 2, 4, 2, "P value", ["447.876","693.815","495.089","703.435"], "Univers-Light", "9.620", "header", "Str"],
      [ 3, 7, 3, "Significant", ["507.701","693.815","555.158","703.435"],"Univers-Light","9.620","header","Str"]],
    [ [3, 8, 0, "A1", ["437.877","672.445","455.602","682.065"], "Univers-Light", "9.620", "data", "Str"],
      [ 34, 18, 1, "-1.3104", ["507.702","672.445","523.212","682.065"], "Univers-Light", "9.620", "data", "Num"],
      [ 35, 18, 2, "0.09691", ["507.702","662.482","523.212","672.102"], "Univers-Light", "9.620", "data", "Num"],
      [ 36, 18, 3, "No", ["507.702","652.520","523.212","662.140"], "Univers-Light", "9.620", "data", "Str"]],
 :  ] }
```

Figure 3.8: Results from TAble Organization

text box, text line, column, as well as information such as coordinates, font size, font style, cell function, and data type. All this information can be used to reconstruct a table if needed. Figure 3.8 shows the actual output of the first two rows of the table extracted from this example.

3.3 Evaluation of Table Identification

To evaluate the usefulness and generality of the TAO compared to other PDF table extractors, such as PDF-TREX and TableSeer (see Chapter 2, Section 2.1 Table Analysis), I perform a systematic assessment of its capability to detect and recognize tables. Detecting tables means identifying all the tabular elements in the PDF documents, without considering if their content was reconstructed accurately. Thus, I compute *Precision* as the fraction of the correct number of tables identified divided by the total number of tables found. *Recall* as the fraction of the correct tables found, divided by the total number of tables in the documents. Also, the F1-measure, which is an average that combines precision and recall. Recognizing tables means correctly identifying and associating cells and their content to a particular table. For the recognition task, **precision** is the fraction of the correct table cells found divided by the total number of table cells found in a table and **recall** is the fraction of the correct number of table cells found divided by the total number of actual table cells in a table.

3.3.1 Baseline and Datasets of Table Identification

To compare and contrast the performance and generality of TAO, I use two other automatic table extractors as the baseline: PDF-TREX and TableSeer (see Section 2.1).

PDF-TREX [43] is an heuristic-driven approach for table recognition and extraction from PDF documents. PDF-TREX code is not publicly available, thus, I report its results published in [43]. TableSeer [38] detects tables from documents, extracts tables metadata, and indexes and ranks tables. TableSeer's code is publicly available and can be downloaded from `http://tableseer.soft112.com`. Thus, all the experiments produce TableSeer outputs.

The first dataset *TREX* is publicly available and extensively documented in [43]. E. Oro and M. Ruffolo created it and can be downloaded from `http://staff.icar.cnr.it/ruffolo/pdftrex/dataset.zip`. This dataset intends to be a standard benchmark for automatic table extraction. *TREX* consists of one hundred PDF single-column documents written in English and Italian languages, with 148 tables. These documents do not reflect the format of standard research papers, but rather technical reports containing tables with a variety of formats and styles. I use this dataset to compare the performance and generality against PDF-TREX and TableSeer. Generality refers to the capability of the system to accurately extract tables from a collection of heterogeneous documents that do not adhere to the rigid style norms found in research papers.

The second dataset is obtained from the Cornell University Library site `http://arxiv.org/`. It is composed of twenty scientific publications selected randomly. This dataset *CORNELL* includes one- and two-column documents in English with 79 tables. In addition, I use the *COMBINED* dataset, which results from merging the TREX and CORNELL datasets to generate a more diverse one.

3.3.2 Results and Discussion of Table Identification

I compare TAO's precision and recall for table detection and table recognition with a baseline containing the two best automatic table extractors currently available in the literature: PDF-TREX and TableSeer. The experiments use the three datasets described in the previous section.

In the first set of experiments I compare TAO, PDF-trex, and TableSeer using the

Table 3.1: Experiments Using General Reports

Table Detection			
Method	***Precision***	***Recall***	***F1 measure***
TAO	0.913	0.925	0.919
PDF-TREX	0.862	0.984	0.917
TableSeer	1.0	0.121	0.216
Table Recognition			
Method	***Precision***	***Recall***	***F1 measure***
TAO	0.899	0.835	0.864
PDF-TREX	0.753	0.965	0.846
TableSeer	0.005	0.005	0.005

TREX dataset. I compute precision, recall, and F1 measure for *Table Detection* and for *Table Recognition*. Table 3.1 summarizes the results for the two tasks and three approaches. From the first set of experiments, TAO yields an average F1-measure of 91.9 percent for Table Detection and of 86.4 percent for Table Recognition using the *TREX* dataset. Compared to PDF-TREX, TAO performs similarly; however, TAO performs better than TableSeer. TableSeer performs poorly for the experiments using *TREX* dataset due to the fact that documents in this dataset do not adhere to the rigorous formatting standards common for scientific publications in the literature.

TAO has no issues detecting and extracting tables in another language because tables are composed of rows and columns and the language is irrelevant. Therefore, TAO's structural approach detects tables in Italian and English languages. The language of the table will pose a challenge for systems based on syntactic analysis of table cells. The main TAO's limitation includes its reliance on PDFMiner, which cannot extract information from a PDF file when the document is password protected or there exists a malformation in the creation of the document. Even though PDFMiner only converts 88 PDF documents to XML out of 100, TAO is still able to detect 137 tables out of the 148 and also identifies a total of 12,504 correct table cells out of 14,466.

The second experiment uses the *CORNELL* dataset. As mentioned above, this dataset is composed by scientific publications with one and two columns and a total of 79 tables. Table 3.2 reports a summary for table detection and table recognition applying the two methods TAO and TableSeer. For this experiment and the next one, I could not obtain results using PDF-TREX because this tool is not publicly available.

Table 3.2: Experiments Using Scientific Documents

Table Detection			
Method	***Precision***	***Recall***	***F1 measure***
TAO	0.893	0.848	0.813
PDF-TREX	NA	NA	NA
TableSeer	0.925	0.784	0.849
Table Recognition			
Method	***Precision***	***Recall***	***F1 measure***
TAO	0.869	0.956	0.919
PDF-TREX	NA	NA	NA
TableSeer	0.866	0.925	0.895

In the case of the *CORNELL* dataset, TableSeer performs slightly better than TAO for Table Recognition. One of the limitations in TAO using the *CORNELL* dataset is that the system produces more false positives than TableSeer. In particular, these false positives are associated to Figures and Equations embedded in the document. To solve this issue, the method needs to detect when the rows do not contain table cells, but other elements.

Finally, for the sake of completeness and to quantitatively assess the generality of the method compared to TableSeer, I report aggregated experiments using the *COMBINED* dataset. Table 3.3 reports an average for table detection and recognition for TAO and TableSeer.

Using the *COMBINED* dataset, TAO yields an average F1-measure of 89.5 percent for Table Detection and of 88.9 percent for Recognition, while TableSeer pro-

Table 3.3: Experiments Using General and Scientific Documents

Table Detection			
Method	***Precision***	***Recall***	***F1 measure***
TAO	0.903	0.886	0.895
PDF-TREX	NA	NA	NA
TableSeer	0.962	0.453	0.616
Table Recognition			
Method	***Precision***	***Recall***	***F1 measure***
TAO	0.884	0.894	0.889
PDF-TREX	NA	NA	NA
TableSeer	0.436	0.465	0.450

duces an average F1-measure of 61.6 percent and 45 percent respectively. Compared to TableSeer, TAO is more robust and general because TableSeer depends on particular keywords to perform accurate Table Detection. In this experiment TAO recognizes and extracts a total of 17,242 correct table cells, while TableSeer recognizes 6,389 table cells. Similarly, TAO is more general and robust than PDF-TREX because TAO can handle double-column documents and PDF-TREX only works for one-column as documented in [43].

TAO was evaluated on a variety of PDF documents with more than 225 tables, and more than 400 pages. Its performance was compared to a baseline from related work. TAO allowed us to perform an automatic detection and organization of tables in PDF documents, including scientific publications and other less rigorous types of documents. This process overcame related work limitations and performed satisfactorily on PDF documents with large and small number of pages, with single and double columns, and with various tabular formats.

Chapter 4

Table Interpretation to Synthesize Documents

The large production of scientific publications hinders the rapid analysis of their content. Therefore, I propose to perform automatic analyses of tables and text within digital publications to find specific facts.

Table interpretation refers to understanding information contained in tables. The main challenges to understand a table include finding an appropriate context of its information, finding semantic relationship between a table's content and the text of its publication, and representing a table's interpretation.

4.1 Understanding Tables and Documents

Practices to include semantic annotations and unified vocabulary are difficult to implement, and a single standard to publish scientific work is not feasible. Therefore, I develop a framework with a comprehensive method to analyze relevant information from publications, regardless the lack of standards in some scientific disciplines.

Table interpretation is the process of finding the meaning of a table's content. To perform manually this interpretation, a researcher needs to know the context of the information before understanding the meaning embedded in a table (i.e., the meaning that the table's author wants to transmit). Table interpretation is a demanding task when a researcher has no previous knowledge of a table's content and context. In addition to the large number of tables and documents to analyze, the main challenge to perform automated table interpretation is to identify the context of a table and its document.

To understand tables contained in a publication, researchers need to find the concepts, that is, entities contained in a table's cells. This task can be challenging because each concept can have a specific context. I use *semantic annotations*, which are pointers to [ontology or dictionary] definitions, Web pages, or documents containing more detailed information to understand concepts. In addition, semantic relationships exist between these concepts and the text of a table's publication. The semantic annotations and relationships are explicit elements to understand a table and its publication.

This framework takes advantage of the structured information in tables and the unstructured text in scientific publications. The digital publications undergo text analysis to identify its metadata (i.e., provenance information and context), and a table recognition process to identify tables' content. To understand a table's content, this approach uses Natural Language Processing; and to disambiguate entities, it uses a publication's context, a general ontology, and a method using the Latent Semantic Indexing with context. The recognized entities from tables guide the search of semantic relationships in text. To perform this search, the approach uses an unsupervised method, which determines the relationships importance with a classifier. Also, structural relationships are recovered from tables. Finally, the findings are used as a short summary, that is, a synthesis of a publication.

Therefore, my main contributions include a comprehensive framework to automatically analyze documents by means of identifying important entities, extracting structural relationships from tables, and extracting semantic relationships between entities. In addition, a method to integrate information within a publication. To conclude, the results show that the semantic relationships extracted from tables and text provide highly ranked relationships to identify valuable information and analyze a publication promptly.

Similarly to using the extracted information to analyze a publication promptly, this work can potentially be used to search queries, retrieve information, and to reproduce scientific research if appropriate. This work assists not only to understand tables from digital documents, but also to manage table's information and to integrate elements of digital documents, such as tables and text, into semantic relationships.

For the remainder of this chapter, Section 4.2 contains an overview of the Semantic Web and Sources of Knowledge. Section 4.3 contains the approach to interpret tables and extract semantic relationships to understand digital publications. Finally, Section 4.4 contains a set of experiments, results and discussion of the evaluation and findings.

4.2 The Semantic Web and Sources of Knowledge

The Semantic Web intends to present meaningful information where people and computers can work in cooperation, having a global Web with machine-readable data [28]. The Semantic Web principles [92] that help using the Web more efficiently and consulting interlinked information follow:

1. Machine-readable
2. Interoperable
3. Reusable
4. Flexible
5. Ubiquitous

Although the Semantic Web principles are intrinsic to the Web, other digital documents can benefit from these principles. For example, documents in PDF format are commonly found on the Internet. Although this type of documents can work independently of the Web, PDF documents present restrictions to navigate to other documents seamlessly [14]. Therefore, this type of documents cannot directly integrate to the Semantic Web functions. I propose a method to integrate digital publications in this format to the Semantic Web. To do so, I use a machine-readable format to synthetize relevant information from publications and sources of knowledge following the Semantic Web principles.

There exist structured and unstructured knowledge sources that provide meaning to data [93]. The most common structural knowledge sources are thesaurus, taxonomies, and ontologies. Other structural knowledge sources include dictionaries, concepts, associations, and glossaries. For instance, the Oxford dictionary [94] and WordNet. The latter is an English lexical database containing sets of synonyms to represent concepts [95].

An ontology is a set of precisely defined terms about a specific domain. In general, an ontology is accepted by its domain's community [96]. Moreover, ontologies are widely used to provide a strong structure to represent meaning, relationships, and inference rules. Ontologies provide a semantic description of the objects and their relationships and are built to share knowledge [97]. Therefore, they are appropriate to build interoperability. Ontologies have proved to give form and structure to knowledge.

Different groups in various fields work to control ontologies. By 2007, the Open Biomedical Ontologies (OBO) consortium contained sixty ontologies [98]. A special type of ontology represents relationships, such as the OBO Relation Ontology [99] and nominal relations [100]. The relational ontology defines relationships, such as *occurs in* and *is part of*. R. Girju and her colleagues [100] use seven defined relations: cause-effect, instrument-agency, product-producer, origin-entity, theme-tool, part-whole, content-container to evaluate their methods for classification of semantic relationships between nominal entities.

More recently, the Semanticscience Integrated Ontology (SIO) [75] defines relationships for the Bioinformatics field. SIO contains relationships from different ontologies, such as the OBO Relational Ontology. Therefore, SIO contains a more complete set of definitions to represent semantic relationships. In addition, SIO's relationships represent a hierarchical structure. SIO uses the Ontology Web Language (OWL) and is available at `http://semanticscience.org/ontology/sio.owl`.

Even though SIO was developed for Bioinformatics, several of its relationships are generic and useful to represent relationships from other areas. For this work, I select the most general definitions of relationships from SIO to represent relationships derived from tables and text within publications of different areas. To better control these definitions, I categorize them into four sections 1) classification, 2) association, 3) aggregation, and 4) generalization (see Appendix B).

Another recent effort in knowledge sources is the general vocabulary schema.org [76]. This vocabulary derives from the Resource Description Framework (RDF) data modeling. The schema.org represents a data model with different hierarchical types, containing subclasses and properties. Using this general vocabulary allows representing information of publications from different domains. This vocabulary is preferred for this framework to represent information to identify a publication because of its common language and classification. In particular, I use it to represent the context of publications as keywords, and the basic information describing a publication, that is, metadata including creator and headline.

On the other hand, Wikipedia in English contains more than five million of documents to date [101]. Wikipedia presents organized information and its community keeps maintaining its content. DBpedia is a general ontology generated from Wikipedia, which contains curated information and follows the Semantic Web principles, such as machine-readable and reusable. DBpedia contains a set of entities with a type and properties with their respective values. For instance, the entity `"Mexico"` has a type *populated place*, and contains the property *capital* with the value *Mexico City*. This framework uses Wikipedia, DBpedia, the Oxford dictionary, the Internet, and the academic search Engine BASE as external knowledge sources to enrich and disambiguate entities for table interpretation.

In addition to the previously mentioned sources of knowledge, I use internal knowledge sources within a publication: tables, context, and text from digital documents. These sources of knowledge enable us to interpret tables and understand documents using functional and semantic analyses.

4.3 Table Interpretation

I present a comprehensive framework to understand tables. Figure 4.1 highlights the processes described in this section for this framework, including **Discovery of Metadata and Context, Categorical Entities, and Relationships**. These elements serve to synthesize a digital document.

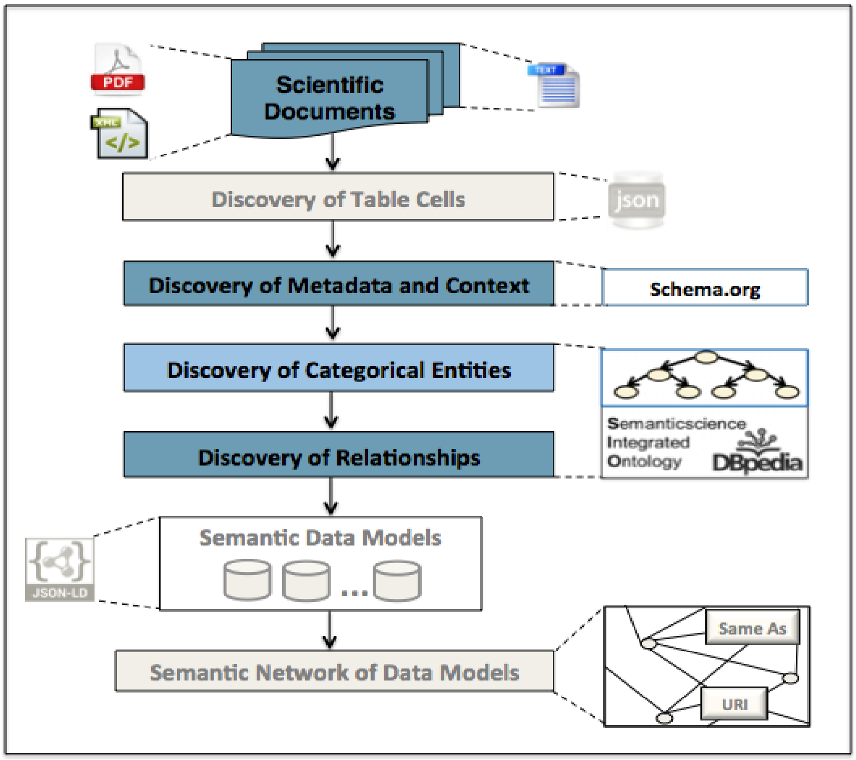

PDF
XML
Scientific Documents
TEXT
Discovery of Table Cells
json
Discovery of Metadata and Context
Schema.org
Discovery of Categorical Entities
Semanticscience Integrated Ontology
DBpedia
Discovery of Relationships
JSON-LD
Semantic Data Models
...
Semantic Network of Data Models
Same As
URI

Figure 4.1: The Table Interpretation Process

M. Gobel et al. define table interpretation as rediscovering the meaning of the tabular structure, using functional analysis and semantic interpretation [45]. My work is based on this definition and extends it to find important content from a publication that can be used as a synthesis.

Recent graphical features allow users to design colorful and sophisticated tables. Even though tables' format and style have evolved through the years, still the basic functional elements of a table are headers and data. X. Wang's doctoral work defines areas and elements of tables in more detail [32]. To simplify this study, I focus on identifying headers and data. A header in a table is a class representing a column or row of cells, and the cells under a header represent data.

For PDF documents, I perform a functional analysis of table's content using TAO [29] as explained in the previous Chapter 3, Section 3.2. After recognizing the functionality of each table cell, I recognize its data type. Besides extracting table information from PDF documents, I use table tags to identify data and metadata of tables embedded in other non-PDF documents, such as XML. Independently of a table's source document format, I use regular expression methods to identify data type of each cell and tags for cell functionality.

After the functional analysis, I perform a semantic analysis to understand a table's content, to relate table's concepts to a source of knowledge (i.e., ontology), to relate these concepts to the text of a table's publication, and to extract and represent semantic relationships.

In particular, Figure 4.1 shows the three main steps to perform semantic analysis. The tasks contained in this analysis include the discovery of a) a table's source document context and metadata to identify a publication, such as keywords, author, and title; b) entities contained in table cells; c) entities' disambiguation and annotation; d) relationships between entities within a table; and e) relationships between a table's entities to text in a digital document.

In summary, to provide a full understanding of a table's content, this method preserves metadata of publications, uses semantic annotations for entities in each table, discovers semantic relationships between entities, and represents these findings as a synthesis of each publication.

4.3.1 Discovery of Metadata and Context

Metadata represents the main information to identify a publication, such as name and author. *Context* represents the topic embedded in the text of a publication. In this section, I present the importance of metadata and context, as well as the approach to recognize these elements in a digital publication.

Metadata

Metadata information is important for resource discovery, electronic resources organization, interoperability, digital identification, archive and preservation [102]. While metadata describes relevant information to find the provenance of a publication, the lack of metadata in digital publications obstructs the dissemination of scientific work. In addition, having a publication's provenance allows identifying possible fake information on digital documents. L. Moreau discusses the importance of provenance in the Web, as well as problems associated with it [103].

Metadata standards are developed by organizations such as the International Organization for Standardization, American National Standards Institute, and World Wide Web Consortium. There exist several standards for metadata. For instance, the Dublin Core Metadata Element is a standard describing Web-based documents [104], such as title, creator, and subject. Another metadata standard is the Learning Object Metadata standard (IEEE 1484.12.1-2002) to allow us reusing learning resources, such as computer-based training and distance learning [102]. Besides the differences between metadata standards, every organization can use different format to represent a metadata standard, for instance using formats such as the Standard Generalized Mark-up Language or XML, and even using a database to represent metadata attributes [12].

For this work, I use the schema.org vocabulary [76]. In particular, I use specific properties of the scholarly article object, such as creator, headline, and keywords. The schema.org controlled vocabulary helps represent scientific documents and facilitate data collaboration and sharing. This vocabulary allows representing information from different fields with minimum dependency on ontologies.

The method to recognize metadata in digital publications searches directly for tags describing the metadata elements in documents, including PDF format. These tags include `<author>`, `<article-title>` and `<keywords>`. If not found, this framework accepts a digital document in plain text or PDF format. If the digital document is in PDF, it is converted to text using PDFMiner [41]. When a document lacks tags, pattern matching searches the first page of the publication to extract this information. Whenever the keywords are not defined in the metadata of a document, another process searches for them to represent the context of a publication.

Context

Digital publications contain unstructured data (i.e., prose), to describe the topic reported. The free text provides important concepts relevant to the publication, which can help to find the context. As part of its metadata, some publications already define keywords to describe the subject of a document.

To find the keywords in a publication, the text of a publication undergoes a preprocessing step, where I eliminate stop words, and small and large length words. Then, the text is converted to n-grams to get their weights. For simplicity, I use 1-grams that represent words.

I apply the Term Frequency Inverse Document Frequency (TFIDF) method (see Equation 4.1) to find the weights of the most relevant concepts in a publication. I use the TFIDF method because it is widely used to detect important concepts that represent the content of documents and even the context. It can also be used to perform queries [105].

$$TFIDF = TF \times IDF \tag{4.1}$$

This method works on the premises that term frequency (i.e., TF) is important for relevant words, but it is not the only factor to determine the words relevance. Then, it uses the inverse document frequency (i.e., IDF), which is the logarithm of the total number of documents in the collection of documents to analyze, divided by the number of documents where the word to analyze appears (see Equation 4.2).

$$IDF = log\left(\frac{N}{n_t}\right) \tag{4.2}$$

Most times, the TFIDF method uses a fixed collection of documents to generate the weights of the keywords. Instead, this method uses the Internet as the collection of documents. In particular, Wikipedia contains more than five million of documents to date [101]. Therefore, N is five million of documents and n_t is the number of documents in the collection containing this term.

From this discovery process, I select the first five keywords with the highest weights. these keywords are preserved as a list of concepts, which become part of the metadata of a publication. The integration of these concepts enables us representing and using better the subject of publications.

4.3.2 Entity Recognition, Annotation, and Disambiguation

For entity recognition, I first find the entities on each table. Later, the entities guide the search of semantic relationships. To recognize entities, I need the output from the functional analysis performed previously (see Section 3.2.3 Table Composition and Functional Analysis). This analysis detected the functionality of each table cell, such as a header cell to indicate a column and cells containing data.

The functional analysis of tables embedded in PDF documents from TAO [29] produces a JSON document with the table cells grouped by row. Moreover, this functional analysis produces a similar JSON representation with table cells organized by rows from XML documents. The functional analysis' output facilitates searching for entities in table cells. Because text cells may contain ambiguous entities, this framework recognizes and disambiguates such entities.

Entity Recognition and Annotation

To find entities in table cells, each cell with data type string undergoes a Natural Language Processing that includes a) noun phrase analysis, b) tokenization, and c) part of speech (POS) tagging.

First, each text cell is tokenized and tagged using the Python Natural Language ToolKit [106] and a defined model. Second, the tagging assists to create a tree of nouns. This tree is parsed to find singular or plural nouns that represent continuous entities. Third, if no entities are found with this process, the TextBlob tool [107] is applied for noun phrase analysis to recover entities.

I develop a model to ensure that tagging is unambiguous, that is when a word can be used in different parts of speech or is not tagged correctly. For instance, the word “present” can be used as a noun or verb. The model learns from sentences how to detect nouns depending on the use they have in sentences. For instance, the sentences *I receive a present* and *I present a project* help identifying the first usage as a noun and the second as a verb.

After finding an entity on each string cell, this entity is searched on DBpedia. To perform this search, I use DBpedia's naming convention. For each entity, an entity's name is converted using the first capital letter and joining words with a dash. For instance, the entity “diabetes management” converts to “Diabetes_management.” In addition, DBpedia redirects searches that refer to the same concept. For instance, searching for “Glycemic_control” or “Diabetes_treatment”, I find the resource for "Diabetes_management" because both concepts appear in the property `"wikiPageRedirects"` of this entity.

DBpedia offers a resource for each entity that is a Web page containing a description, a type, and properties of a particular entity. For instance, the entity Diabetes corresponds to the Web page `http://dbpedia.org/page/Diabetes_mellitus`, containing the entity "Diabetes_mellitus" with type disease and several properties, such as the property abstract describing this entity. Therefore, this description relates to the annotation *"Is A"* for a specific entity. This hyperlink represents an additional annotation for further consultation of properties. Each hyperlink in DBpedia represents a Universal Resource Identifier (URI), which enriches an entity because it contains additional information related to it.

Although DBpedia is a useful structured ontology, it may not contain every entity's description. Furthermore, a concept may have more than one meaning, that is, an ambiguous concept. For instance, the concept *Race* can refer to a race of cars or a person's race. For these cases, DBpedia either shows 1) a generic description containing the words $(may \vee can) \wedge (mean \vee refer \vee stand)$ in the abstract property of an ambiguous entity or 2) a list of possible concepts under the property *wikiPageDisambiguates*. At this point, DBpedia shows at least forty different concepts that can be used for the entity *Race*. Whenever the process finds this property, the Oxford dictionary and the Internet serve as sources of knowledge to annotate and disambiguate those entities.

The next step is to find categorical entities, that is, unique entities. To do so, the unique entities detected in the ontology DBpedia [56] serve for this purpose. For each unique entity, a semantic annotation and its description are stored using the relationship *IsA* and URI. If an entity is ambiguous or it does not exist in DBpedia, this process searches for a Uniform Resource Locator (URL) to define or explain this entity.

Entity Disambiguation

A general disambiguation method to solve this problem is to use a dictionary to look up for the different meanings of the unresolved word, and consider the corpus of the data to get the correct meaning for an entity. Instead, to find a description for an ambiguous entity automatically, I search the Internet for a particular Web resource that contains an explanation of this entity considering the source document's context. The Web resource is identified as a URL, which may point to a Web page or to another document in any format, for example, PDF.

Whenever an entity does not exist in Dbpedia or it is ambiguous, I use the API Bing and a tailored Latent Semantic Indexing (LSI) analysis [57] to find the closest explanation to the entity at hand and its context. I select LSI because it performs well categorizing documents using only several hundred dimensions [108]. This entity disambiguation method is shown in [30].

In addition to the previous method, I research two different methods for entity disambiguation. In the first method, I tailor a query vector containing the sentences where the entity at hand was mentioned. To increase the weight in a publication's context, I increase the query vector with the context, title and abstract of a publication. The Web Search API from Microsoft Cognitive Services [109] (previously called Bing API) and the LSI analysis help finding an entity disambiguation and annotation. The main steps of this method follow:

1. Search for the ambiguous entity x
2. Create vector D with a set of the n documents retrieved, where $n = 100$
3. For each $d_i \in D$ where $1 \leq i \leq 300$, normalize vector with most relevant words
4. Create vector s, which contains the context, title and abstract of a publication

5. Create vector u, which contains the sentences where the entity x appears

6. Create query vector $q = s \cup u$

7. Normalize vector q

8. Apply latent semantic indexing

9. Select the most similar element of vector D to vector q

In this way, I obtain the Web page with an explanation of the entity at hand. The URL is stored as a non-formal annotation to represent the entity of interest, and it can point to a Web page or a digital document, for example, PDF. Note that differently from a URI, a URL may change over time.

The second method extends the first method because it includes more sources of knowledge than DBpedia, i.e., the Oxford dictionary and scientific publications retrieved from the search engine BASE. Then, after searching DBpedia, it searches the Oxford dictionary and finally it uses the BASE API instead of the Web Search API to determine a URL to explain an ambiguous entity. This method takes advantage of an established ontology, dictionary, and repository of scientific documents with reliable definitions of entities.

After entities and annotations are located in a table, I create a list of unique entities to avoid searching for the entities already detected. Once all the tables in a document are processed, this list contains all the categorical entities from a publication.

4.3.3 Discovery of Relationships

Generic relationships are powerful abstractions to represent semantics because they are high level templates for relating real world entities [110]. Entities are also known as classes (e.g., country, language, and person).

A semantic relationship is composed of a structural binary relationship that is domain independent. The definition of a semantic relationship is primarily based on concepts from [110]. A structural relationship represents static constrains and rules, (i.e., classification, generalization, grouping, aggregation). The semantic relationship is domain independent and because of its generality, it can be applied to different areas.

This framework uses binary relationships. A binary relationship (R) contains important semantics between a generic relationship and two arguments. Figure 4.2 shows the three components of a relationship and its cardinality. The values of cardinality indicate the minimum and maximum number of arguments to have in a relationship, as explained in [111]. The first cardinality (m,n) goes from Argument A to B, where m can take the values of [0,1] and n can take [1,N]. Similarly, the values of the cardinality (p,q) go from Argument B to A.

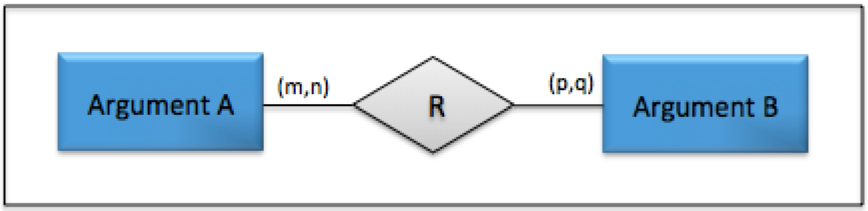


Figure 4.2: A Semantic Binary Relationship (R)

I divide the generic relationships from SIO into four categories: classification, association, generalization and aggregation. A brief description of these categories and their relationships follow. For a complete list of the SIO relationships used in this framework, refer to Appendix A.

1. Classification contains relationships that relate a class with a set of entities

sharing same properties. An instance cannot change its class. The relationships of this category include *has attribute*, *is attribute of*, *is member of*, and *has value*.

2. Association contains relationships that represent a structural connection among entities. The relationships of association include *is related to*, *is referred to by*, *is spatiotemporally related to*, *is located in*, *is connected to*, *refers to*, *is comparable to*, *has evidence*, *references*, *is referenced by*, *is implementation*, *has implementation*, *is causally related with*, *is causally related from*, and *has basis*.

3. Aggregation contains relationships that connect a whole to its components. The relationships of aggregation include *is source of*, *is part of*, *is proper part of*, *has input*, *is input in*, *is output of*, and *is derived from*.

4. Generalization contains relationships that relate super-classes to sub-classes. The relationships of generalization include *is participant in* and *has participant*.

Even though these generic relationships provide semantics in tables, the relationships are not exhaustive. Then, this approach discovers non-defined relationships containing the entities of interest, which are discovered with the method explained in Section 4.3.2 Entity Recognition, Annotation, and Disambiguation.

The components of a generic relationship include the name of a relationship and its arguments, which can be entities, classes, and attributes. A class is an entity or an object that represents a concept. A property is an attribute of a given entity, which can also be an entity. An instance is a value of a property, which can also be an entity. Besides the semantic annotations obtained from the entity recognition, annotation, and disambiguation, entities from tables help in detecting relationships in tables and text within a publication.

Discovery of Semantic Relationships

To discover the relationships between entities derived from table cells, this approach uses as input the functional analysis explained in Section 3.2.3 Table Composition and Functional Analysis. In particular, I present how the relationships between entities in tables and between entities in the text of a publication are detected.

Relationships in Tables

To find relationships in tables, I include all table cells. First, I select the entities of tables with a header cell function. This approach relates each entity in a header to its dependent data cells. As mentioned, the functional analysis produces a representation containing for each table's cell its data, cell function, and data type. In addition, this representation contains indices that indicate the number of column for each row, which facilitates to relate header cells to data cells.

Formally, a JSON format represents each relationship. As follows, I show an example of a relationship extracted from a table. In particular, a relationship is a 6-tuple containing the number of a relationship in each publication, first and second arguments of a relationship, name of a relationship, original text, and identifier of a relationship in SIO [75]. The keys to represent a relationship are **id**, **argument A**, **label**, **argument B**, **text**, and **sio_id**.

```
{
   "id":"1",
   "argumentA":"Disease",
   "argumentB":"Diabetes"
   "label":  "has attribute",
   "text":"Disease has attribute Diabetes",
   "sio_id": "SIO_000008"
}
```

Relationships in Publications

Similarly to a table, I use the text of a table's source publication to find additional relationships. The text describes in more detail a table's content because it should contain at least its caption or description in the narrative of a document. Because the writing on a publication varies, the writing style can make it clear or difficult to understand. Therefore, I use different methods to detect relationships in text.

To detect the semantic relationships from a table's entities to the text of a table's source publication, I research two different methods. In the first method, I perform a) Open Information Extraction, b) segmentation, and c) pattern matching.

The Open Information Extraction methods identify relationships in plain text without training data for a particular set of documents. Reverb [66] belongs to the family of OIE, which uses an unsupervised learning and finds relationships not defined previously. In addition, Reverb assigns a confidence to each relationship extracted. Although Reverb is efficient, some relationships may contain incomplete arguments. For instance, from the sentence *"obesity and weight gain are associated with an increased risk of diabetes"*, Reverb finds with 0.80 confidence the relationship (*obesity and weight gain*; *are associated*; *with*). Because Reverb might miss crucial information on a relationship, I use its output only as a preliminary guide in the relationship extraction procedure.

I use Reverb to select relationships with a confidence greater than 0.70. From these relationships, I search for the ones containing known entities from the entity recognition process. The resulting relationships guide the search to find relationships in the text of a publication. The complete text undergoes segmentation, which is the process to separate the sentences, such as indicated in [112]. Then, segmentation is part of the Natural Language Processing (NLP) using Textblob [107], a semantic tool for this regard. Then, the relationships from Reverb are used to perform pattern matching on the original document to find the relationships' arguments. I use Reverb as explained in [30].

Because Reverb misses some relationships containing entities derived from tables, I research a second method to detect relationships between categorical entities from text.

This second method breaks the dependency on using Reverb. This process is based on natural language processing and contains four steps: a) segmentation, b) pattern matching, c) part of speech tagging, and d) relationship composition.

First, I perform segmentation in the text of a publication, using the Python Natural Language ToolKit [106]. From this step, I recover a set of sentences per publication. Second, I use the entities detected in the previous section to apply pattern matching to the sentences recovered. Third, for each sentence containing a known entity, I perform tokenization and part of speech tagging to detect relationships and additional entities. Finally, I apply Cartesian product to sets of entities to compose relationships.

For each sentence, I detect a tagged verb and use it to divide a sentence. Then, I search for new entities in each side of a detected verb. For instance, the sentence *Obesity is related to lack of exercise* decomposes into verb = "is related", left argument = "obesity", and right argument = "to lack of exercise". From these arguments, I search for unique entities not previously found in tables or context. This example contains the entities "obesity" and "exercise". The former was previously found in a table, then it is already disambiguated. While the latter undergoes the disambiguation process as explained in Section 4.3.2 Entity Disambiguation.

A challenge with the part of speech process is that there may exist ambiguity, that is when the same word can take different tags depending on its usage. For instance, the word "select" in the sentences *select the option* and *I prefer a select group* can be used as a verb or an adjective. Similarly, there are uncommon words that look like a verb, but they are a noun. For example, the noun “artesunate” that is a medicine for malaria.

To avoid ambiguous tags, I create a model with fifty sentences containing known words that can be used in different parts of speech. Then, this method uses the Python Natural Language ToolKit as a default tagger and the defined model to improve this issue. To identify verbs, in particular I use a combination of different tags, such as `VB`, `VBZ`, `MD VB`, `MD VBZ`, and `VB VBZ`, to detect verbs in different tenses and conjugations.

It is likely to find sentences that contain entities that have not been recognized previously. Then, this method detects, disambiguates, and uses them in a Cartesian product to form informative relationships. From the sentence *Obesity is related to lack of exercise and excesive eating.* The recovered sets of entities include $A =$ [“*obesity*”], derived from the left argument to the verb “is related”, as well as the set $B =$ [“*exercise*”, “*eating*”] derived from the right argument. To take advantage of the information in this sentence, I perform Cartesian product of sets of entities $A \times B$ to form relationships. From this case, I get the relationships: *Obesity is related to exercise* and *Obesity is related to eating*, while conserving the original text where a relationship is found.

I also extend this method to solve the co-reference resolution issue in sentences from publications. This issue occurs when a pronoun refers to a previously mentioned entity. Scientific publications’ authors use active voice mostly. Then, the pronoun “we” identifies the authors of a publication. From this observation, I detect this type of sentences to create relationships from the entity publication to other entities. For example, *We investigate the relationship between diabetes and exercise* contains the pronoun “we” on the left of the verb “investigate”. I replace the pronoun “we” with the name of this sentence’s publication to compose and recover informative relationships. Therefore, this process identifies a publication as an entity.

If a tag representing a verb is identified, I use this verb to represent a relationship. From the sentence *Obesity is related to lack of exercise*, the method obtains the relationship *is related*. Finally, I attempt to associate each relationship with a definition from an ontology to represent it.

The *Semanticscience Integration Ontology* [113] serves to formally represent relationships. Although SIO [75] defines relationships for objects, processes, and attributes in Bioinformatics, several of its definitions are useful to represent relationships in other areas of study. I select the most general definitions of relationships from SIO to represent relationships of publications in any domain. Finally, this approach attempts to associate each relationship with one of the SIO definitions (See Appendix A).

Finally, I use primarily pattern matching to associate a definition of a relationship from SIO with the extracted relationships. If a match is found, then I keep its identifier, label, and arguments. As follows, I show an example of the elements composing a relationship.

```
{
   "identifier":"2",
   "argumentA":"Obesity",
   "argumentB":"Exercise"
   "label":  "is related to",
   "text":"Obesity is related to lack of exercise",
   "sio_id": "SIO_000001"
}
```

If a relationship is not found in this ontology, this method can use either the verb(s) or relationship found by Reverb. These relationships definitions are stored by publication and can extend the SIO ontology.

The functional and semantic analyses assist to find metadata, context, entities, and relationships. In the following section, I describe a working example to show how to organize these elements and compose a synthesis of a publication.

4.3.4 Working Example to Synthesize a Publication

To depict this framework, I use as a working example the publication *Neuropeptidomic Components Generated by Proteomic Functions in Secretory Vesicles for Cell-Cell Communication* [114] by Vivian Hook et al. This publication contains common elements of interest for analysis: text and tables. Furthermore, this two-column document will help us to demonstrate partial results from this framework.

Working Example: Metadata and Context

Given this publication in PDF format, this framework executes TAO [29] to extract the tables embedded in this document and to recover the functional analysis of each table cell. Using the structured organization from TAO, this approach discovers the metadata to keep the provenance and context of a publication.

For simplicity, I include metadata of a publication with the properties: creator, headline, and keywords. Although the metadata is short, it allows to identify a publication rapidly, to store its context, and more importantly to keep its provenance. A context supports entity disambiguation and further analysis for documents. In addition, the metadata is related to entities and semantic relationships extracted with this approach.

The representation of this information includes `"creator"`, `"headline"`, and `"keywords."` The following JSON organization shows the metadata information. The annotation `"schema"` indicates that the concepts are defined from the `https://schema.org` vocabulary with type ScholarlyArticle.

```
{
   "schema":"https://schema.org/ScholarlyArticle",
   "schema:headline":"Neuropeptidomic Components Generated by
    Proteomic Functions in Secretory vesicles for Cell-Cell
    Communication",
   "schema:creator":[
      "Vivian Hook",
      "Steven Bark",
      "Nitin Gupta",
      "Mark Lortie",
      "Weiva D. Lu",
      "Nuno Bandeira",
      "Lydiane Funkelstein",
      "Jill Wegrzyn",
      "Daniel T. O Connor",
      "Pavel Pevzner"
   ],
   "schema:keywords":[
      "bioinformatics",
      "cell-cell communication",
      "mass spectrometry",
      "neuropeptides",
      "neuropeptidomics",
      "proteomics",
      "secretory vesicle"
   ]
}
```

Working Example: Entity Recognition, Annotation, and Disambiguation

The functional analysis generates a JSON file with the information of the table grouped by row. The output for each cell includes at least content, data type, and function of a cell (header or data label).

Table 4.1: Excerpt from Table Neuropeptides in the Nervous and Endocrine Systems

Neuropeptides	**Physiological Functions**
Insulin	Glucose metabolism
Somatostatin	Growth regulation

From the organization of tables extracted, all string cells undergo a process to identify relevant concepts. I relate these concepts to an annotation from the ontology DBpedia [115]. If a concept is not found or needs disambiguation, I use Latent Semantic Indexing [57] and a context to find a URL from the Web. For more details, see Section 4.3.2 Entity Recognition, Annotation, and Disambiguation. Table 4.1 contains an extract of a table in the *working example*. Using these methods, I identify the concepts *Neuropeptides* with annotation `http://dbpedia.org/page/Neuropeptides`, *Somatostatin* with `http://dbpedia.org/page/Somatostatin` and *Growth regulation*. This last concept is not found in DBpedia. Therefore, this entity undergoes the disambiguation method, which finds a URL to explain *Growth regulation*: `https://academic.oup.com/endo/article-abstract/121/1/352/2541431`. Notice that this approach can find another document containing an explanation about this concept.

In addition, this method recovers descriptions from DBpedia. I use such descriptions to create an *"IsA"* relationship, for the entity `"Neuropeptides"`, I recover *"IsA"*: *"small protein-like molecules (peptides) used by neurons to communicate with each other..."*

This framework finds automatically important concepts with annotations for each publication. A publication might share annotation of concepts from different tables. I create a global list of entities to reuse annotations within a publication as appropriate. Therefore, this method does not search the Internet for annotations already determined. The recognized entities with annotations are the main components of a semantic relationship.

Working Example: Discovery of Relationships

Using the concepts from entity recognition, I identify headers, data, and relationships between these table cells. This framework's table organization facilitates finding relationships between columns and rows. The relationships on the same row and column form closely linked relationships. For instance, the first row contains concepts in headers (e.g., Neuropeptides, Physiological Functions) and the other cells contain values that can be an entity instance or attribute. From Table 4.1, I use entities detected and get the following relationships:

- `Neuropeptides` *has attribute* `Insulin`,
- `Neuropeptides` *has attribute* `Somatostatin`,
- `Physiological Functions` *has attribute* `Glucose metabolism`, and
- `Physiological Functions` *has attribute* `Growth regulation`.

To represent each relationship, I use a 6-tuple containing **id**, **argument A**, **label**, **argument B**, **text**, and **sio_id**. An example showing the first relationship in table 4.1 from the *working example* follows.

```
{
   "id":"1",
   "argumentA":"Neuropeptides",
   "argumentB":"Insulin"
   "label":  "has attribute",
   "text":"Neuropeptides has attribute Insulin",
   "sio_id": "SIO_000008"
}
```

Similarly, the entities represent concepts in a publication that guide the semantic relationships extraction from text. To find the relationships in text, an unsupervised learning method and pattern matching assist for this process. The concepts identified in tables are associated to the text of a publication. The resulting associations compose semantic relationships.

For instance, these are two relationships derived from the *working example*: a) *"neuropeptides* ***is related to*** *short peptides ranging in length from ∼3 to 40 amino acid residues"*, and b) *"neuropeptides* ***represent*** *one of two main classes of neurotransmitters"*.

The Semanticscience Integrated Ontology [75] helps represent the relationships detected. This representation is composed of a 6-tuple including an identifier of the relationship, first argument, relationship label, second argument, original text, and a definition identifier. This representation uses two different types of identifiers, the first identifier refers to a relationship within a publication (e.g., **id:**1), and the second identifier to the definition of a relationship in SIO (e.g., **sio_id:**SIO_000020). An example of the representation of a relationship discovered in the text of this publication follows. Note that I use the label "denotes" as a relationship "is a".

```
{
   "id":"1",
   "argumentA":"neuropeptides",
   "argumentB":"peptides"
   "label":"denotes",
   "text":"neuropeptides are related to short peptides ranging
    in length from 3-40 amino acid residues",
   "sio_id": "SIO_000020"
}
```

Finally, I create a short summary containing metadata, concepts, annotations, and semantic relationships, which represent a synthesis per publication. See Chapter 5, Section 5.3 Organization of a Semantic Data Model.

4.4 Evaluation of Table Interpretation

To assess this framework quantitatively and qualitatively, I design three sets of experiments to evaluate 1) the number and quality of entities recognized for each table, 2) annotations and disambiguation for entities, and 3) the semantic relationships extracted. I report settings and results as shown in [30], as well as present new results for the extraction of semantic relationships.

4.4.1 Dataset and Experiments of Table Interpretation

For this assessment, I use a dataset containing fifty publications with different topics. The dataset *Pubmed* contains publications downloaded from the Pubmed Web site `ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/`. In summary, the dataset *Pubmed* contains 449 text pages and 133 tables with different tabular formats embedded either in one- or two-column documents. I manually prepared a gold standard containing the correct entities for each table in the publications of the dataset.

This framework is evaluated to test and measure its ability to (1) recognize and annotate entities, (2) disambiguate entities, and (3) identify semantic relationships between entities with high confidence.

The first experiment evaluates the method's accuracy to recognize and annotate entities. First, I use the measures *recall* and *precision* to obtain the average F1-measure for entity recognition. *Recall* is the ratio between the correct number of entities detected and the total number of entities in a table. *Precision* is the ratio between the correct number of entities detected and the total number of entities detected.

For the second experiment, the entity disambiguation method uses the same measures *recall* and *precision*. In particular, I quantify the effect of including information regarding the context of the publication in the disambiguation process. In addition, I evalute the quality of the disambiguation process. Although this process may be automated using domains of established and recognized institutions, such as universities, digital libraries, and cliniques. A human judge classifies the URLs obtained from this process as reliable and non-reliable. This manual evaluation is necessary to ensure that the URL is related to a reliable but not commonly known institution.

The third experiment evaluates both quantitatively and qualitatively the semantic relationships found by the method. Such relationships are extracted from tables or text. The hypothesis is that relationships containing relevant concepts in a structured presentation should be more explicit and compact. First, I measure the total number of relationships with high rank, in particular a confidence ≥ 0.70 using Reverb. Second, I measure the total number of relationships extracted with the method using relevant entities. In addition, a human judge evaluates qualitatively these relationships classifying them as complete and incomplete. A complete relationship contains the two arguments of a relationship.

4.4.2 Results and Discussion of Table Interpretation

The dataset *Pubmed* is used to perform a complete assessment for each one of the functions of this framework. As follows, I present the findings for each one of the experiments.

Experiment Entity Recognition and Annotation

The first experiment evaluates the method's ability to recognize and annotate entities. To measure entity recognition, the gold standard contains $2,314$ entities, which $1,834$ of them are recognized. I obtain a recall of 79.2 percent and a precision of 94.3 percent, yielding a F1-measure of 86.1 percent for recognition (see Table 4.2).

From the entities annotated, I find $1,262$, that is 72.5 percent of entities in DBpedia. From this number, 785 contain a unique description in DBpedia, that is, 45.1 percent. Therefore, I obtain 785 annotations using DBpedia. For the rest, 27.4 percent of entities need disambiguation. The method uses the context of a publication and the LSI process as described in [30] to annotate 955 entities. Then, 54.9 percent of entities are annotated with the LSI + context method. From the total of $1,834$ entities recognized, $1,740$ are correctly annotated, yielding a recall of 94.8 percent, precision of 97 percent, and F1-measure of 95.9 percent (see Table 4.2).

Table 4.2: Experiment Entity Recognition and Annotation

Entity Recognition			
Entities	***Recall***	***Precision***	***F1 measure***
Recognized	0.79	0.94	0.86
Annotated	0.95	0.97	0.96

Experiment Entity Disambiguation

For the second experiment, the entity disambiguation methods are evaluated. In particular, I quantify the effect of including the context of each publication in the entity disambiguation process. From the previous experiment, I find $1,740$ annotations of entities and 955 of them undergo the disambiguation process. The first part of this experiment does not account for a context, still 838 entities are discovered and disambiguated correctly without context. The precision is 89 percent and recall 87 percent, with an F1-measure of 88 percent. For the second part of this experiment, I use at least the three more relevant keywords as context, yielding 900 entities disambiguated and annotated. The precision is 95 percent and the recall 94 percent, producing an F1-measure of 94.5 percent. Table 4.3 presents the comparison between disambiguation with and without context.

Table 4.3: Experiment Entity Disambiguation

Entity Disambiguation				
Method	***Recall***	***Prec.***	***F1***	***NR urls***
No context	0.87	0.89	0.88	12.3%
Context	0.94	0.95	0.94	5.8%

In addition, a judge verifies and determines whether the URLs detected are reliable or non-reliable (see Table 4.3 column *NR urls*). While the reliable URLs are originated from known organizations and domains mainly, the non-reliable URLs require further analysis. This URL review ensures that a Web page or document is related to the previously unresolved entity, and consequently to the publication. However, it does not ensure an exact description of an entity. The resulting non-reliable URLs with no context are 117, that is 12.3 percent of the total disambiguated entities, and the number of non-reliable URLs found with context are 55, that is 5.8 percent. Therefore, the context helps reduce more than half of non-reliable links.

Even though the results with context are slightly better than with non-context, the quality of the URLs using the context increases considerably. For instance, a publication containing research findings about obesity, the entity `Gain` using the method with no context finds the URL `http://gainworldwide.org/`, which is a site for global aid network. While using a context, it finds the URL `https://www.sciencedaily.com/releases/2010/02/100222182137.htm`, which is a site about weight gain during pregnancy and increasing risk of gestational diabetes. It is obvious that the first link belongs to an established organization, but it does not relate to the entity `Gain` for this publication. On the other hand, the second URL explains the risks of weight gain during pregnancy, which is related to the publication.

Using context, I find more reliable links consistently. The links belong to organizations, schools, government, clinics, dictionaries, and scientific and digital libraries, among others. On the contrary, when a context is not included, I find several commercial URLs promoting services or products, unavailable sites, sexual sites, and ill-intentioned links. As mentioned earlier, unlike URIs the URLs can become unavailable or can change over time. To ensure reliability, the method could keep a set of at least three URL annotations to monitor and avoid this situation.

Experiment Semantic Relationships

The third experiment evaluates both quantitatively and qualitatively the semantic relationships. I report the number of relationships originated from tables and text, as well as the average of relationships from the dataset of publications.

For this experiment, I find $11,268$ relationships from tables. Using Reverb, I find 865 high ranking (confidence ≥ 0.70) relationships from text. On the other hand, not using Reverb, I find $1,397$. The average number of relationships extracted per publication when using table information is 225, while the average number of relationships extracted using only text is 17 per publication. This average increases to 27 when using categorical entities to search relationships.

Table 4.4: Experiment Semantic Relationships

Semantic Relationships		
Method	***Rel. found***	***Rel. complete***
Tables	11,268	10,102
Text Reverb	865	703
Text New	1,397	1,336

A human judge analyzes the completeness of relationships manually. The relationships from tables contain $10,102$ or 89 percent of complete relationships. From the total of relationships extracted from text using Reverb, 703 or 81 percent of relationships are complete. The rest, that is 19 percent is labeled as incomplete. For the approach that does not use Reverb, a judge classifies 1,336 or 90% of them as complete, while the rest 10% is labeled as incomplete. The improved approach finds more relationships than the previous one based on Reverb. The number of complete relationships almost doubles the number found with the previous method. Therefore, the number of relationships extracted from each article increases using relevant entities.

In addition, the semantic relationships from text increases because the method uses a publication's context (i.e., keywords) as entities, as well as concepts found in tables' cells. The entities composing relationships ensure their relevance. On the other hand, some of the incomplete or malformed relationships from text include information from tables. An area of improvement for this approach is to extract and eliminate tables' information in a document to avoid processing twice this information. In addition, this step could eliminate some false positives relationships from text.

From the qualitative results, the relationships extracted from tables are more complete than from text. This confirms the initial hypothesis about extracting relevant concepts and structured information embedded in tables. Although this framework bases its analysis in concepts from tables to extract relationships and generate a synthesis, it can still be used when a publication lacks tables because it finds metadata and can find semantic relationships from text. The semantic relationships can be found using a publication's context (i.e., keywords).

The keywords and concepts from tables increases the number of entities. Therefore, the quantity and quality of semantic relationships from text increases when relying on entities and natural language processing. Finally, regarding the relationships extracted from text, the method improves the accuracy of relationships extracted by Reverb searching directly for important entities.

In summary, I identified the metadata and context of publications that contain tables. Second, I identified entities from the tables. In addition, I used semantic annotations to ensure that the entities were unique. Third, I determined the relationships among the entities and other elements in the tables and the text in a publication, and represented the semantic relationships in a structured format. I created a synthesis for each publication containing its metadata, entities, and relationships. Finally, I evaluated the performance of this framework using a set of publications and measured the quality and accuracy of the findings.

Chapter 5

Automated Generation of Semantic Data Models

Scientific digital documents contain valuable information that very often, machine-readable approaches cannot consume directly. The large volume of digital publications produced daily makes it difficult to analyze them rapidly. Therefore, the scientific community needs novel methods to discover the legacy of knowledge embedded in a vast number of digital publications.

To facilitate this analysis, I present a conceptual design to create semantic data models from scientific documents. I also develop methods to achieve the daunting task of extracting and representing semantic information from digital publications systematically. Specifically, I develop an automated framework to construct the models, which are based on detection and extraction of table information within documents [29], table interpretation and extraction of semantic relationships [30], and organization of other relevant information, such as provenance and context. The union of the models allows us not only to analyze, but also to integrate, manage, share, and compare information.

5.1 Publications with Semantic Interoperability

To find concrete information embedded in a vast number of scientific documents, it is necessary to analyze documents with semantic interoperability. This interoperability refers to integrated information systems containing context, and hiding syntax and structural heterogeneity [25].

The integration of embedded information systems keeps growing rapidly with mobile devices and applications [116], facilitating the everyday interaction among people, data, and devices in the era of the Internet of Things. In contrast, because the research community relies on a variety of publishing standards and traditional policies to circulate its findings, there is limited interoperability among scientific disciplines. To alleviate this problem, S. Peroni [26] introduces the semantic publishing and referencing ontologies to facilitate using scientific articles. This kind of publishing, however, is not widely adopted.

Other challenges to analyzing publications include the lack of a unified vocabulary in publications within the same domain. Furthermore, the variety of topics and formats constrain the reusability of scientific information. The International Association of Scientific, Technical and Medical Publishers reports on the different techniques used by researchers and scholars to read scientific publications [117]. A. Renear and C. Palmer [118] study the evolution of the average time that a researcher spends reading a journal article. It has decreased from 50 to just over 30 minutes, suggesting that researchers scan articles instead of reading them. Researchers lack methods to channel and reuse this information at the same speed as publishing does.

To facilitate understanding publications, a few tools allow to integrate semantic annotations of relevant concepts, such as BioLit [18]. Others allow users to select content of interest within a publication to search for additional information, such as the visual tool Utopia [119] that allows interaction in PDF documents. For scalability, however, it is better to search for related information automatically and do not rely on interaction.

In science, the Semantic Web is paramount for scientific advancement and interoperability among disciplines [120], [121]. This initiative provides principles [92] to facilitate reusing information, however, these rules mainly focus on markup languages, for instance, XML. Because the scientific community uses diverse formats, it should extend these principles to other commonly used formats, for example, PDF.

The information technologies to store and manage information have evolved from using plain text files to relational, to object-relational, to “not only SQL” and graph databases. It is still difficult, however, to represent explicit semantics in any data organization. To minimize this problem, I recall the concept of Semantic Data Model proposed by different authors a few decades ago [122], [123]. J. Peckham and F. Maryanski review different semantic data models and establish two common elements in *Semantic Data Models* [6]:

> a) relationships between data objects (i.e., entities) that support the manner in which the user perceives the real-world enterprise, and
> b) semantics for the relationships that specify the acceptable states, transitions, and responses of the database system.

To increase semantic interoperability and reusability, I present the conceptual design of a semantic data model to characterize publications and facilitate information analysis. I use this concept to integrate relevant information from common elements in publications. To compose a semantic data model, I extract a collection of semantic relationships from digital publications with well-defined entities and relationships. I dynamically discover entities not defined *a priori* as the main components of semantic relationships, which are derived from structural associations represented in tabular patterns and unstructured text. The semantic relationships hide the integration of diverse sources of information. Finally, I preserve a publication's context and metadata to identify the source of each model.

Each semantic data model represents a synthesis of a publication. I stress the importance of keeping the original semantics of each individual synthesis to facilitate relating information among scientific publications and disciplines. Moreover, I use sources of knwoledge, such as a dictionary, a vocabulary, and general ontologies following the Semantic Web principles to compose independent data models. Yet, the models can interact with one another using common context and concepts.

To relate information among publications, A. Nenkova and colleages [79] mention the importance of using citations to discover relationships with related, previous, and future work. SimRank [24] uses references and titles of publications to infer relationships. Although these works successfuly find relationships among publications, the relationships are not necessarily semantic. Therefore, I use the semantic relationships from the data models for this matter.

Using a collection of data models, I compose a semantic network of publications, where each model contains minimal and relevant information, such as well-defined semantic relationships and provenance of a document. The network assists us in managing and analyzing large datasets of publications. More importantly, the models facilitate to discover knowledge in an organized mode and to track the source of the findings. Furthermore, given that this investigation integrates concrete and available information, it can potentially assist in reproducing research in diverse scientific areas.

In this chapter, I present a conceptual design to create semantic data models, an organization approach, as well as an approach to using a single model and a collection of them. Figure 5.1 highlights the processes described in this section, which allow us a) to create a semantic data model from each publication; b) to build a semantic network, which is a collection of semantic data models; and c) to retrieve information from a network of publications.

Therefore, the main contributions include 1) a formal definition to dynamically generate semantic data models, using integrated information from digital documents; and 2) an approach to creating a semantic network of data models to manage and reuse relevant information from scientific documents of any domain.

For the remainder of this chapter, Section 5.2 contains the components in a semantic data model and its conceptual design is explained in Appendix A. Section 5.3 describes the special linked data format to organize a semantic data model. Section 5.4 contains an application to create a semantic network of digital publications. Finally, Section 5.5 contains an evaluation using a set of experiments to find semantic relationships, to identify similar semantic relationships among publications, and to measure time for model generation. To conclude, I report results and discuss the findings.

5.2 Conceptual Design of Semantic Data Models

To detect and extract relevant information from scientific documents, I generate semantic data models from publications systematically. I detect entities not defined *a priori* in tabular structures and unstructured text, as well as relationships among entities within a publication and with external sources of knwoledge, such as a dictionary, ontologies, and the Internet. Figure 5.1 features the last two processes of this framework to build **Semantic Data Models** and **Semantic Network of Data Models**. In this section, I define the components of Semantic Data Models.

From the elements analyzed in a publication, I detect the finite components of a semantic data model that include metadata, table cells, entities (i.e., concepts),

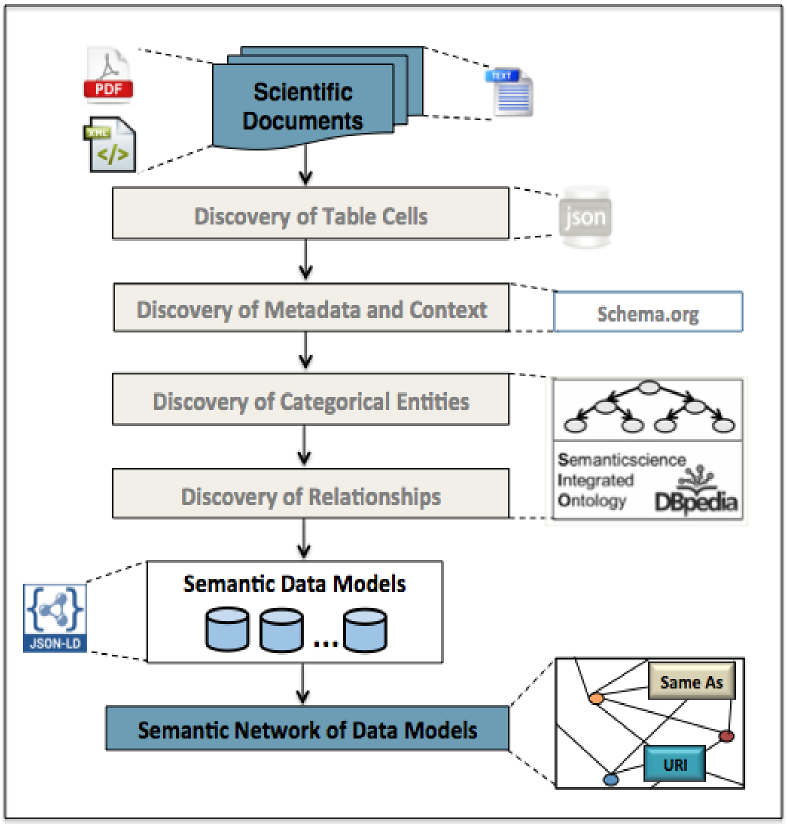

PDF
XML
Scientific Documents
TEXT
Discovery of Table Cells
json
Discovery of Metadata and Context
Schema.org
Discovery of Categorical Entities
Discovery of Relationships
Semanticscience Integrated Ontology
DBpedia
JSON-LD
Semantic Data Models
...
Semantic Network of Data Models
Same As
URI

Figure 5.1: The Semantic Data Integration Process

as well as semantic relationships extracted from tabular structures and unstructured text. Formally, a semantic data model contains a 3-tuple $SDM = \{M, E, SR\}$, where M includes metadata and context, E entities, and SR semantic relationships. Appendix A contains more details of this conceptual model.

5.3 Organization of a Semantic Data Model

To find the most convenient representation for a data model from digital documents, I analyze several formats. One of the most common languages to describe ontologies is the RDF, which is based on concepts and abstract syntax. RDF represents meaningful relationships between concepts using directed graph data models, URI-based vocabulary, data types, literals, XML serialization syntax, expression of simple facts, and deduction [124]. Every graph representing concepts comprises a *triple*: subject, object, and predicate, representing a property of the subject. RDF uses URI, which is a global identification for a resource that is common across the Web. URI is a generalization of the URL [125] that can be used as a namespace in RDF. Then, RDF is a widely used language that can be represented in format XML, however, RDF does not include a feature to describe a context.

JSON excels in representing data in a portable, compatible, and easy-to-use format. An extended version of this format is JSON-Linked Data (JSON-LD) [126], which facilitates the usage of linked data. JSON-LD is a novel format commonly used to interchange information for REST (REpresentational State Transfer) services [127]. The advantages of this format are that JSON-LD allows a user to specify a context, define the type of information represented, and enable integration with the Semantic Web using an explicit machine-readable format. Another benefit of JSON-LD includes representation of relationships not limited to triples, using lists to avoid repetition. This format is compatible with RDF and other variants such as N-quads [128].

For the exposed advantages, this framework outputs a JSON-LD representation where I define a context of a semantic data model, containing an environment of its components. The first element of a semantic data model is its context. Note that this context is different from the one used in a publication that is represented with keywords.

The elements of a context include all the resources used to represent metadata, entities, attributes, and relationships. For a publication's metadata, I use the vocabulary resource schema.org to represent author, title, and keywords. If an entity is represented using JSON, I can convert JSON to JSON-LD seamlessly. For instance, the following entity is represented in JSON:

```
 "Neuropeptides":
 {
            "URI":"http://dbpedia.org/page/Neuropeptides",
            "IsA":"small protein-like molecules (peptides)"
 }
```

I can easily convert it to JSON-LD, using part of a URI as the context of an entity. The context "http://dbpedia.org/page/" indicates that it is a resource from the ontology DBpedia. This property abstract belongs to the entity *Neuropeptides*:

```
{
 "context": "http://dbpedia.org/page/"
 "Neuropeptides":
 {
       "id":"Neuropeptides",
       "abstract":"small protein-like molecule (peptides)..."
 }
```

Therefore, a semantic data model's context contains the location of resources for the sources of knowledge, such as ontologies and vocabulary. These resources facilitate annotation of entities, definition of relationships, and vocabulary standardization. This approach also organizes metadata, entities, annotations, and relationships into this special JSON file that characterizes and synthesizes each publication, allowing simplification and interoperability.

In particular, JSON-LD serves to generate a descriptive document. The document contains the publication's metadata (i.e., headline, creator, keywords, email), entities with annotations (i.e., URL, URI, IsA), and the actual information from extracted relationships (i.e., relationship identifier, arguments, definition of relationship, definition identifier). The document in turn contains other links, such as the ones used to annotate entities and to define relationships.

I build each semantic data model collecting the elements of SDM. The representation contains explicitly M, E, and SR as explained in Appendix A. This organization provides the description of entities and relationships to understand tables, and consequently publications. Therefore, the organization of a semantic data model has a two-fold purpose:

- to represent and understand the content of tables embedded in a publication, including semantic relationships among unique concepts within a document, and
- to identify the blueprint or synthesis of a publication, including provenance and concrete information.

I present an extract of the representation of the *working example* introduced in Chapter 4, Section 4.3.4. Note that the context of a semantic data model is represented with `"keywords"`, while the context to represent resources uses the word `"context"`. Appendix C contains a context of resources used in the representation of this data model.

```
{
"context": {
     "headline":
      {
         "@id": "http://schema.org/ScholarlyArticle",
         "@type": "@id"
      }, ...
}
```

```
"headline":"Neuropeptidomic Components
 Generated by Proteomic Functions in Secretory vesicles for
 Cell-Cell Communication",
"creator":[
   "Vivian Hook",
  ...
],
"keywords":[
   "bioinformatics",
   ...
],
"email": [
  "vhook@ucsd.edu"
],
"entities": {
    "Neuropeptides": {
         "URI":"http://dbpedia.org/page/Neuropeptides",
         "IsA":"small protein-like molecules (peptides)"
         },
    "somatostatin": {
         "URI":"http://dbpedia.org/page/Somatostatin",
        },
    "Peptide": {
         "URI":"http://dbpedia.org/page/Peptide",
        }
    "Amino acid residues": {
         "URI":"http://dbpedia.org/page/Protein_structure",
        }
     }
```

```
"relationships": {
    "1": {
         "id":"1",
         "argumentA":"neuropeptides",
         "argumentB":"somatostatin"
         "label":"has attribute",
         "relationship":"neuropeptides has attribute
          somatostatin",
         "sio_id": "SIO_000008"
         },
     "2": {
          "argumentA":"neuropeptides",
          "argumentB":"peptides",
          "label":"denotes",
          "relationship":"neuropeptides are short peptides
          ranging in length from 3-40 amino acid residues",
          "sio_id": "SIO_000020"
         }
      }
 }
```

The context of a JSON-LD document helps locate resources on it. From the previous example, the context `"http://schema.org/ScholarlyArticle/headline"` is represented by the key `"headline"` described in the vocabulary schema.org. For simplicity, I only use the class ScholarArticle. The property email is represented using the class Thing because it does not appear in the ScholarArticle class. The email property can also be found in the classes ContactPoint, Organization, and Person of this vocabulary.

I represent relationships from the resource SIO. Appendix B shows the context

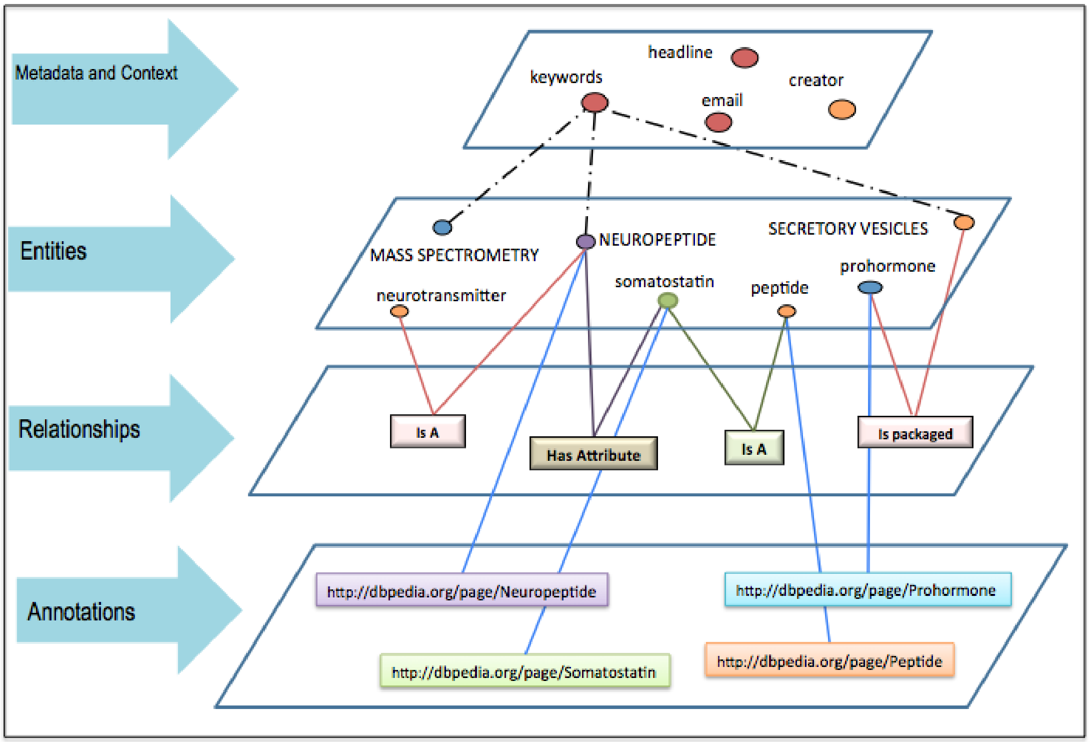


Figure 5.2: Components of a Semantic Data Model

`http://semanticscience.org/resource/` and key `"sio_id"`, which indicates the location of the formal definitions of relationships in SIO and its type as an id. For instance, a relationship with id SIO_000001 can be related to this context: `http://semanticscience.org/resource/SIO_000001`, which resolves the definition of this relationship and gives access to a specific resource.

Figure 5.2 shows additional relationships of a semantic data model collected from the *working example*. The entities in capital letters indicate that they are also part of the keywords. The well-defined relationships in the models exist at each level of its components. I find relationships among keywords, entities, and structured associations. Additional concepts exist that may not be part of the categorical entities in the model organization, for instance, the concepts *prohormone* and *peptide* found in text. Then, researchers can use these relationships to find specific results, settings of experiments, and associations among concepts in scientific environments. For simplicity, the direction of relationships are not shown.

This representation using a context organizes relevant information from a digital publication. The organization facilitates the creation of semantic data models in a structured organization to allow a) reusing information; b) having interoperability (i.e., connections) among common elements, such as metadata, entities, and relationships; and c) integrating information within and among publications. In the following section, I explain an approach to obtain a network of the data models.

5.4 Semantic Network of Data Models

After integrating semantic information from publications and representing it in a flexible format, the next step is to use the systematic representation to achieve the ultimate goal of this investigation: to relate the semantic data models with one another using common entities and finding their relationships across publications. Figure 5.3 depicts the main elements of semantic data models and the possibilities of connections within and among publications. Even though the goal of this investigation is to recover semantic information, the models also contain metadata that can be exploited, such as the creator of a publication.

5.4.1 Approaches to Use Semantic Data Models

To use the data models, I consider their portability and readiness that provide the ability to store and query them with no additional processing. To store the models, because of their ability to expand and scale [129], I consider using a NoSQL database, however, because the models are flexible and descriptive, I can use other means to analyze their embedded information. I explore two platforms ready for Big Data analysis, Apache Spark [130] and Networkx [131].

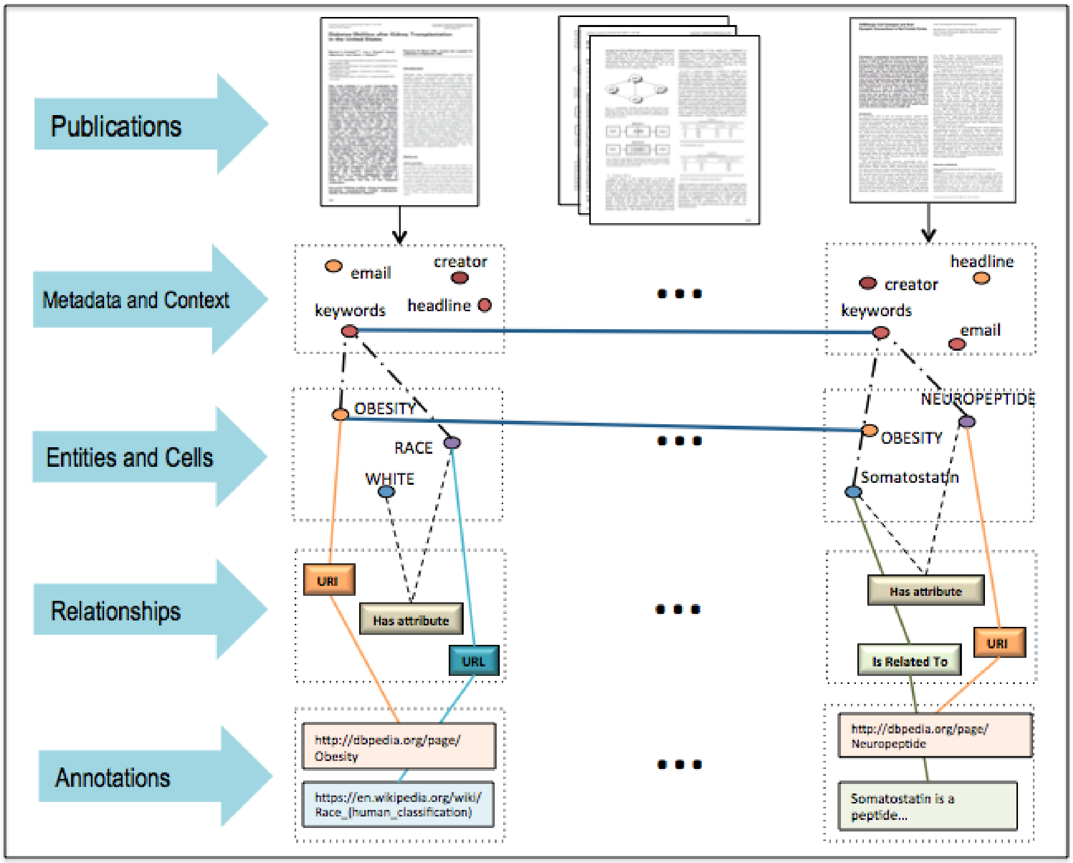


Figure 5.3: A Semantic Network of Data Models

First, Spark provides advantages over other data analysis systems due to its functions for large-scale data processing, reliability, and parallelization [132]. A researcher can use any method to reuse and interoperate among a set of semantic data models generated with the approach.

I use Spark to analyze and query several publications. To do so, I concatenate a set of semantic data models into a unique file. I avoid further processing and create a document, where each line represents a data model in JSON format.

The organization of the semantic data model allows Spark to read the models as a data set. Spark allows us to select specific information as using a database. Spark allows us to query the elements of a model. For instance, I can query author and title of a publication.

Second, I use Networkx [131] to build a network of data models. Networkx is a favorable tool that allows us to create a graph. To do so, I unravel the components of the semantic relationships (e.g., entities and relationships) contained in the models. Each unambiguous entity represents a node, and the relationship between two nodes represent an association, that is, an edge. Furthermore, Networkx allows us to define properties in the elements of a graph, i.e., nodes and edges. I take advantage of this property to include the semantic annotations or explanations composing a relationship in the model and other relevant information.

Figure 5.4 shows how a relationship from the *working example* is composed of two nodes and an edge. The edge has the property "relationship" and can have more properties. A node has a unique identifier. Similarly, an edge is composed of node$_1$, node$_2$, and the relationship between those nodes. In addition, I use annotations, such as the name of a publication and the text that shows how an entity participates in a relationship. To compose a relationship from the sentence *neuropeptides are peptides ranging in length...*, I obtain the nodes *neuropeptides* and *peptides* and the relationship *are* as an edge. To preserve the origin of this relationship, I use two properties in the edge between these nodes. The first annotation contains the whole sentence where the relationship is found and the second one contains the name or its publication. Using this process, I define nodes and edges from each data model.

To demonstrate the construction of a network from a semantic data model, the *working example* serves for this purpose. Figure 5.5 shows an example of the entities and relationships composing a network. In particular, a network is a directed graph $G = (V, E)$, where V represents the nodes and E the edges. Later, Networkx allows

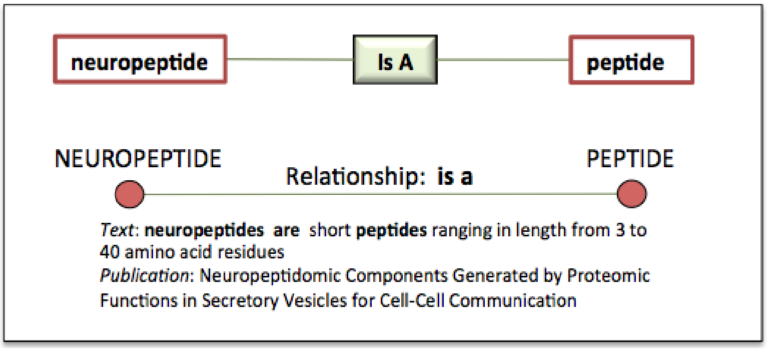
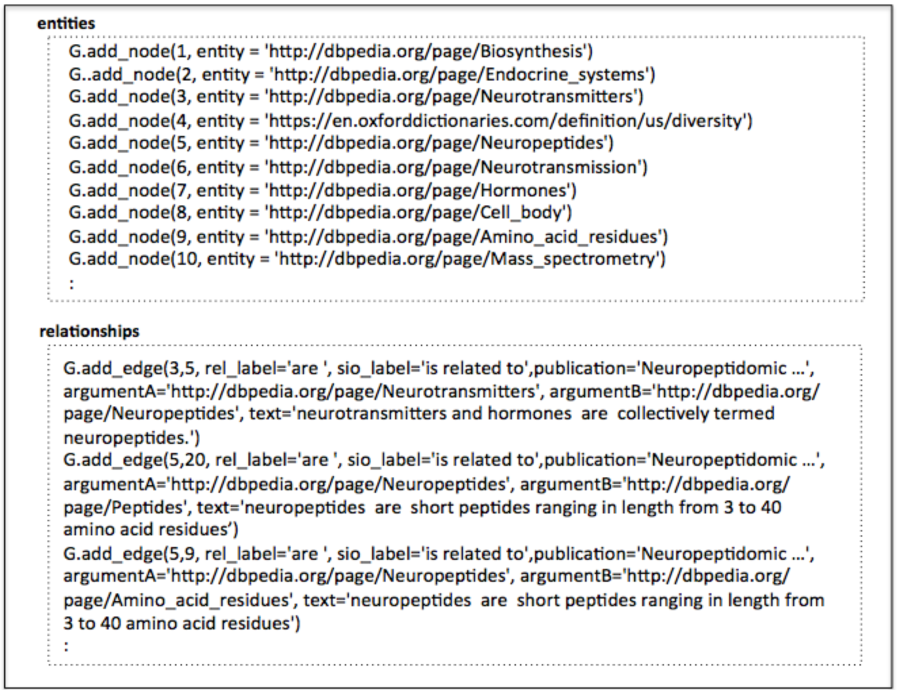

neuropeptide
Is A
peptide
NEUROPEPTIDE
Relationship: **is a**
PEPTIDE
Text: **neuropeptides are** short **peptides** ranging in length from 3 to 40 amino acid residues
Publication: Neuropeptidomic Components Generated by Proteomic Functions in Secretory Vesicles for Cell-Cell Communication

Figure 5.4: Elements to Build a Semantic Network of Data Models

us to relate the models with one another using a node identifier $v \in V$.


entities

```
G.add_node(1, entity = 'http://dbpedia.org/page/Biosynthesis')
G..add_node(2, entity = 'http://dbpedia.org/page/Endocrine_systems')
G.add_node(3, entity = 'http://dbpedia.org/page/Neurotransmitters')
G.add_node(4, entity = 'https://en.oxforddictionaries.com/definition/us/diversity')
G.add_node(5, entity = 'http://dbpedia.org/page/Neuropeptides')
G.add_node(6, entity = 'http://dbpedia.org/page/Neurotransmission')
G.add_node(7, entity = 'http://dbpedia.org/page/Hormones')
G.add_node(8, entity = 'http://dbpedia.org/page/Cell_body')
G.add_node(9, entity = 'http://dbpedia.org/page/Amino_acid_residues')
G.add_node(10, entity = 'http://dbpedia.org/page/Mass_spectrometry')
:
```

relationships

```
G.add_edge(3,5, rel_label='are ', sio_label='is related to',publication='Neuropeptidomic ...',
argumentA='http://dbpedia.org/page/Neurotransmitters', argumentB='http://dbpedia.org/
page/Neuropeptides', text='neurotransmitters and hormones are collectively termed
neuropeptides.')
G.add_edge(5,20, rel_label='are ', sio_label='is related to',publication='Neuropeptidomic ...',
argumentA='http://dbpedia.org/page/Neuropeptides', argumentB='http://dbpedia.org/
page/Peptides', text='neuropeptides are short peptides ranging in length from 3 to 40
amino acid residues’)
G.add_edge(5,9, rel_label='are ', sio_label='is related to',publication='Neuropeptidomic ...',
argumentA='http://dbpedia.org/page/Neuropeptides', argumentB='http://dbpedia.org/
page/Amino_acid_residues', text='neuropeptides are short peptides ranging in length from
3 to 40 amino acid residues')
:
```

Figure 5.5: Nodes and Edges to Build a Network Using the *Working Example*

5.4.2 Analysis of a Semantic Network

The important nodes in a semantic data model provide insights for an exploratory analysis. Centrality measurements help finding important nodes. The centrality measures include degree, closennes, and betweenness. S. Uddin et al. [85] mention the importance of these measures to co-relate authors of scientific publications. Instead, this approach uses the measures to detect important nodes, i.e., entities, and their relationships derived from different publications. K. Batool and M. Niazi [133] compare centrality measures in different networks. The degree and closeness centrality measures prove to be useful, and betweeness measure varies depending on the topology of a network.

The centrality measures for undirected graphs defined in [133] follow. Degree of a node v is the number of edges that leave or get to this node. Equation 5.1 shows how to calculate degree centrality, where k_v refers to the degree of a node v and n is the number of nodes in a network.

$$C_D = \frac{k_v}{n-1} \tag{5.1}$$

Closeness centrality of nodes v and t uses the length of shortest paths to measure the distance between two nodes. Equation 5.2 shows how to calculate closeness centrality, where $distance(v,t)$ refers to the number of nodes that v touches to get to t.

$$C_C = \sum \frac{1}{distance(v,t)} \tag{5.2}$$

Betweenness centrality calculates the number of times that a node v participates in a shortest path between other two nodes. Equation 5.3 shows how to calculate betweeness centrality, where $\sigma_{st}(v)$ refers to the number of paths that v participates and σ_{st} is the number of shortest paths from s to t .

$$C_B = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{5.3}$$

For instance, from the *working example*, the most important node is *neuropeptide* according to the measures of centrality for degree, closeness, and betweeness. Other relevant nodes include *peptide*, *neurotransmitter*, *secretion*, and *mass spectometry.*

5.4.3 Applications of a Semantic Network

The conceptual design allows us to create semantic data models that contain a synthesis of a publication. I can analyze each model individually. Similarly, a set of models can be analyzed globally. In addition, I can create a network of models with common context or entities. Then, a semantic network of data models allows us to better exploit information from digital publications.

The applications of a network of semantic data models include the ability to integrate, manage, share, query, compare, and contrast information among publications. The main elements to use in a semantic network are entities, context, and semantic relationships. Because the semantic relationships are binary, structural, and domain independent, they facilitate the integration of information.

A semantic network of data models allows researchers and scholars to integrate publications with similar content and find the semantic relationships in publications with this commonality. In addition, the common semantic relationships allow us to identify concrete information in each publication, and to compare the information among publications.

A semantic network allows us to find semantic relationships with common entities and context, as well as other relationships among publications. Finally, the characterization of the data models allows us to use their information in different data management systems, including machine-readable.

In the following section, I show results of generating data models using this framework and querying entities to find specific semantic relationships in a network of data models.

5.5 Evaluation of Semantic Data Models

I evaluate this framework with two sets of experiments. The first set computes the average time to generate the semantic data models from discovering context, metadata, table extraction, and semantic relationships to model organization. The second set of experiments consists on creating networks of semantic relationships from two collections of data models.

5.5.1 Dataset and Experiments of Data Models

To assess this framework, I use two datasets downloaded from the PubMed website at `ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/`. The first dataset $Pubmed_1$ contains twenty publications obtained from querying the keyword *diabetes* with 210 pages and 18 tables. The second dataset $Pubmed_2$ contains four hundred and eighty publications obtained randomly from different topics with 3,600 pages and 390 tables.

I evaluate this approach using two sets of experiments to test and measure this framework in terms of its ability to (1) generate semantic data models and (2) identify important entities and semantic relationships among publications, and (3) measure the time of generation.

In the first set of experiments, I create two semantic networks of data models using *Pubmed*$_1$ and *Pubmed*$_2$. I obtain statistics for each network and use centrality measures to report important concepts from the semantic relationships composing the networks of publications. In addition, I report relationships from different models sharing similar annotations. The second assessment evaluates the average time to generate semantic data models using the dataset *Pubmed*$_2$.

5.5.2 Results and Discussion of Data Models

During this evaluation, I generate and organize semantic data models, as well as detect semantic relationships in networks of digital publications. Finally, I present the average time to generate the models.

Experiment Network with Semantic Relationships Among Publications

The first set of experiments consists of creating a semantic network of data models. First, I generate a network for the dataset *Pubmed*$_1$ that includes publications with the keyword "diabetes". From the network generated with *Pubmed*$_1$, this graph contains 547 nodes and $1,552$ edges.

Table 5.1 shows the most relevant nodes and their centrality measure in this network. For simplicity, I measure the centrality of the network using an undirected graph. As expected, the entity "Diabetes_mellitus" is the most important node in this network, having high centrality measures for degree, closeness, and betweeness. The other relevant nodes represent concepts related to this entity, such as *glucose, insulin, and dysglicemia.* In addition, the entity *patients* is the second most important with high closeness and betweeness with other nodes. This entity is common among publications containing research related to people having this disease.

Table 5.1: Entities with Centrality Measures from Network of Pubmed_1

Entity	***Degree***	***Closeness***	***Betweeness***
Diabetes_mellitus	0.238	0.496	0.316
Patients	0.163	0.434	0.154
Glucose	0.082	0.389	0.088
Insulin	0.055	0.385	0.028
Dysglycemia	0.049	0.355	0.008

A network contains all the entities and their relationships found in the models. Therefore, I retrieve semantic relationships from a network of data models. In particular, Networkx allows us querying semantic relationships containing the same entity among all models representing publications.

Figure 5.6 shows some relationships including the entity "Diabetes_mellitus" in four data models from publications. For instance, this framework uses the sentence *"The low impact of genetics on metabolic diseases is further reinforced by the growing incidence of diabetes and obesity over the last decades"* [134] to generate the relationships *genetics is reinforced diabetes* and *genetics is reinforced obesity*. The two metioned relationships appear in this figure on the second column. The relationships include information from the publications a) "Early Identification of Type 2 Diabetes" [135], b) "Gut microbiota and diabetes: from pathogenesis to therapeutic perspective" [134], c) Ionizing radiation and aging: rejuvenating an old idea [136], and d) "Clinical characteristics of patients with type 2 diabetes mellitus at the time of insulin initiation: INSTIGATE observational study in Spain" [137]. The latter is a study from Spain, which shows the usefulness of this approach to analyze international research in English.

For the second part of this experiment, I present results from the creation of a network with dataset Pubmed_2. The resulting graph contains $10,167$ nodes with $29,762$ edges.

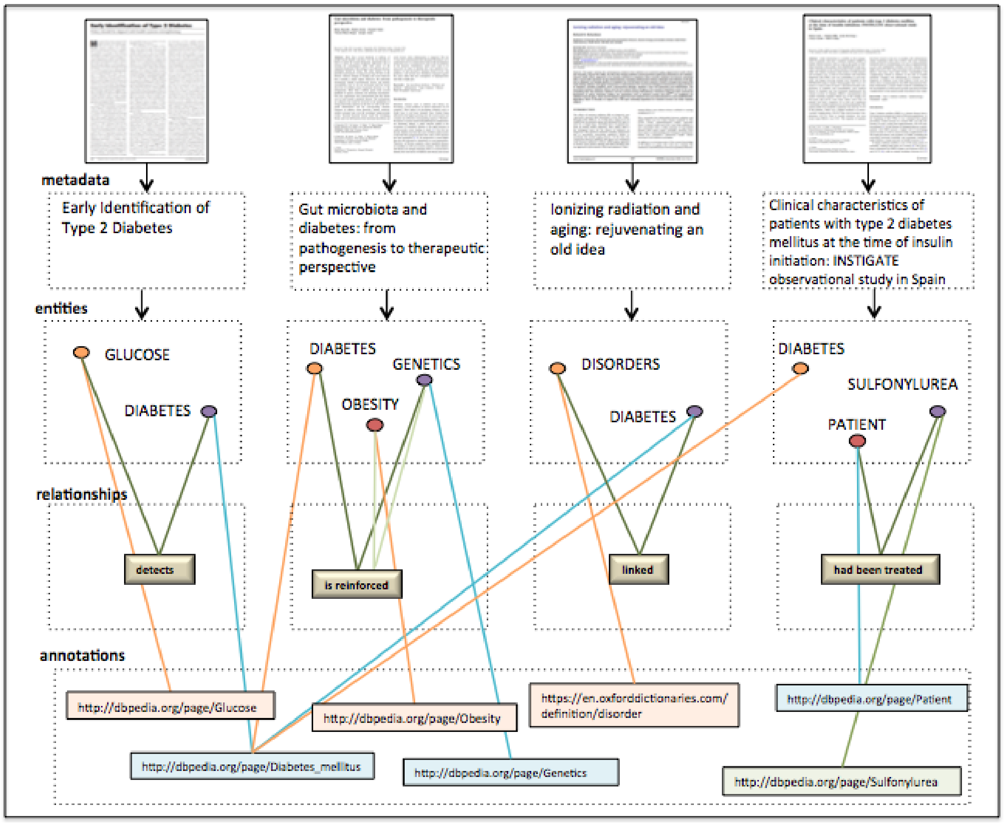

metadata
Early Identification of Type 2 Diabetes
Gut microbiota and diabetes: from pathogenesis to therapeutic perspective
Ionizing radiation and aging: rejuvenating an old idea
Clinical characteristics of patients with type 2 diabetes mellitus at the time of insulin initiation: INSTIGATE observational study in Spain
entities
GLUCOSE
DIABETES
DIABETES
GENETICS
OBESITY
DISORDERS
DIABETES
DIABETES
SULFONYLUREA
PATIENT
relationships
detects
is reinforced
linked
had been treated
annotations
http://dbpedia.org/page/Glucose
http://dbpedia.org/page/Obesity
https://en.oxforddictionaries.com/definition/disorder
http://dbpedia.org/page/Patient
http://dbpedia.org/page/Diabetes_mellitus
http://dbpedia.org/page/Genetics
http://dbpedia.org/page/Sulfonylurea

Figure 5.6: Data Models Using Pubmed$_1$

Table 5.2 shows the ten most relevant entities and their centrality measures. In particular, I report their degree, closeness, and betweeness. For simplicity, this analysis uses an undirected graph.

From this network, the first four entities of the table, such as *protein, cell, gene,* and *treatment* have similar behavior for degree, closeness, and betweenness centrality. Even though the entity *cancer* has less connections than the entities *diabetes, ghrelin, glaucoma,* and *neurons*, the entity *cancer* is closer to other nodes because its closeness measure is greater than the rest. On the other hand, the entity *glaucoma* slightly participates in more paths than *diabetes, ghrelin, neurons,* and *autophagy.* Comparing this network with the one from the previous experiment, the centrality

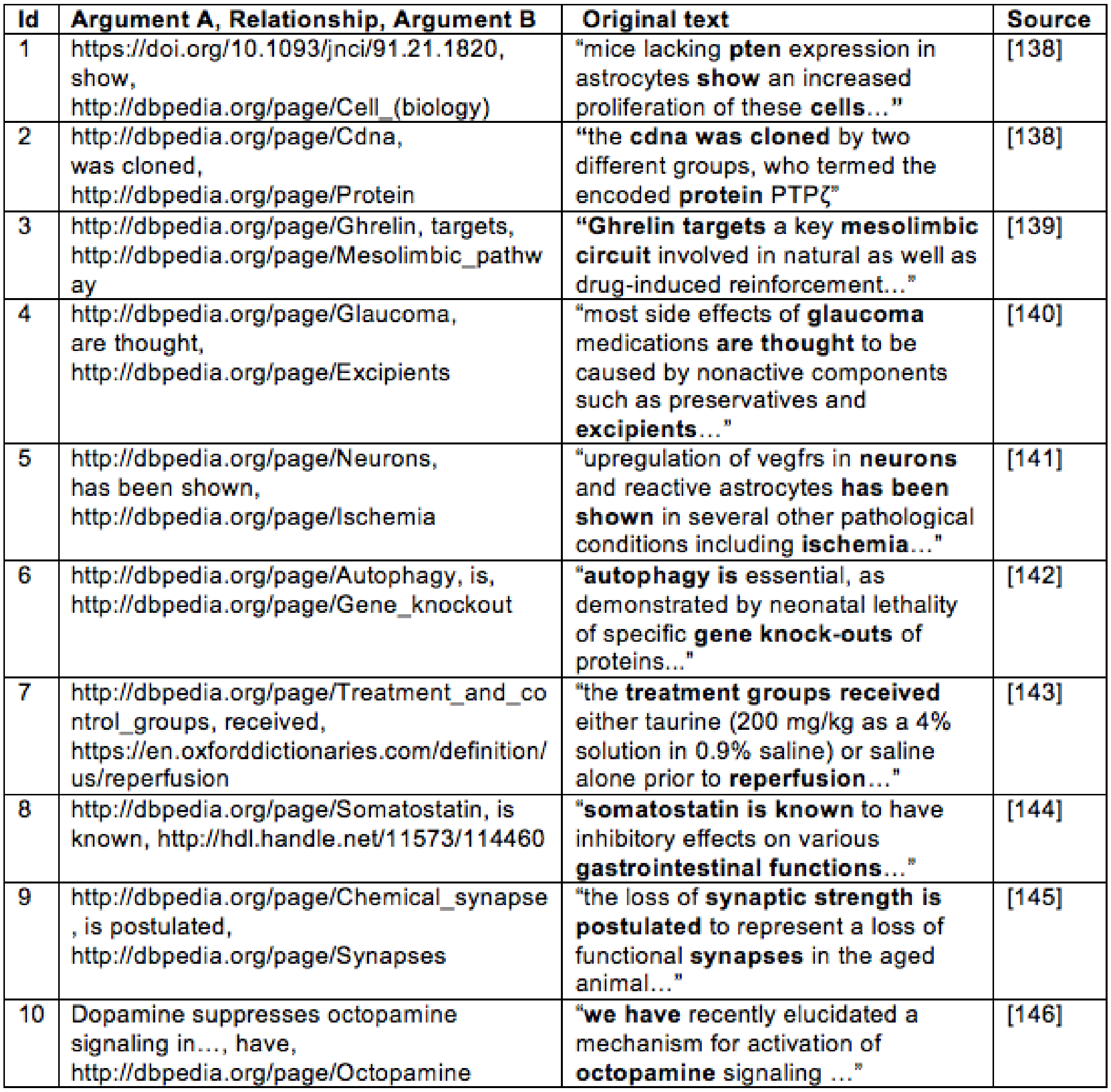
Table 5.2: Entities with Centrality Measures from Network of Pubmed_2

Entity	***Degree***	***Closeness***	***Betweeness***
Protein	0.033	0.2833	0.0294
Cell	0.029	0.2830	0.0197
Gene	0.022	0.2784	0.0160
Treatment	0.020	0.2716	0.0103
Diabetes	0.015	0.2620	0.0065
Ghrelin	0.014	0.2537	0.0058
Glaucoma	0.014	0.2560	0.0070
Neurons	0.012	0.2575	0.0060
Cancer	0.011	0.2669	0.0040
Autophagy	0.010	0.2562	0.0033

measures are smaler given the size of this network.

Using some relevant entities, Figure 5.7 shows relationships discovered among publications. The relationships are found in “Protein tyrosine phosphatases in glioma biology” [138], “Glutamatergic regulation of ghrelin-induced activation of the mesolimbic dopamine system” [139], “’Randomized trial of brinzolamide/brimonidine versus brinzolamide plus brimonidine for open-angle glaucoma or ocular hypertension” [140], “Cellular distribution of vascular endothelial growth factor A (VEGFA) and B (VEGFB) and VEGF receptors 1 and 2 in focal cortical dysplasia type IIB” [141], “A ’radical’ mitochondrial view of autophagy-related pathology” [142], and “Taurine protects against lung damage following limb ischemia reperfusion in the rat by attenuating endoplasmic reticulum stress-induced apoptosis” [143].

From the first relationship, for the entity “PTEN” this framework finds the annotation https://doi.org/10.1093/jnci/91.21.1820 , which talks about PTEN Gene and Integrin Signaling in Cancer. The remaining relationships annotate entities using DBpedia, such as *Ischemia*. The cartesian product of entities in this approach allows us to get relationships, therefore, these examples are not exhaustive. For instance, the text from relationship four originates additional relationships involving



Id	Argument A, Relationship, Argument B	Original text	Source
1	https://doi.org/10.1093/jnci/91.21.1820, show, http://dbpedia.org/page/Cell_(biology)	"mice lacking **pten** expression in astrocytes **show** an increased proliferation of these **cells**…"	[138]
2	http://dbpedia.org/page/Cdna, was cloned, http://dbpedia.org/page/Protein	"the **cdna was cloned** by two different groups, who termed the encoded **protein** PTPζ"	[138]
3	http://dbpedia.org/page/Ghrelin, targets, http://dbpedia.org/page/Mesolimbic_pathway	"**Ghrelin targets** a key **mesolimbic circuit** involved in natural as well as drug-induced reinforcement…"	[139]
4	http://dbpedia.org/page/Glaucoma, are thought, http://dbpedia.org/page/Excipients	"most side effects of **glaucoma** medications **are thought** to be caused by nonactive components such as preservatives and **excipients**…"	[140]
5	http://dbpedia.org/page/Neurons, has been shown, http://dbpedia.org/page/Ischemia	"upregulation of vegfrs in **neurons** and reactive astrocytes **has been shown** in several other pathological conditions including **ischemia**…"	[141]
6	http://dbpedia.org/page/Autophagy, is, http://dbpedia.org/page/Gene_knockout	"**autophagy is** essential, as demonstrated by neonatal lethality of specific **gene knock-outs** of proteins..."	[142]
7	http://dbpedia.org/page/Treatment_and_control_groups, received, https://en.oxforddictionaries.com/definition/us/reperfusion	"the **treatment groups received** either taurine (200 mg/kg as a 4% solution in 0.9% saline) or saline alone prior to **reperfusion**…"	[143]
8	http://dbpedia.org/page/Somatostatin, is known, http://hdl.handle.net/11573/114460	"**somatostatin is known** to have inhibitory effects on various **gastrointestinal functions**…"	[144]
9	http://dbpedia.org/page/Chemical_synapse, is postulated, http://dbpedia.org/page/Synapses	"the loss of **synaptic strength is postulated** to represent a loss of functional **synapses** in the aged animal…"	[145]
10	Dopamine suppresses octopamine signaling in…, have, http://dbpedia.org/page/Octopamine	"**we have** recently elucidated a mechanism for activation of **octopamine** signaling …"	[146]

Figure 5.7: Relationships of Entities in Pubmed_2

the entity *preservative*. This framework also finds concepts composed with more than one word, for instance the entities *Mesolimbic_pathway* and *Gene_knockout*. Finally, the Oxford dictionary helps annotating the entity *reperfusion*.

In addition, a network allows us to query other entities, such as somatostatin, octopamine, and synapses. The last three relationships are extracted from the publications “Alterations in Somatostatin Cells and Biochemical Parameters Following Zinc Supplementation in Gastrointestinal Tissue of St reptozotocin-Induced Diabetic Rat” [144], “Blueberry-enriched diet ameliorates age-related declines in NMDA receptor-dependent LTP” [145], and “Dopamine suppresses octopamine signaling in C. elegans: possible involvement of dopamine in the regulation of lifespan” [146].

From this set of relationships, for the ninth relationship the framework annotates the entity *gastrointestinal functions* with a document called “Somatostatin and the gastrointestinal tract” with URL `http://hdl.handle.net/11573/114460`. In addition, the last relationship contains the name of a publication as argument A. This relationship uses the co-reference resolution for the pronoun “we”. Then, this framework replaces the reference of the authors with the name of their publication. Therefore, this framework detects publications as entities that interact with concepts.

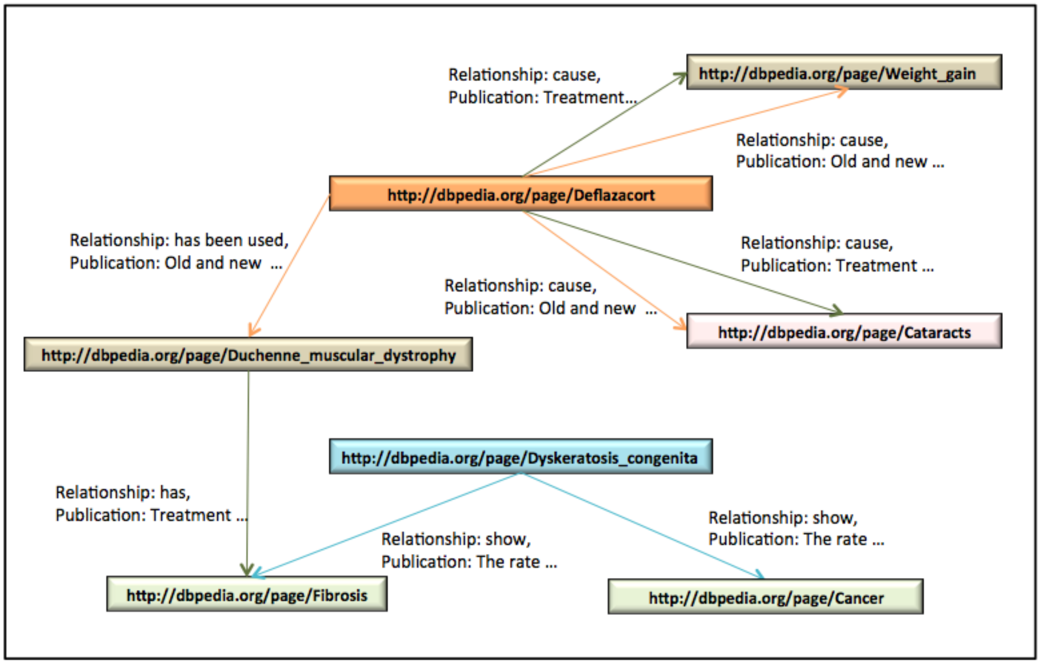

Relationship: cause,
Publication: Treatment...
http://dbpedia.org/page/Weight_gain
Relationship: cause,
Publication: Old and new ...
http://dbpedia.org/page/Deflazacort
Relationship: has been used,
Publication: Old and new ...
Relationship: cause,
Publication: Treatment ...
Relationship: cause,
Publication: Old and new ...
http://dbpedia.org/page/Cataracts
http://dbpedia.org/page/Duchenne_muscular_dystrophy
http://dbpedia.org/page/Dyskeratosis_congenita
Relationship: has,
Publication: Treatment ...
Relationship: show,
Publication: The rate ...
Relationship: show,
Publication: The rate ...
http://dbpedia.org/page/Fibrosis
http://dbpedia.org/page/Cancer

Figure 5.8: Relationships of Entities in Three Models Using Pubmed$_2$

For demonstration purposes, I present a graphic representation of a set of entities and their relationships from three models. The publications include "The rate of leukocyte telomere shortening predicts mortality from cardiovascular disease in elderly men" [147], "Treatment of dystrophinopathic cardiomyopathy: review of the literature and personal results" [148], and "Old and new therapeutic developments in steroid treatment in Duchenne muscular dystrophy" [149]. In particular, Figure 5.8 shows examples of relationships sharing some entities, such as *weight gain, cataracts, deflazacort, fibrosis, duchene muscular distrophy, dyskeratosis congenita,* and *cancer.* Each edge contains annotations regarding the name of a relationship and name of a publication. For space constrains, the edges do not include the original text where a relationship is found.

This figure shows how the entities can help discover additional information. For instance, the shortest path from one entity to another, as shown from *deflazacort* to *fibrosis*, using as a bridge the entity *duchene muscular distrophy.* In addition, it shows how two different models contain the same relationships between two entities, such as *deflazacort causes weight gain* and *deflazacort causes cataracts.* The relationships are extracted from the original text of each publication. It is important to refer to the model containing the text where a relationship is extracted. For instance, these relationships are found in the original text *"Deflazacort appears to cause less weight gain and less bone mass deterioration, but more often it is associated with the development of asymptomatic cataracts"* [148]. The sentence gives more information about the relationship, indicating that "Deflazacort causes less weight gain" and that "Deflazacort is related to the development of asymptomatic cataracts." The two publications [148] and [149] share common entities and relationships. Therefore, the network helps researchers finding, comparing, and contrasting entities and relationships, and the annotations provide more specific information for further analysis.

Even though each publication is represented with a semantic data model that contains disambiguated entities, adding up the number of entities per model might contain redundant entities. It is at the moment of creating a network when the unique entities in a set of models are discovered. Table 5.3 shows a summary of the characteristics of the models generated from the datasets analyzed. In particular, it shows the number of entities found in the models separately, the unique entities composing a network, and the total number of relationships.

Table 5.3: Entities and Relationships from Datasets

Dataset	***Entities in Models***	***Entities in Network***	***Relationships***
Pubmed$_1$	857	547	1,552
Pubmed$_2$	26,691	10,167	29,762

The organization of the data model facilitates finding relationships among entities and later among models. The relationships can extend to several data models, as long as they contain common entities. The number of relationships from each model varies because they depend on the number of entities composed by the keywords representing a context and concepts from tables. Therefore, if a publication lacks tables with relevant information, the model may lack concrete relationships. In contrast, a rich publication with tables can produce more valuable data models.

Some publications contain direct relationships in natural language and others hide relationships in the richness and redundance of the language. It is important using clear and simple language in science. Using my approach, I demonstrate how relevant entities from different publications can be related to each other. Therefore, it is crucial to normalize name(s) of concepts within a discipline, that is, unify vocabulary. In addition, these relationships use mainly entities from the ontology DBpedia.

Experiment Generation of Semantic Data Models

For the second experiment, I measure the average time to generate semantic data models using *Pubmed*$_2$. In particular, I obtain the time of extracting metadata and context; detecting tabular structures and extracting tables' information; disambiguating and enriching table concepts, that is, entities; extracting semantic relationships; and finally organizing information to compose the models in the dataset. From this experiment, I report the average time of the automated generation of a semantic data model.

The average time to generate the components of semantic data models for the dataset *Pubmed*$_2$, including the approach to find relationships from text, as well as entities not defined in a context or table. The results show that this approach takes about 24 minutes in average to generate each model. The generation of data models is feasible to reduce the time that a researcher spends analyzing an article, to analyze a large number of publications, and more importantly, to gain rapid insight to scientific documents and determine if they need further consultation.

In summary, I found structured semantic relationships that represent insights, as well as specific content of a publication. In particular, I demonstrated the creation of networks of semantic data models that represent syntheses of publications. I also found relevant concepts with their semantic relationships among publications using a network of models. Finally, I computed the average time to automatically generate semantically enriched data models.

Chapter 6

Conclusion of Dissertation

To solve problems related to analyzing a large quantity of scientific publications, which lack 1) a unified vocabulary, 2) a standard format in tables, and 3) semantic interoperability, especially those in PDF format. This investigation explored automated methods to create semantic data models in any domain without knowing a priori their components: entities and relationships from scientific publications. In addition, this investigation evaluated the feasibility of automated methods to accomplish my thesis statement and to facilitate the retrieval, integration, comparison, sharing, and interoperability among publications. In particular, I developed a conceptual data model to discover and represent the elements of semantic data models from publications and built a framework to create the semantically enhanced data models. The entities with their semantic relationships, extracted from text and tabular information embedded in publications, composed the models, as well as information from general ontologies and vocabularies.

To develop this investigation, this document presented five chapters. One chapter reviewed related literature, and four chapters described the methods developed in this study and main contributions. First, I investigated how to detect, extract, and represent information presented in tables within digital publications. Second, I interpreted the data in the tables for each document to identify semantic relationships and entities in order to synthesize the information. Third, I used relevant information contained in publications, such as metadata, entities, and relationships to characterize the semantic data model, which allowed us to manage and analyze information obtained from the digital publications.

The second chapter, "Literature Review", presented definitions and related work in the areas of table identification and table interpretation, as well as in the areas of entity disambiguation and extraction of semantic relationships. Finally, this chapter reviewed data integration methods, semantic data models, tools to provide semantic annotations, summarization methods, and comparison approaches among publications.

The third chapter, "Table Detection and Information Extraction", was based on the article, *TAO: System for Table Detection and Extraction from PDF Documents*. I developed methods to automatically detect and organize tables from PDF documents, and the prototype system TAble Organization (TAO) to demonstrate the methods' accuracy. TAO applied PDFMiner to convert the PDF documents into XML format. Even though the PDFMiner's output in XML is large, the information of the original PDF document is preserved and enriched. I developed methods combining layout heuristics and supervised learning methods to process the XML, detecting the location of the tables within the document and extracting the corresponding table cells.

TAO was implemented using three main processes *Document Conversion*, *Table Detection* and *Table Extraction*, generating the output of the system in a JSON document, which allows interoperability and information exchange. The JSON document contains the representation of the tables cells within the document. The output is also enriched with information not displayed in the PDF document, such as coordinates of the table cells. This extra information allows structural analysis of a table. The output document can easily be stored and managed in a database to facilitate analysis, sharing and collaboration.

TAO was evaluated on a variety of PDF documents with a variety of tables, and I compared its performance to a baseline from related work. TAO overcame related work limitations and performed satisfactorily on scientific and non-scientific documents, on PDF documents with large and small number of pages, with single and double columns, and with various tabular formats.

The fourth chapter, "Table Interpretation to Synthesize Documents", based on the article *Table Interpretation and Extraction of Semantic Relationships to Synthesize Digital Documents*, presented a framework for this investigation. For their simplicity to present structured content and to organize information, I used tables in digital publications to recover summarized information and structural relationships among conceptual entities in these documents. In addition, I used unstructured text in publications to find the context of a document and semantic relationships of entities.

I developed a framework with functional and semantic analyses to understand a table's content and its publication. this framework contains four steps: 1) metadata and context detection, 2) entity recognition from tables, and annotation and disambiguation using external sources of knowledge 3) discovery of relationships, and 4) organization of the metadata, entities, and semantic relationships per publication. The context and metadata helped to preserve the information to identify the source publication containing tables. Using the semantic analysis, I performed entity recognition from tables embedded in a publication. In addition, the entities were enriched using annotations. Finally, the entities and functional analysis helped finding relationships from tables and text.

This approach used statistical methods, Natural Language Processing, and unsupervised learning to extract important information, such as context, entities with annotations, and semantic relationships. This information characterizes each publication as a synthesis of each document. Because each synthesis contains relevant information, it facilitates to consult and analyze publications promptly, and to interoperate between publications.

I demonstrated my approach with a working example and a quantitative assessment. The method worked for PDF and XML formats, including various layouts. In addition, the general ontologies and vocabulary representing the findings allowed us to account for publications among different domains. I obtained a promising number of entities and semantic relationships for each document. this framework organized relevant information to generate a publication's synthesis, which allows us consulting a publication promptly.

The fifth chapter, "Automated Generation of Semantic Data Models", formally defined the components of the model characterization. These models contain relevant information that can be used to understand tables and summarize a publication. To develop the models, I primarily used the output from the previous analyses: Table Identification and Interpretation, as well as Information Extraction of Context, Metadata, and Semantic Relationships.

I developed a comprehensive framework to automatically generate semantic data models from digital documents. The data models contained relevant entities from tabular layouts and the context of a document; the entities were semantically enriched with annotations and extracted relationships.

The organization of semantic data models using a vocabulary and general ontologies enabled us to synthesize and analyze documents from diverse areas of knowledge. In particular, I used defined relationships to represent association, classification, aggregation, and generalization. The discovery of well-defined relationships in the models allowed researchers to find structured results, experimental settings, and hierarchical associations among concepts in scientific documents.

The representation of metadata, entities and semantic relationships were the common components of the models in a flexible format to facilitate interoperability. To show this interoperability, I created a semantic network using a set of data models derived from publications of different topics and another with publications with similar content. The networks of publications served to find semantic relationships containing similar entities or context among publications. I used centrality measures to detect the most relevant nodes in the networks.

The experiments yielded structured semantic relationships, which allowed us to compare and contrast information from different digital publications. In addition, I could verify the provenance of the information obtained with this approach. Given the proliferation of websites and possible documents with fake information, the detection of semantic relationships matched with metadata to identify the provenance of a digital document made the data models more reliable. I also measured the time to generate the data models. The approach was capable and feasible to analyze documents faster than a person could do it manually.

In summary, to investigate the viability of the thesis statement, I developed automatic methods to analyze tables in digital publications, to detect and extract semantic information, and to gain knowledge about the documents examined. Finally, this document presented the results of this investigation, and the contributions to the current literature in the areas of Information Extraction, Information Modeling, and Semantic Analytics.

6.1 Future Work

I plan to develop tools for efficient Data Analytics to better exploit semantic data models. For instance, these tools can be used to determine the uncertainty of the semantic relationships detected in the models. In addition, I aim to study Data Reasoning while considering the causality of the relationships that are identified by using the proposed models.

Although this framework identified data type for table cells, some of the cells may contain values that require normalization. For instance, the entity *distance* can contain values using different measurement units, such as feet versus meters. Therefore, I intend to continue my research on Data Quality to support semantic normalization to map the difference between the representations of entities' attributes.

For this dissertation, the semantic analysis focused on documents written in English. The multicultural scientific community requires access to information published in many languages. I envision that these methods could perform semantic analysis in other languages with a few modifications.

In addition to tabular structures and free text, I plan to analyze datasets related to digital publications. At times, datasets published as supplementary material are under-utilized because they lack easy accessibility, and they have the same variations (data, format, and layout) as the information presented in tables.

This investigation has also two lines of research regarding sources of knowledge, one is to develop an ontology of relationships commonly used in scientific documents from any domain, and the other is to create and use unified vocabularies for scientific areas to better manage information.

Finally, I plan to use this project to create clusters of facts that are related in diverse disciplines. In addition, this project can perform the following: 1) locate people who may be interested in working together in the same area of interest, 2) locate research already done to avoid repeating the same path, and 3) help researchers select unexplored areas to investigate.

6.2 Lessons Learned

Many publications already define metadata regardless of the document's format; however, some publications have no metadata at all. As scientists, establishing metadata in publications facilitates the dissemination of that work. In the same way, the guidelines for all scientific conferences and journals should require mandatory metadata. These conferences and journals have the authority to mandate the inclusion of metadata, and their review processes can control that requirement.

Regarding metadata for tables, it is necessary a table definition in the schema.org vocabulary, including definitions for table title, table cell, and specific definitions to identify a table column header, and table data cells. Useful tags to describe data and header cells would be useful. In addition, the PDF format could include tags for table identification; these tags can be the same ones used on the Web to avoid having multiple tag names for the same element. The schema.org vocabulary also lacks a definition for semantic relationships. A definition of the class semantic relationship could assist in representing the two different arguments and its relationship in a linked document.

Information extraction and semantic problems could be avoided if the scientific community establishes formal semantic publishing methods in any format and domain. Using defined rules while presenting results could decrease the time required to detect, extract, and integrate information; and most importantly, discover knowledge. In addition to the preferred narrative method to present research, I envision each scientific domain to define rules to include metadata, methods and results for scientific publications. These rules can be expressed in simple tabular structures.

Commercial search engines have the power to control how information is explored. Using the search engines frequently increases the cost, and these engines may promote products or services during a search. Many times, researchers depend on the output of the search engines for academic interest. From this issue, one of the areas where scientists can join efforts is to improve access to academic search engines, which are dedicated to research purposes exclusively.

The problems exposed in this work point to the representation and management of digital publications from any domain. The Internet of Things is advancing rapidly, and scientific publications have barely been prepared to the rapid interaction of information among disciplines required at this time. The dissemination of scientific work depends on digital libraries and mainly on ourselves, who can be responsible to include metadata in all our publications.

The normalization of tables using semantic annotations can avoid adding noise on the design of a simple tabular representation. Then, researchers can obtain more direct information from these valuable sources of knowledge. Finally, the inclusion of publishing rules to report results and other elements could facilitate the analysis of scientific articles, as well as the creation of a scientific Semantic Web.

Appendices

Appendix A

Conceptual Design of a Semantic Data Model

The components of a semantic data model include a) metadata and context, b) cells and categorical entities with annotations, and c) semantic relationships.

A.1 Metadata and Context

The first elements in the semantic data model are *metadata* and *context*, which help describe a publication. Metadata is crucial in the creation of a data model because it not only allows us to identify each model, but also to relate the entities and semantic relationships to their original documents.

To depict the components of the model, I use the entity-relationship (ER) model as described by R. Elmasri and S. Navathe in their book *Fundamentals of databases systems* [111]. The figures contain rectangles representing entities, diamonds representing relationships, and MIN and MAX values for cardinality shown in parentheses. The cardinality values indicate the minimum and maximum number of participants/arguments in a relationship, representing the structural constraint of the participation of an entity in a relationship.

Figure A.1 shows a conceptual design with the components of metadata in the semantic data models. Metadata represents a publication, which has a context as a set of five keywords in the approach, but this number may vary because I respect the keywords defined by authors. Each publication relates to a unique title, which can be identified with a unique DOI. One or more authors can write each title. If a first author has one or more collaborators, for each pair *first author-collaborator*, there is an isCoauthor relationship. Finally, each author has one email address. Each author can have one email address. To perform the extraction of these elements, refer to Chapter 4, Section 4.3.1 Discovery of Metadata and Context.

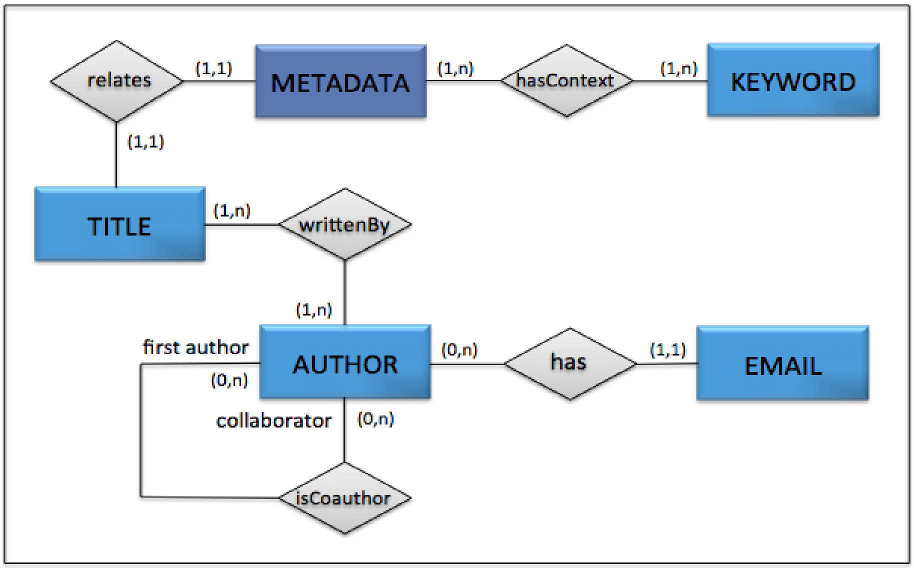


Figure A.1: Metadata Components

A.2 Categorical Entities

Figure A.2 shows the conceptual design of the table cells and categorical entities E. The set of cells contained in the columns and rows of a given table help discover entities and relationships. The relationship between cells is indicated by *hasAttribute*, where a **header cell** *has attribute* contained in a **data cell**. The cardinality of header to data cell indicates that a header can have zero to N data cells, while a data cell can have one header. To detect cells, I use the process explained in Chapter 3, Section 3.2.3 Table Extraction.

A set of entities E representing classes is the second element to collect for the creation of a semantic data model. The information in tabular structures allows us to find relevant concepts, that is, entities within a publication. Therefore, to detect entities, I use all cells with data type string. In addition, I include the keywords that represent a context in this set of entities E.

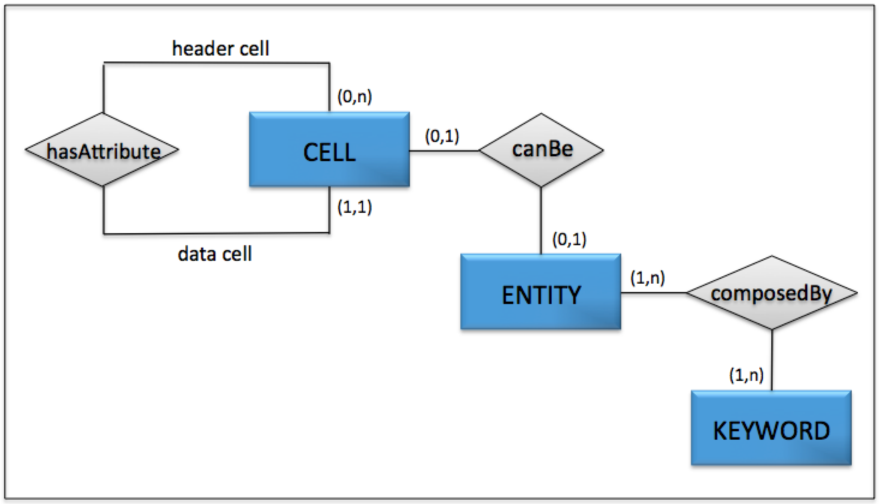


Figure A.2: Cells and Entities

The cardinality of the semantic relationship **cell** *canBe* **entity** indicates that an entity can exist in a cell's content. The cardinality of the semantic relationship

entity *composedBy* **keyword** indicates that E at least contains a keyword, and that a keyword may be a component of one or more entities as well. To detect entities, I use the process explained in Chapter 4, Section 4.3.2 Entity Recognition, Annotation, and Disambiguation.

A.3 Semantic Relationships

The last element of a semantic data model is a set of semantic relationships SR, which are mainly composed of structural and semantic relationships obtained from tables and text. Additional relationships are recovered from external sources of knowledge, such as a dictionary, a general ontology, and the Internet.

Figure A.3 shows the structural relationships from a table. They comprise table cells exclusively and use the relationship *hasAttribute*. A table has at least four cells (i.e., two rows and two columns) and a cell relates to a unique table.

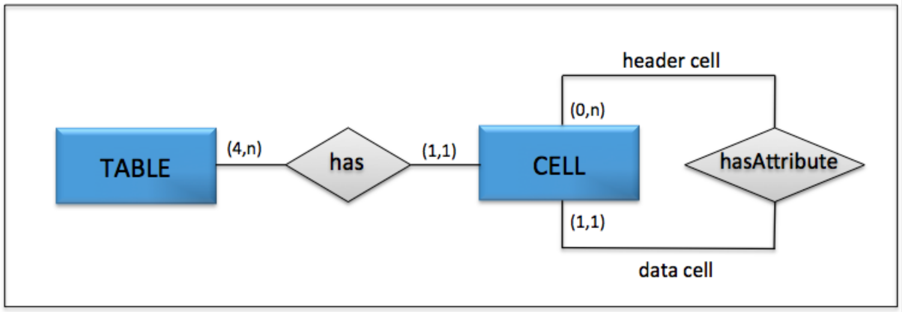


Figure A.3: Structural Relationships from Table Cells

On the other hand, the relationships derived from tables and text contain entities and semantic annotations. The annotations for unique entities contain information from DBpedia, the Oxford dictionary, or the Internet. Figure A.4 shows how an **entity** relates to its **semantic annotations**. For instance, a source knowledge can be an ontology's URI that contains the definition of an entity. The semantic relationships for these annotations include *defines* or *IsA*, and *describes*. If a URI exists in DBpedia, then an entity has a unique definition and URI. If the semantic annotation for a given entity is not found, I use the Oxford dictionary and the Internet to find the semantic relationship a **URL** *explains* an **entity**. Note that a URL can explain one or more entities, while an entity can have none or more URLs in the model. This figure also shows how an **entity** *associates* to another **entity** that could be recovered from a text phrase. An entity can be associated to zero or n entities.

The semantic relationship *associates* can have different labels describing relationships from text. Some of these labels are defined in SIO (see Appendix B), which contains the formal definitions for relationships. If the relationships are not defined in SIO, they are represented by verbs found in text. To detect semantic relationships, I use the processes explained in Chapter 4, Section 4.3.3 Discovery of Relationships.

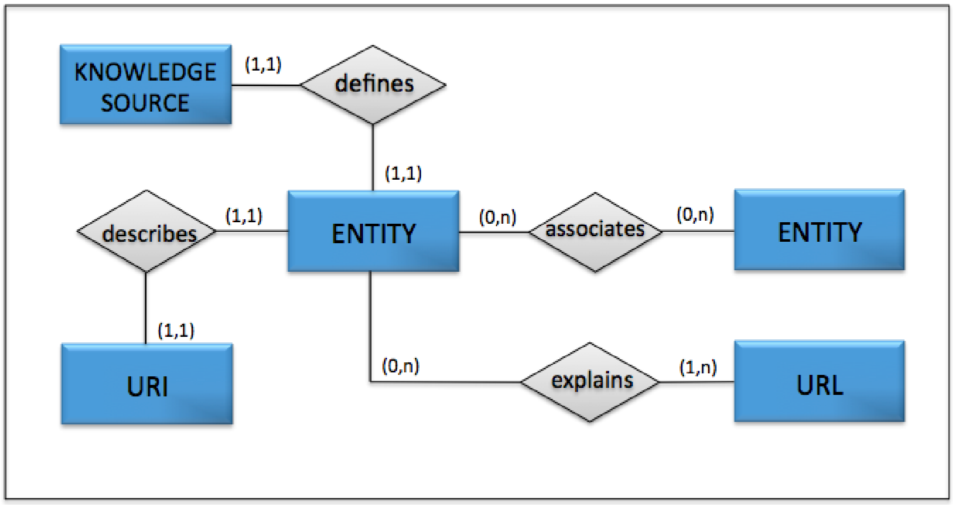


Figure A.4: Semantic Relationships from Tables and Text

Appendix B

Definitions from the SIO: Semanticscience Integrated Ontology

These definitions belong to the SemanticIntegrated Ontology, and I divide them into four sections: association, classification, aggregation, and generalization. The definitions contain original attributes from SIO [75]: identifier, comment, description, and label. In addition, some definitions contain type and properties (e.g., inverse, subproperty, equivalent, synonym).

ASSOCIATION

```
{
"@id": "http://semanticscience.org/resource/SIO_000001",
"@comment": "’is related to’ is the top level relation in SIO ",
"@description": "A is related to B iff there is some relation between A and B. ",
"@label": "is related to ",
"@type rdf:resource": "http://www.w3.org/2002/07/owl#SymmetricProperty"
}


{
"@id": "http://semanticscience.org/resource/SIO_000212 ",
"@description": "A is referred to by B iff B is an informational entity that makes reference to A.",
"@label": "is referred to by ",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000001"
}


{
"@id": "http://semanticscience.org/resource/SIO_000020",
"@description": "is a relation between an entity A and B that is a sign or indication of, or what specifically means",
"@label": "denotes",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000210",
}

{
"@id": "http://semanticscience.org/resource/SIO_000061",
"@description": "A is located in B iff the spatial region occupied by A is part of the spatial region occupied by B.",
"@equivalentTo": "OBO_REL:located_in",
"@label": "is located in",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000322",
"@type rdf:resource": "http://www.w3.org/2002/07/owl#TransitiveProperty"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000203",
"@description": "A is connected to B iff there exists a fiat, material or temporal path between A and B.",
"@label": "is connected to",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000322",
"@type rdf:resource": "http://www.w3.org/2002/07/owl#TransitiveProperty"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000628",
"@description": "refers to is a relation between one entity and the entity that it makes reference to.",
"@inverseOf": "http://semanticscience.org/resource/SIO_000212",
"@label": "refers to”,
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000001"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000736",
"@description": "is similar to is a relation between two entities that share one or more features.",
 "@label": "is comparable to”,
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000001",
"@type rdf:resource": "http://www.w3.org/2002/07/owl#SymmetricProperty"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000772",
"@description": "has evidence is a relation between a proposition and something that demonstrates the truth of the assertion.",
"@inverseOf": "http://semanticscience.org/resource/SIO_000773",
"@label": "has evidence",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000631"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000631",
"@description": "references is a relation between one entity and the entity that it makes reference to by name, but is not described by it.",
"@hasSynonym": "mentions",
"@label": "references”,
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000628"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000252",
"@description": "is reference for is a relation between a document that provides information about an entity.",
"@inverseOf": "http://semanticscience.org/resource/SIO_000631",
"@label": "is referenced by”,
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000212"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000233",
"@description": "is implementation of is a relation between an information entity and a specification that it conforms to.",
"@inverseOf": "http://semanticscience.org/resource/SIO_000234",
"@label": "is implementation",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000011"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000234",
"@description": "has implementation is a relation between a specification and an implementation that conforms to it.",
"@label": "has implementation",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000008"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000243",
"@description": "A transitive, symmetric, temporal relation in which one entity is causally related with another non-identical entity.",
"@label": "is causally related with",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000322",
"@type rdf:resource": "http://www.w3.org/2002/07/owl#TransitiveProperty"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000352",
"@description": "a is causally related from b iff there is a causal chain of events from b to a",
"@label": "is causally related from",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000243"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000641",
"@description": "has basis is a relation between a realizable entity and the quality that forms the basis for it.",
"@label": "has basis",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000008",
"@hasSynonym": "based on"
}
```

CLASSIFICATION

```
{
"@id": "http://semanticscience.org/resource/SIO_000008",
"@description": "has attribute is a relation that associates a entity with an attribute where an attribute is an intrinsic characteristic such as a quality, capability, disposition, function, or is an externally derived attribute determined from some descriptor (e.g.  a quantity, position, label/identifier) either directly or indirectly through generalization of entities of the same type.",
"@label": "has attribute",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000001"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000011",
"@description": "is attribute of is a relation that associates an attribute with an entity where an attribute is an intrinsic characteristic such as a quality, capability, disposition, function, or is an externally derived attribute determined from some descriptor (e.g. a quantity, position, label/identifier) either directly or indirectly through generalization of entities of the same type.",
"@inverseOf": "http://semanticscience.org/resource/SIO_000008",
"@label": "is attribute of",
"@subPropertyOf": " http://semanticscience.org/resource/SIO_000001"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000095",
"@description": "is member of is a mereological relation between a item and a collection.",
"@label": "is member of",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000011"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000300",
"@description": "A relation between a informational entity and its actual value (numeric, date, text, etc).",
"@label": "has value",
"@subset": "sadi"
}
```

AGGREGATION

```
{
"@id": "http://semanticscience.org/resource/SIO_000219",
"@description": "is source of is a relation between a source of information about some entity",
"@label": "is source of",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000011"
}

{
"@id": "http://semanticscience.org/resource/SIO_000068",
"@description": "is part of is a transitive, reflexive and anti-symmetric mereological relation between a whole and itself or a part and its whole.",
"@equivalentTo": "OBO_REL:part_of",
"@label": "is part of",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000061",
"@type rdf:resource": "http://www.w3.org/2002/07/owl#TransitiveProperty"
}

{
"@id": "http://semanticscience.org/resource/SIO_000093",
"@description": "is proper part of is an asymmetric, irreflexive (normally transitive) relation between a part and its distinct whole.",
"@equivalentTo": "OBO_REL:part_of",
"@label": "is proper part of",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000068",
"@type rdf:resource":"http://www.w3.org/2002/07/owl#AsymmetricProperty"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000230",
"@description": "has input is a relation between a process and an entity, where the entity is present at the beginning of the process.",
"@label": "has input",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000132",
"@subset": "sadi"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000231",
"@description": "is input in is a relation between an entity and a process, where the entity is present at the beginning of the process.",
"@inverseOf": "http://semanticscience.org/resource/SIO_000230",
"@label": "is input in",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000062",
"@subset": "sadi"
}
```

```
{
"@id": "http://semanticscience.org/resource/SIO_000232",
"@description": "is output of is a relation between an entity and a process, where the entity is present at the end of the process.",
"@label": "is output of",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000062",
"@subset": "sadi"
}
```

{
"@id": "http://semanticscience.org/resource/SIO_000244",
"@description": "A transitive temporal relation in which one entity was materially formed from another non-identical entity.",
"@label": "is derived from",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000352",
"@type rdf:resource": "http://www.w3.org/2002/07/owl#TransitiveProperty"
}

GENERALIZATION {
"@id": "http://semanticscience.org/resource/SIO_000062",
"@description": "is participant in is a relation that describes the participation of the subject in the (processual) object.",
"@inverseOf": "http://semanticscience.org/resource/SIO_000132",
"@label": "is participant in",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000322"
}

{
"@id": "http://semanticscience.org/resource/SIO_000132",
"@description": "has participant is a relation that describes the participation of the object in the (processual) subject.",
"@label": "has participant",
"@subPropertyOf": "http://semanticscience.org/resource/SIO_000322"
}

Appendix C

Context to Characterize a Semantic Data Model

```
"context":
{
     "headline":
      {
         "@id": "http://schema.org/ScholarlyArticle/headline",
         "@type": "@id"
      },
      "creator":
      {
         "@id": "http://schema.org/ScholarlyArticle/creator",
         "@type": "@id"
      },
      "keywords":
      {
         "@id": "http://schema.org/ScholarlyArticle/keywords",
         "@type": "@id"
      },
      "email": "http://schema.org/email",
      "sameAs":
      {
         "@id": "http://schema.org/ScholarlyArticle/sameAs",
         "@type": "@id"
      },
      "argumentA":
      {
         "@id": "http://semanticscience.org/resource/SIO_000000",
         "@type": "@id"
      },
      "argumentB":
      {
         "@id": "http://semanticscience.org/resource/SIO_000000",
         "@type": "@id"
      },
      "label": "http://semanticscience.org/resource/label",
      "sio_id":   {
         "@id": "http://semanticscience.org/resource/",
         "@type": "@id"
      },
      "URL": {
      "@id": "http://schema.org/url",
      "@type": "@id"
      },
      "URI":   {
         "@id": "http://dbpedia.org/page/",
         "@type": "@id"
      }
```

```
      "IsA":   {
         "@id": "http://dbpedia.org/page/",
         "@type": "dbo:abstract"
      }
}
```

References

[1] P. Larsen and M. Von Ins, “The rate of growth in scientific publication and the decline in coverage provided by science citation index,” *Scientometrics*, vol. 84, no. 3, pp. 575–603, 2010.

[2] E. Orduna-Malea, J. M. Ayllón, A. Martín-Martín, and E. D. López-Cózar, “Methods for estimating the size of google scholar,” *Scientometrics*, vol. 104, no. 3, pp. 931–949, 2015.

[3] M. Khabsa and C. L. Giles, “The number of scholarly documents on the public web,” *PloS one*, vol. 9, no. 5, p. e93949, 2014.

[4] D. Pieper and F. Summann, “Bielefeld academic search engine (base) an end-user oriented institutional repository search service,” *Library Hi Tech*, vol. 24, no. 4, pp. 614–619, 2006.

[5] W. L. Neuman, *Social research methods: Quantitative and qualitative approaches*, vol. 13. Allyn and bacon Boston, MA, 2005.

[6] J. Peckham and F. Maryanski, “Semantic data models,” *ACM Computing Surveys (CSUR)*, vol. 20, no. 3, pp. 153–189, 1988.

[7] H. P. NCBI, “Ncbi.nlm.nih.gov.” `https://www.ncbi.nlm.nih.gov/pubmed`. Accessed: October 01, 2017.

[8] A. E. Association, “American Economic Association: Journals, aeaweb.org.” `https://www.aeaweb.org/journals`. Accessed: October 01, 2017.

[9] IEEE, “Ieee.org.” `https://www.ieee.org/publications_standards/publications`. Accessed: October 01, 2017.

[10] T. S. W. S. Association, “Introduction — swsa.” `http://swsa.semanticweb.org/`. Accessed: October 01, 2017.

[11] AAAI, “Association for the advancement of artificial intelligence.” https://www.aaai.org. Accessed: October 21, 2017.

[12] M. O. Perez-Arriaga, S. Wilson, K. P. Williams, J. Schoeniger, R. L. Waymire, and A. J. Powell, “Omics metadata management system,” *Journal of Bioinformation*, vol. 2, no. 4, pp. 165–172, 2015.

[13] T. Bienz, R. Cohn, and A. Systems, *Portable document format reference manual.* Addison-Wesley Reading, MA, USA, 1993.

[14] D. Shotton, “Semantic publishing: the coming revolution in scientific journal publishing,” *Learned Publishing*, vol. 22, no. 2, pp. 85–94, 2009.

[15] T. Berners-Lee, W. Hall, J. A. Hendler, K. O’Hara, N. Shadbolt, and D. J. Weitzner, “A framework for web science,” *Foundations and trends in Web Science*, vol. 1, no. 1, pp. 1–130, 2006.

[16] T. Groza, S. Handschuh, K. Möller, and S. Decker, “Salt-semantically annotated LaTeX for scientific publications,” *The Semantic Web: Research and Applications*, pp. 518–532, 2007.

[17] F. Ronzano and H. Saggion, “Knowledge extraction and modeling from scientific publications,” in *International Workshop on Semantic, Analytics, Visualization*, pp. 11–25, Springer, 2016.

[18] J. L. Fink, S. Kushch, P. R. Williams, and P. E. Bourne, “Biolit: integrating biological literature with databases,” *Nucleic acids research*, vol. 36, no. suppl 2, pp. W385–W389, 2008.

[19] L. Penev, T. Erwin, J. Miller, V. Chavan, and C. Griswold, “Publication and dissemination of datasets in taxonomy: Zookeys working example,” *ZooKeys*, vol. 11, p. 1, 2009.

[20] ZooBank, “Zoobank.org.” http://www.zoobank.org. Accessed: October 01, 2017.

[21] E. of Life, “Eol.org.” http://www.eol.org. Accessed: October 01, 2017.

[22] H. Saggion and F. Ronzano, “Natural language processing for intelligent access to scientific information.,” in *COLING (Tutorials)*, pp. 9–13, 2016.

[23] A. Nenkova and K. McKeown, “A survey of text summarization techniques,” in *Mining Text Data*, pp. 43–76, Springer, 2012.

[24] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 538–543, ACM, 2002.

[25] A. P. Sheth, "Changing focus on interoperability in information systems: from system, syntax, structure to semantics," in *Interoperating geographic information systems*, pp. 5–29, Springer, 1999.

[26] S. Peroni, *The Semantic Publishing and Referencing Ontologies*, pp. 121–193. Cham: Springer International Publishing, 2014.

[27] T. Green, "We need publishing standards for datasets and data tables," *Learned publishing*, vol. 22, no. 4, pp. 325–327, 2009.

[28] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[29] M. O. Perez-Arriaga, T. Estrada, and S. Abad-Mota, "Tao: System for table detection and extraction from pdf documents," in *The 29th Florida Artificial Intelligence Research Society Conference*, pp. 591–596, AAAI, 2016.

[30] M. O. Perez-Arriaga, T. Estrada, and S. Abad-Mota, "Table interpretation and extraction of semantic relationships to synthesize digital documents," in *Proceedings of the 6th International Conference on Data Science, Technology and Applications*, pp. 223–232, 2017.

[31] M. F. Hurst, *The interpretation of tables in texts.* PhD thesis, University of Edinburgh, Scotland, 2000.

[32] X. Wang, *Tabular abstraction, editing, and formatting.* PhD thesis, University of Waterloo, Ontario, 1996.

[33] J. Hu, R. S. Kashi, D. P. Lopresti, and G. Wilfong, "Medium-independent table detection," in *Electronic Imaging*, pp. 291–302, International Society for Optics and Photonics, 1999.

[34] D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy, "Table-processing paradigms: a research survey," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 2-3, pp. 66–86, 2006.

[35] R. Zanibbi, D. Blostein, and J. R. Cordy, "A survey of table recognition," *Document Analysis and Recognition*, vol. 7, no. 1, pp. 1–16, 2004.

[36] S. Balakrishnan, A. Y. Halevy, B. Harb, H. Lee, J. Madhavan, A. Rostamizadeh, W. Shen, K. Wilder, F. Wu, and C. Yu, "Applying webtables in practice.," in *Conference on Innovative Data Systems Research (CIDR)*, 2015.

[37] T. Hassan and R. Baumgartner, "Table recognition and understanding from pdf files," in *Document Analysis and Recognition, 2007. Ninth International Conference on Document Analysis and Recognition.*, vol. 2, pp. 1143–1147, IEEE, 2007.

[38] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "Tableseer: automatic table metadata extraction and searching in digital libraries," in *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pp. 91–100, ACM, 2007.

[39] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "Improving the table boundary detection in pdfs by fixing the sequence error of the sparse lines," in *Document Analysis and Recognition, 2009. 10th International Conference on Document Analysis and Recognition.*, pp. 1006–1010, IEEE, 2009.

[40] T. Hassan, "Pdf to html conversion," tech. rep., University of Warwick, 2003.

[41] Y. Shinyama, "Pdfminer: Python pdf parser and analyzer," 2015.

[42] Y. Liu, P. Mitra, and C. L. Giles, "Identifying table boundaries in digital documents via sparse line detection," in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 1311–1320, ACM, 2008.

[43] E. Oro and M. Ruffolo, "Pdf-trex: An approach for recognizing and extracting tables from pdf documents," in *Document Analysis and Recognition, 2009. 10th International Conference on Document Analysis and Recognition.*, pp. 906–910, IEEE, 2009.

[44] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage pdf documents via visual seperators and tabular structures," in *Document Analysis and Recognition, 2011. 11th International Conference on Document Analysis and Recognition*, pp. 779–783, IEEE, 2011.

[45] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "A methodology for evaluating algorithms for table understanding in pdf documents," in *Proceedings of the 2012 ACM symposium on Document engineering*, pp. 45–48, ACM, 2012.

[46] M. J. Cafarella, A. Y. Halevy, Y. Zhang, D. Z. Wang, and E. Wu, "Uncovering the relational web.," in *11th International Workshop on the Web and Databases (WebDB)*, Citeseer, 2008.

[47] V. Mulwad, T. Finin, Z. Syed, and A. Joshi, "Using linked data to interpret tables.," in *Proceedings of the First International Conference on Consuming Linked Data (COLD)*, vol. 665, pp. 109–120, 2010.

[48] E. Oro and M. Ruffolo, "Xonto: An ontology-based system for semantic information extraction from pdf documents," in *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, vol. 1, pp. 118–125, IEEE, 2008.

[49] R. Rastan, H.-Y. Paik, and J. Shepherd, "Texus: A task-based approach for table extraction and understanding," in *Proceedings of the 2015 ACM Symposium on Document Engineering*, pp. 25–34, ACM, 2015.

[50] P. Venetis, A. Halevy, J. Madhavan, M. Paşca, W. Shen, F. Wu, G. Miao, and C. Wu, "Recovering semantics of tables on the web," in *Proceedings of the Very Large Database Endowment*, vol. 4, pp. 528–538, VLDB Endowment, 2011.

[51] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 25–26, Association for Computational Linguistics, 2007.

[52] K. Abdalgader and A. Skabar, "Short-text similarity measurement using word sense disambiguation and synonym expansion," in *Australasian Joint Conference on Artificial Intelligence*, pp. 435–444, Springer, 2010.

[53] P. University, "About WordNet, wordnet.princeton.edu." `https://wordnet.princeton.edu`. Accessed: October 01, 2017.

[54] 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.

[55] O. Medelyan, S. Manion, J. Broekstra, A. Divoli, A.-L. Huang, and I. Witten, "Constructing a focused taxonomy from a document collection," in *The Semantic Web: Semantics and Big Data* (P. Cimiano, O. Corcho, V. Presutti, L. Hollink, and S. Rudolph, eds.), vol. 7882 of *Lecture Notes in Computer Science*, pp. 367–381, Springer Berlin Heidelberg, 2013.

[56] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia-a crystallization point for the web of data," *Web Semantics: science, services and agents on the world wide web*, vol. 7, no. 3, pp. 154–165, 2009.

[57] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[58] V. C. Storey, “Understanding semantic relationships,” *”The International Journal on Very Large Data Bases (VLDB)”*, vol. 2, no. 4, pp. 455–488, 1993.

[59] A. Sheth, I. B. Arpinar, and V. Kashyap, “Relationships at the heart of semantic web: Modeling, discovering, and exploiting complex semantic relationships,” in *Enhancing the Power of the Internet*, pp. 63–94, Springer, 2004.

[60] D. Hristovski, C. Friedman, T. C. Rindflesch, and B. Peterlin, “Exploiting semantic relations for literature-based discovery.,” in *AMIA Annual Symposium Proceedings Archive*, 2006.

[61] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, “Open information extraction from the web,” *Communications of the ACM - Surviving the data deluge*, vol. 51, no. 12, pp. 68–74, 2008.

[62] M. Ruiz-Casado, E. Alfonseca, and P. Castells, “Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia,” in *International Conference on Application of Natural Language to Information Systems*, pp. 67–79, Springer, 2005.

[63] A. Moro and R. Navigli, “Integrating syntactic and semantic analysis into the open information extraction paradigm.,” in *International Joint Conference on Artificial Intelligence*, pp. 2148–2154, 2013.

[64] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, “Toward an architecture for never-ending language learning.,” in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, vol. 5, pp. 1306–1313, 2010.

[65] N. Nakashole, G. Weikum, and F. Suchanek, “Patty: A taxonomy of relational patterns with semantic types,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1135–1145, Association for Computational Linguistics, 2012.

[66] A. Fader, S. Soderland, and O. Etzioni, “Identifying relations for open information extraction,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1535–1545, Association for Computational Linguistics, 2011.

[67] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, “Open information extraction: The second generation.,” in *22nd International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 11, pp. 3–10, 2011.

[68] A. Doan, A. Halevy, and Z. Ives, *Principles of data integration.* Elsevier, 2012.

[69] C. H. Goh, S. Bressan, S. Madnick, and M. Siegel, “Context interchange: New features and formalisms for the intelligent integration of information,” *ACM Transactions on Information Systems (TOIS)*, vol. 17, no. 3, pp. 270–293, 1999.

[70] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data-the story so far,” *Semantic services, interoperability and web applications: emerging concepts*, pp. 205–227, 2009.

[71] R. Hull and R. King, “Semantic database modeling: Survey, applications, and research issues,” *ACM Computing Surveys (CSUR)*, vol. 19, no. 3, pp. 201–260, 1987.

[72] F. Steimann, “On the representation of roles in object-oriented and conceptual modelling,” *Data & Knowledge Engineering*, vol. 35, no. 1, pp. 83–106, 2000.

[73] E. Pafilis, S. I. O’Donoghue, L. J. Jensen, H. Horn, M. Kuhn, N. P. Brown, and R. Schneider, “Reflect: augmented browsing for the life scientist,” *Nature biotechnology*, vol. 27, no. 6, pp. 508–510, 2009.

[74] D. C. Comeau, R. I. Doğan, P. Ciccarese, K. B. Cohen, M. Krallinger, F. Leitner, Z. Lu, Y. Peng, F. Rinaldi, M. Torii, *et al.*, “Bioc: a minimalist approach to interoperability for biomedical text processing,” *Database*, vol. 2013, p. bat064, 2013.

[75] M. Dumontier, C. J. Baker, J. Baran, A. Callahan, L. L. Chepelev, J. Cruz-Toledo, R. Nicholas, D. Rio, G. Duck, L. I. Furlong, K. Nichealla, D. Klassen, J. P. McCusker, N. Queralt-Rosinach, M. Samwald, N. Villanueva-Rosales, M. D. Wilkinson, and R. Hoehndorf, “The semanticscience integrated ontology (sio) for biomedical research and knowledge discovery.,” *J. Biomedical Semantics*, vol. 5, pp. 1–11, 2014.

[76] D. M. schema.org, “Schema.org.” `http://schema.org/docs/datamodel.html`. Accessed: October 01, 2017.

[77] J. Piskorski and R. Yangarber, “Information extraction: past, present and future,” in *Multi-source, multilingual information extraction and summarization*, pp. 23–49, Springer, 2013.

[78] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: A brief survey," *arXiv preprint arXiv:1707.02268*, pp. 1–9, 2017.

[79] A. Nenkova, S. Maskey, and Y. Liu, "Automatic summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*, pp. 1–86, Association for Computational Linguistics, 2011.

[80] S. Teufel and M. Moens, "Summarizing scientific articles: experiments with relevance and rhetorical status," *Computational linguistics*, vol. 28, no. 4, pp. 409–445, 2002.

[81] E. Baralis, L. Cagliero, S. Jabeen, and A. Fiori, "Multi-document summarization exploiting frequent itemsets," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp. 782–786, ACM, 2012.

[82] C. Di Sciascio, L. Mayr, and E. Veas, "Exploring and summarizing document collections with multiple coordinated views," in *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics*, pp. 41–48, ACM, 2017.

[83] C. Zhang, *DeepDive: a data management system for automatic knowledge base construction.* PhD thesis, The University of Wisconsin-Madison, 2015.

[84] A. Constantin, S. Pettifer, and A. Voronkov, "Pdfx: fully-automated pdf-to-xml conversion of scientific literature," in *Proceedings of the 2013 ACM symposium on Document engineering*, pp. 177–180, ACM, 2013.

[85] S. Uddin, L. Hossain, and K. Rasmussen, "Network effects on scientific collaborations," *PloS one*, vol. 8, no. 2, p. e57546, 2013.

[86] L. Bornmann and R. Mutz, "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references," *Journal of the Association for Information Science and Technology*, vol. 66, no. 11, pp. 2215–2222, 2015.

[87] S. Kim, K. Han, S. Y. Kim, and Y. Liu, "Scientific table type classification in digital library," in *Proceedings of the 2012 ACM symposium on Document engineering*, pp. 133–136, ACM, 2012.

[88] D. Noonburg, "xpdf: A c++ library for accessing pdf," 2009.

[89] B. Litchfield, "Pdfbox," 2004.

[90] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernandez, M. Kay, J. Robie, and J. Siméon, "Xml path language (xpath)," *World Wide Web Consortium (W3C)*, 2003.

[91] M. O. Perez-Arriaga and T. P. Caudell, "A study of brain structure evolution in simple embodied neural agents using genetic algorithms and category theory," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pp. 2494–2500, IEEE, 2009.

[92] N. Shadbolt, W. Hall, and T. Berners-Lee, "The semantic web revisited," *Intelligent Systems, IEEE*, vol. 21, no. 3, pp. 96–101, 2006.

[93] R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 2, p. 10, 2009.

[94] O. E. Dictionary, "Oxford english dictionary online," *Mount Royal College Lib., Calgary*, vol. 14, 2004.

[95] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[96] M.-A. Aufaure, B. Le Grand, M. Soto, and N. Bennacer, "Metadata and ontology based semantic web mining," *Web semantics and ontology*, pp. 259–296, 2006.

[97] N. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," *Knowledge Systems Laboratory, Stanford University*, 2001.

[98] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, *et al.*, "The obo foundry: coordinated evolution of ontologies to support biomedical data integration," *Nature biotechnology*, vol. 25, no. 11, pp. 1251–1255, 2007.

[99] B. Smith, W. Ceusters, B. Klagges, J. Köhler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. L. Rector, and C. Rosse, "Relations in biomedical ontologies," *Genome biology*, vol. 6, no. 5, p. R46, 2005.

[100] R. Girju, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney, and D. Yuret, "Semeval-2007 task 04: Classification of semantic relations between nominals," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 13–18, Association for Computational Linguistics, 2007.

[101] S. wikipedia.org, "En.wikipedia.org." `https://en.wikipedia.org/wiki/Wikipedia:Statistics`. Accessed: October 01, 2017.

[102] N. Press, "Understanding metadata," *National Information Standards*, vol. 20, 2004.

[103] L. Moreau, "The foundations for provenance on the web," *Foundations and Trends in Web Science*, vol. 2, no. 23, pp. 99–241, 2010.

[104] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf, "Dublin core metadata for resource discovery," *Internet Engineering Task Force RFC*, vol. 2413, no. 222, p. 132, 1998.

[105] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[106] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pp. 63–70, Association for Computational Linguistics, 2002.

[107] S. Loria, "Textblob: simplified text processing," *Secondary TextBlob: Simplified Text Processing*, 2014.

[108] Z. Anthony and R. J. Price, "Document categorization using latent semantic indexing," in *Proceedings 2003 Symposium on Document Image Understanding Technology*, pp. 1–10, UMD, 2003.

[109] M. C. Services, "Web search." `https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api`. Accessed: October 30, 2017.

[110] M. Dahchour, A. Pirotte, and E. Zimányi, "Generic relationships in information modeling," in *Journal on Data Semantics IV*, pp. 1–34, Springer, 2005.

[111] R. Elmasri and S. B. Navathe, *Fundamentals of database systems.* Pearson, 2008. pp. 222–224.

[112] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Computational linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[113] Sio.semanticscience.org, "Sio.semanticscience.org." `http://sio.semanticscience.org`. Accessed: October 01, 2017.

[114] V. Hook, S. Bark, N. Gupta, M. Lortie, W. D. Lu, N. Bandeira, L. Funkelstein, J. Wegrzyn, D. T. OConnor, and P. Pevzner, "Neuropeptidomic components generated by proteomic functions in secretory vesicles for cell–cell communication," *The AAPS journal*, vol. 12, no. 4, pp. 635–645, 2010.

[115] DBpedia, "Dbpedia.org." `http://dbpedia.org`. Accessed: October 01, 2017.

[116] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[117] M. Ware and M. Mabe, "The stm report: An overview of scientific and scholarly journal publishing," 2015.

[118] A. H. Renear and C. L. Palmer, "Strategic reading, ontologies, and the future of scientific publishing," *Science*, vol. 325, no. 5942, pp. 828–832, 2009.

[119] T. K. Attwood, D. B. Kell, P. McDermott, J. Marsh, S. Pettifer, and D. Thorne, "Utopia documents: linking scholarly literature with research data," *Bioinformatics*, vol. 26, no. 18, pp. i568–i574, 2010.

[120] J. Hendler, "Science and the semantic web," *Science*, vol. 299, no. 5606, pp. 520–521, 2003.

[121] A. Ruttenberg, J. A. Rees, M. Samwald, and M. S. Marshall, "Life sciences on the semantic web: the neurocommons and beyond," *Briefings in bioinformatics*, vol. 10, no. 2, pp. 193–204, 2009.

[122] M. L. Brodie, "On the development of data models," in *On conceptual modelling*, pp. 19–47, Springer, New York, NY., 1984.

[123] S. D. Urban and L. M. Delcambre, "An analysis of the structural, dynamic, and temporal aspects of semantic data models," in *Second International Conference on Data Engineering, 1986 IEEE*, pp. 382–389, IEEE, 1986.

[124] G. Klyne and J. J. Carroll, "Resource description framework (rdf): Concepts and abstract syntax," 2006.

[125] D. Allemang and J. Hendler, *Semantic web for the working ontologist: effective modeling in RDFS and OWL.* Elsevier, 2011.

[126] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, and N. Lindström, "Json-ld 1.0," *W3C Recommendation (January 16, 2014)*, 2014.

[127] M. Lanthaler and C. Gütl, "On using json-ld to create evolvable restful services," in *Proceedings of the Third International Workshop on RESTful Design*, pp. 25–32, ACM, 2012.

[128] J.-L. Playground, "Loading the playground...." `http://json-ld.org/playground`. Accessed: October 01, 2017.

[129] J. Han, E. Haihong, G. Le, and J. Du, "Survey on nosql database," in *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pp. 363–366, IEEE, 2011.

[130] A. Spark, "Apache spark: Lightning-fast cluster computing," 2016.

[131] A. Hagberg, D. Schult, P. Swart, D. Conway, L. Séguin-Charbonneau, C. Ellison, B. Edwards, and J. Torrents, "Networkx. high productivity software for complex networks," *Webová strá nka https://networkx.lanl.gov/wiki*, 2013.

[132] A. G. Shoro and T. R. Soomro, "Big data analysis: Apache spark perspective," *Global Journal of Computer Science and Technology*, vol. 15, no. 1, 2015.

[133] K. Batool and M. A. Niazi, "Towards a methodology for validation of centrality measures in complex networks," *PloS one*, vol. 9, no. 4, p. e90283, 2014.

[134] R. Burcelin, M. Serino, C. Chabo, V. Blasco-Baque, and J. Amar, "Gut microbiota and diabetes: from pathogenesis to therapeutic perspective," *Acta diabetologica*, vol. 48, no. 4, pp. 257–273, 2011.

[135] K. V. Narayan, J. Chan, and V. Mohan, "Early identification of type 2 diabetes: policy should be aligned with health systems strengthening," vol. 34, no. 1, pp. 244–246, 2011.

[136] R. B. Richardson, "Ionizing radiation and aging: rejuvenating an old idea," *Aging (Albany NY)*, vol. 1, no. 11, pp. 887–902, 2009.

[137] M. Costi, T. Dilla, J. Reviriego, C. Castell, and A. Goday, "Clinical characteristics of patients with type 2 diabetes mellitus at the time of insulin initiation: Instigate observational study in spain," *Acta diabetologica*, vol. 47, no. 1, pp. 169–175, 2010.

[138] A. C. Navis, M. van den Eijnden, J. T. Schepens, R. H. van Huijsduijnen, P. Wesseling, and W. J. Hendriks, "Protein tyrosine phosphatases in glioma biology," *Acta neuropathologica*, vol. 119, no. 2, pp. 157–175, 2010.

[139] E. Jerlhag, E. Egecioglu, S. L. Dickson, and J. A. Engel, "Glutamatergic regulation of ghrelin-induced activation of the mesolimbic dopamine system," *Addiction biology*, vol. 16, no. 1, pp. 82–91, 2011.

[140] S. A. Gandolfi, J. Lim, A. C. Sanseau, J. C. P. Restrepo, and T. Hamacher, "Randomized trial of brinzolamide/brimonidine versus brinzolamide plus brimonidine for open-angle glaucoma or ocular hypertension," *Advances in therapy*, vol. 31, no. 12, pp. 1213–1227, 2014.

[141] K. Boer, D. Troost, W. G. Spliet, P. C. van Rijen, J. A. Gorter, and E. Aronica, "Cellular distribution of vascular endothelial growth factor a (vegfa) and b (vegfb) and vegf receptors 1 and 2 in focal cortical dysplasia type iib," *Acta neuropathologica*, vol. 115, no. 6, pp. 683–696, 2008.

[142] N. Raimundo and G. S. Shadel, "A radical mitochondrial view of autophagy-related pathology," *Aging*, vol. 1, no. 4, p. 354, 2009.

[143] X. Men, S. Han, J. Gao, G. Cao, L. Zhang, H. Yu, H. Lu, and J. Pu, "Taurine protects against lung damage following limb ischemia reperfusion in the rat by attenuating endoplasmic reticulum stress-induced apoptosis," *Acta orthopaedica*, vol. 81, no. 2, pp. 263–267, 2010.

[144] S. Bolkent, S. Bolkent, R. Yanardag, O. Mutlu, and S. Yildirim, "Alterations in somatostatin cells and biochemical parameters following zinc supplementation in gastrointestinal tissue of streptozotocin-induced diabetic rats," *Acta histochemica et cytochemica*, vol. 39, no. 1, pp. 9–15, 2006.

[145] S. J. Coultrap, P. C. Bickford, and M. D. Browning, "Blueberry-enriched diet ameliorates age-related declines in nmda receptor-dependent ltp," *Age*, vol. 30, no. 4, pp. 263–272, 2008.

[146] S. Suo, J. G. Culotti, and H. H. Van Tol, "Dopamine suppresses octopamine signaling in c. elegans: possible involvement of dopamine in the regulation of lifespan," *Aging (Albany NY)*, vol. 1, no. 10, p. 870, 2009.

[147] E. S. Epel, S. S. Merkin, R. Cawthon, E. H. Blackburn, N. E. Adler, M. J. Pletcher, and T. E. Seeman, "The rate of leukocyte telomere shortening predicts mortality from cardiovascular disease in elderly men," *Aging (Albany NY)*, vol. 1, no. 1, p. 81, 2009.

[148] L. Politano and G. Nigro, "Treatment of dystrophinopathic cardiomyopathy: review of the literature and personal results," *Acta Myologica*, vol. 31, no. 1, p. 24, 2012.

[149] C. Angelini and E. Peterle, "Old and new therapeutic developments in steroid treatment in duchenne muscular dystrophy," *Acta Myologica*, vol. 31, no. 1, p. 9, 2012.