

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences



**Design and Implementation of a Chatbot for Answering Questions on
Scientometric Indicators**

A thesis presented by

Víctor Iván López Rodríguez

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

Monterrey, Nuevo León, May, 2022

Dedication

To God. To my family and friends. In dedication to my loving parents, Victor and Gricelda, for being my inspiration and motivation. To my caring siblings, Max, Vanessa, and Melissa, for always being by my side.

Acknowledgements

First of all, I want to say thanks to my advisor, Dr. Héctor Ceballos, for his guidance and support during my graduate studies. In addition, I would like to thank the committee members for their valuable feedback and advice.

I want to thank my family for their motivation and support during this journey. Also, I want to thank my class friends for affronting this challenge together.

Finally, I want to express my gratitude to Tecnológico de Monterrey and CONACyT for their scholarship support during my graduate studies.

Design and Implementation of a Chatbot for Answering Questions on Scientometric Indicators

by

Víctor Iván López Rodríguez

Abstract

Scientometrics is the field of study and evaluation of scientific measures such as the impact of research papers and academic journals. It is an essential field because nowadays, different rankings use key indicators for university rankings, and universities themselves use them as Key Performance Indicators (KPI). The first objective of this research work is to propose a semantic model of scientometric indicators by generating a statistical ontology that extends Statistical Data and Metadata Exchange (SDMX). We develop a case study at Tecnológico de Monterrey following the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. We evaluate the benefits of storing and querying scientometric indicators using linked data in Neo4j to provide flexible and quick access knowledge representation that supports indicator retrieval, discovery, and composition based on a self-knowledge strategy. The semantic representation can answer a simple query using dimensions, query returning values with time intervals, aggregation functions such as average and standard deviation, and calculate a new scientometric indicator with data stored in the ontology.

The second objective of this research work is to integrate the proposed statistical ontology model of scientometric indicators in a chatbot. Building a chatbot requires the use of Natural Language Processing (NLP) as a capability for recognizing users' intent and extracting entities from users' questions. We proposed a method for recognizing the requested indicator and transforming the question expressed in natural language into a query to the semantic model. The chatbot and the ontology model represent a novel framework that can answer questions from the Research Office about scientometric indicators. The chatbot is evaluated in terms of Goal Completion Rate (GCR). It measures how many questions the chatbot answered correctly and correctly identifies intent and entity extraction. The second evaluation approach of the chatbot is a survey that focuses on usability, the strictness of language variations, chatbot comprehension, correlation in chatbot responses, and user satisfaction.

The main contribution of this research is the structural representation of the type of question that can be performed over the indicators modeled with SDMX. We simplify the model training and interpretation of questions by defining complexity levels and extracting entities from the question. We demonstrate how a chatbot can answer questions about any indicator modeled with SDMX. The chatbot can be trained to recognize another way to formulate questions without impacting the semantic representation of the indicators. The model is scalable because we can add more indicators using RDF, and the chatbot will only require minor changes (e.g., adding new dimensions).

Contents

Abstract	v
List of Figures	x
List of Tables	xi
1 Introduction	1
1.1 Problem Definition and Motivation	2
1.2 Hypothesis and Research Questions	4
1.3 Objectives	4
1.4 Contributions	5
1.5 Thesis Structure	5
2 Literature Review	6
2.1 Artificial Intelligence	6
2.2 Conversational Agents	7
2.3 Chatbot	8
2.4 Natural Language Processing	9
2.5 Ontology	11
2.6 SDMX	13
2.7 Linked Data Platforms	14
2.8 Scientometric Indicators	15
2.9 Chatbot Evaluation	15
2.10 Related Work	15
2.11 Summary	17
3 Chatbot Design	18
3.1 Proposal Solution	18
3.2 Methodology	20
3.3 Data Collection	20
3.3.1 Data Understanding	20
3.3.2 Data Preparation	20
3.4 RDF construction	23
3.4.1 Vocabularies and Namespace	24
3.4.2 Dataset	25
3.4.3 Data Structure Definitions (DSD)	25

3.4.4	Dimensions	27
3.4.5	Measures	27
3.4.6	Concept Scheme	29
3.4.7	Observations	31
3.5	Modeling	31
3.5.1	Neo4j Aura	32
3.6	Chatbot Construction and Integration	33
3.6.1	Chatbot Self Knowledge	33
3.6.2	Natural Language Processing	34
3.6.3	Natural Language transformation into Cypher Query	40
3.7	Data Flow between Chatbot and Ontology	42
3.8	Deployment	42
3.8.1	Chatbot Framework	42
3.8.2	Bot Framework Composer	44
3.8.3	Scientometric Indicator API	48
3.8.4	Chatbot Hosting	48
3.9	Evaluation	49
3.9.1	Ontology Self-Descriptive Knowledge Evaluation	49
3.9.2	Query Complexity Evaluation	50
3.9.3	Ontology Storage Optimization in Neo4j	50
3.9.4	Chatbot Test Evaluation with Users	50
4	Results	52
4.1	Ontology Self-Knowledge	52
4.1.1	Indicator Retrieval	52
4.1.2	Measure Discovery	53
4.1.3	Dimension Arrangement Query	54
4.1.4	Indicator Composition	54
4.2	Query Complexity Evaluation	54
4.2.1	Scientometric Indicator retrieval in a specific year period	56
4.2.2	Scientometric Indicator retrieved from a time interval range	57
4.2.3	Aggregation functions of the Number of Publication of all Schools in a quinquennium interval	57
4.2.4	Calculation of a new Scientometric Indicator: Cites per Document	58
4.3	Ontology storage optimization in Neo4j	59
4.4	Chatbot Test Evaluation with Users	60
4.4.1	Log Evaluation	60
4.4.2	Survey Evaluation	63
5	Discussion	65
5.1	Data Preparation and Model Construction	65
5.2	Model Evaluation	66
5.3	Ontology Storage	67
5.4	Indicator Querying	67
5.5	Chatbot Evaluation	68

6 Conclusion	71
6.1 Future Work	73
A Appendix	75
Bibliography	82

List of Figures

1.1	Actual process for scientometric data requests.	2
3.1	General Framework - Diagram	19
3.2	Scientometric Indicators filtering and separation process	23
3.3	Scientometric Indicators melting process	23
3.4	Example of unidimensional Scientometric Indicators Dataset definition	25
3.5	Example of multidimensional Scientometric Indicators Dataset definition	26
3.6	Example of uni dimensional Scientometric Indicators DSD	26
3.7	Example of multidimensional Scientometric Indicators DSD	27
3.8	Example of unidimensional Scientometric Indicators Dimension Definition	28
3.9	Example of multidimensional Scientometric Indicators Dimension Definition	28
3.10	Example of multidimensional Scientometric Indicators Measure Definition	29
3.11	Concept Scheme Definition	30
3.12	Concepts Definition	30
3.13	Example of unidimensional Scientometric Indicator Observation Definition	31
3.14	Example of multidimensional Scientometric Indicator Observation Definition	32
3.15	Example of loading an unidimensional Scientometric Indicator in Neo4j	32
3.16	Scientometric Indicators loaded in Neo4j.	33
3.17	Query for obtaining Scientometric Indicators Dataset labels and description.	38
3.18	Example of matching Tag Set with Indicators' Set.	40
3.19	Natural Language Question into Cypher Query	42
3.20	Data Flow from Input to Chatbot response	43
3.21	Greeting Dialog Flowchart and Chatbot View	45
3.22	Option Dialog Flowchart and Chatbot View	46
3.23	Informative Dialog Flowchart	47
3.24	Informative Dialog Flowchart Query	48
3.25	Informative Dialog Chatbot View	48
3.26	Asking-Question Dialog Flowchart	49
3.27	Asking-Question Dialog Chatbot View	50
4.1	Indicator Retrieval Query	52
4.2	Measure Discovery Query	53
4.3	Dimension Arrangement Query	54
4.4	Indicator Composition from measure drill through Query	56
4.5	Scientometric Indicator retrieval in a certain year period Query	56
4.6	Scientometric Indicator retrieved from a time interval range Query	57

4.7	Aggregation functions of the Number of Publication of all Schools in a quin- quennium interval Query	58
4.8	Calculation of a new Scientometric Indicator: Cites per Document Query . .	59
4.9	Goal Completion Rate: Correct Intention	61
4.10	Goal Completion Rate: Correct Indicator	61
4.11	Goal Completion Rate: Correct Answer	63

List of Tables

3.1	Scientometric Indicators' Dataset.	21
3.2	Scientometric Indicators' Description.	22
3.3	Ontologies used in the model.	24
3.4	Frequently asked Scientometric Indicators' questions.	35
3.5	Intent and Utterances Definition	36
3.6	Intent and Entities labeling	37
4.1	Result for query: Which scientometric indicators are in the model?	53
4.2	Result for query: Which measures (metrics) are available in the model?	54
4.3	Result for query: How many dimensions has every indicator?	55
4.4	Result for query: How is measured the number of publications?	55
4.5	Result for query: How many quinquennial cites where produced in 2021 in the EHE school?	56
4.6	Results for query: Show the number of publications made by the institution, reported in quinquennial periods.	57
4.7	Results for query: Calculate the number of schools and aggregation functions such as sum, average, standard deviation, min and max number of quinquennial publications of all the schools from Tecnológico de Monterrey in 2020.	58
4.8	Results Cypher query: Calculate cites per publication?	59
4.9	Data Evaluation.	60
4.10	Intent Identification	61
4.11	Entity Extraction	62
4.12	Goal Completion Rate: Correct Answer	62
4.13	Survey Data	63
A.1	Complete list of Scientometric Indicators from Research Office	75

Chapter 1

Introduction

Nowadays, technology's impact has made systems generate an immense volume of information, making it increasingly difficult to analyze. Having much information and not exploiting it is a problem we face in this situation. Data is a big asset for decision-making as it allows discovering of opportunity areas and actionable insights into what is happening in the evaluated process. It is crucial to have data integrity and consistency as all the outcomes depend on it. Structure, extraction, and data transformation play a considerable role in ensuring the process's best quality.

Business Intelligence is one of the most important models that make possible data exploitation. It is a data-driven system that combines data gathering, data storage, and knowledge management [45]. Performing all BI processes, from the ETL (Extraction, Transformation, and Load) to the realization of reports and dashboards where information is displayed to answer frequent questions of any process evaluated for decision making. Answering a question in BI platforms implies the conceptualization of the question, revision if the answer relies on the loaded data, construction of reports, and other procedures that could take time to be ready to answer. After having it all set, if someone wants to answer several questions about the data, it may have to look in several sets of constructed reports and may not find the answer, and a new BI requirement must be developed as a consequence. PowerBi, a business analytics service by Microsoft, is recognized as the segment leader of Analytics and Business Intelligent Platforms in 2020 by Gartner. It is a service that provides data extraction, data transformation, interactive visualization, and sharing on the same platform. It has developed a new feature called Q&A. This feature allows users to ask questions using natural language [40] to explore data with your own words and sometimes the fastest and most accessible way to get an answer to your question. In this research work, we will develop a new approach with the same scope and conceptualization of Q&A integrated into a chatbot.

A chatbot is a machine conversation system that interacts with human users via natural conversational language [55]. Chatbots are increasing their appearance in several environments related to software interaction with humans. This software is used to perform tasks such as quickly responding to users, informing them, helping them purchase products, and providing better service to customers [18]. Common chatbots' applications are Frequently Asked Questions (FAQ), Customer Support, and helping users obtain fast answers to their questions. The chatbot creates strength in the methodology by implementing several Natural Language Processing techniques that allow the user to feel a one-to-one conversation. One

of the best advantages of using a chatbot is creating a context in a conversation. Comparing the chatbot against other alternatives such as Online Query or Dashboards, the conversation agent can use the conversation history, including questions and answers, to create a context and have a fluent conversation with correct responses.

Our chatbot's main topic is scientometric indicators. The term scientometrics was coined by Vassily V. Nalimov in the 1960s and referred to the science of measuring and analyzing science, such as a discipline's structure, growth, change, and interrelations [36]. Scientometric analysis studies the quantitative areas of the process of science, science policy, and communication in science by having a focus on the measure of authors, articles, journals, and institutions by understanding citations related to them [70]. Braun and colleagues identify Scientometrics as focused on the study of scientific information, specifically in the analysis of the quantitative aspects of the generation, propagation, and utilization of scientific information to contribute to a better understanding of the mechanism of scientific research activities [63]. Vinkler [62] refers to scientometric indicator as to the measure of a single scientometric aspect of scientometric systems represented by a single scientometric set with a single hierarchical level, also called gross indicators.

1.1 Problem Definition and Motivation

Currently, the Tecnológico de Monterrey's Research Office faces the problem of organizing statistical information about its research units' current and past research works. The actual process shown in Figure 1.1 begins with the Research Office integrating information from diverse data sources such as Scopus and institutional databases.



Figure 1.1: Actual process for scientometric data requests.

The Research Office receives the questions about scientometric indicators and interprets them with their knowledge and context to know what and where to search. A problem in the interpretation stage of the process is when a concept has several definitions, leading to wrong answers to the asked question, which would lead to bad decisions. After receiving the question and making an interpretation of it, the specialist performs statistical operations on scientific data following indicator formulas. Information is obtained from different sources, and it is challenging to ensure that it is up to date. Afterward, statistical operations are made, and the specialist responds to the requesting department. A flaw in this process is that it is done manually, and the information is obtained from different sources, in most cases, worksheets.

Scientometric indicators are calculated each year in Tecnológico de Monterrey by the Research Office for decision-making. Statistical information is gathered from heterogeneous and distributed sources to uncover insights, make predictions, and build smarter systems for the institution [21]. One example of a scientometric indicator is the Citation Count, which is the sum of citations received to date by institutional outputs and answers the question of how much impact an institution's academic unit has [24]. This metric is defined by the Common European Research Information Formation (CERIF) that has been developed as a flexible model to describe any research data and information, both as a database model but also as a transfer method between repositories [28].

Data volume is increasing, and the structure is turning complex as this can produce data inconsistency. Data inconsistency appears when multiple tables in a database dealing with the same data are affected by different inputs. It leads to severe problems like redundant information such as different definitions from the same concept and other types of information in distinct sources. These problems can be approached with knowledge management. Wang et al. [66] stated that dealing with inconsistencies is one of the important challenges in data integration and may be detected in query results when processing user's queries in 3 different levels: schema, data representation, and data value.

Another problem presented in the current situation is that every step is done manually. Automating the overflowing process will improve it and drive it to efficiency. The less time the users have to ask similar questions to the workers and ensure that they get a correct answer will let the users have more time to add value to their projects and tasks and reduce the time and effort spent answering the questions.

Availability also plays a role in this process. There are situations in which the users need a fast answer because an exceptional event occurred or an urgent meeting was set up for taking an important decision that requires the response of some questions about the indicators to be validated. It would be impossible to ask this kind of question as it could be in not work schedules, weekends, or the answer is needed in minutes. As we mentioned before, interpretation and statistical operations could take time to be ready to answer the questions. This necessity requires an improvement in the presented scenario, and software automated service is the key.

Our motivation is to have a reliable ontology model that ensures data integration and interoperability by building a flexible, updatable, and easy-to-maintain model providing quick access for several applications. Our approach is evaluated by performing scientometric indicator discovery, enquiring, and composition supported by the SDMX data model. The proposed model has multiple advantages against other alternatives to solve the mentioned problem. Creating an ontology offers knowledge representation and the ability to solve complex tasks such as dialogue in natural language and reasoning. Munir [44] described that one of the significant advantages of using ontology is the ability to define a semantic model of the data combined with the associated domain knowledge. The ontology model will also produce knowledge conceptualization of the domain, and it is valuable for the department's internal processes.

1.2 Hypothesis and Research Questions

The main *hypothesis* of my research is that a chatbot can be used along with an ontology model of the information to prompt correct answers to any question given in the scientometric indicator context.

In this way, the main research question is formulated as follows:

How could a research statistic's chatbot answer questions with a semantic ontology and structured scientometric indicators as an extension of SDMX vocabulary?

The following research questions were stated to answer the main research question:

- Which benefits are obtained by integrating an ontology as a knowledge base with a chatbot?
- How to fully understand users' intents related to scientometric indicators?
- If there is not enough information to answer the question, how will the chatbot converse to create a context to gather the needed data?
- Which important keywords does the chatbot need to recognize to perform the tasks?
- On what will the SDMX extension consist of for modeling the Indicators?

The chatbot will be validated in terms of completeness. Completeness refers to a specific set of indicators in which the chatbot recognizes and answers correctly. Time validation is a subject of study in performance; unfortunately, previous work does not exist that we can use to compare the new development in this research work.

1.3 Objectives

This research's general objective is to develop a chatbot using an ontology model and indicators modeled with an SDMX extension that understands natural language questions related to the scientometric indicators and answers correctly.

1. To propose and generate an ontology model whose characteristics allow the model to be updatable and provide quick access.
2. Apply an ontology model that allows the chatbot to understand and classify the user's input to extract the correct information and answer the question correctly.
3. Evaluate the use of SDMX for modeling the scientometric indicators.
4. Propose and design a scalable chatbot that creates a context for missing information for answering a question, and that can be used in the future in other academic areas or industries.

1.4 Contributions

- A suitable computational framework for answering questions in the scientometric indicator domain, which integrates an ontology model as a knowledge database.
- A scalable model that is updatable and is able to add more indicators to the model without changing the structure.
- Propose different evaluation approaches in terms of self-descriptive knowledge, query complexity, and time and storage optimization from the actual process.
- A logic process for understanding a natural language question and extracting valuable information to translate it to a query language.
- A deployed chatbot in a simulation environment that answers the scientometric indicator domain questions.
- A new dataset that contains questions about the domain as utterances with their intent and entities classification.

1.5 Thesis Structure

This thesis consists of 6 chapters, including this introduction. Literature Review is presented in Chapter 2. Chapter 3 describes the solution proposal of designing and implementing a chatbot that integrates an ontology modeled with scientometric indicators using SDMX as a knowledge base. Results are shown in Chapter 4. In Chapter 5, we discuss the results and provide an analysis of them. Finally, in Chapter 6, we present the Conclusions and Future Work.

Chapter 2

Literature Review

This section defines essential concepts to understand how the research works. The section is divided into the following subsections: Artificial Intelligence, Conversational Agents, Chatbot, Ontology, SDMX, Related Work, and Summary.

2.1 Artificial Intelligence

In 1956, the field of Artificial Intelligence (AI) was founded at a workshop at Dartmouth College by John McCarthy and defined as the science and engineering of making intelligent machines [39]. AI is a field of computer science and engineering concerned with the computational understanding of what is commonly called intelligent behavior and with the creation of artifacts that exhibit such behaviour [54]. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from experience [26].

Research in AI has focused on the following components of intelligence described by Copeland [26]:

- **Learning:** There are several different forms of learning as applied to artificial intelligence. Learning such as try and error or generalization are implemented on computer programs. An example is a computer chess program randomly moving until a checkmate is found. The solution is stored, and then the program learns from the steps and movements and tries to reach the solution again.
- **Reasoning:** To reason is to draw inferences appropriate to the situation. Inferences are classified as either deductive or inductive. Deductive reasoning case the truth of premises and guarantees the truth in conclusion. It appears in mathematics and logic to elaborate irrefutable theorems from axioms and rules. Inductive reasoning case the truth of the premises and gives support to the conclusion. It is common in science and appears in data collection and prediction.
- **Problem solving:** It may be characterized as a systematic search through a range of possible actions to reach some predefined goal or solution. It is classified into special and general purposes. A special-purpose method often solves a particular problem by

exploiting particular features of the problem's situation and environment. A general-purpose method can apply to a wide variety of problems.

- **Perception:** Different kinds of sensors scan the environment and classify them into different objects with certain relationships in a space state.
- **Language:** A language is a system of signs having meaning by convention. Computer programs are intended to communicate and respond fluently in human language to queries and queries in a restricted context.

Artificial Intelligence (AI) is classified into two types:

- **Weak AI:** is trained and focused on performing specific tasks. Weak AI drives most of the AI that surrounds us today. An example of weak AI applications is Apple's Siri and Amazon's Alexa, the IBM Watson computer that vanquished human competitors on Jeopardy, and self-driving cars [13].
- **Strong AI:** fully replicates the autonomy of the human brain—AI that can solve many types or classes of problems and even choose the problems it wants to solve without human intervention[13].

AI is not a futuristic vision but rather something here today and being integrated with and deployed into various sectors. It includes finance, national security, health care, criminal justice, transportation, and smart cities. There are numerous examples where AI is already making an impact on the world and augmenting human capabilities in significant ways [68].

2.2 Conversational Agents

Agents are primarily computer programs that can do work for the users. They are responsible for doing tasks on behalf of the users. Agarwal et al. [16] described characteristics of agents:

- **Autonomous:** The agent goes beyond a simple software program where functionality is considered. It has a fair degree of control over its actions, does not always have to wait for commands, and can make decisions of its own and function independently.
- **Persistent:** Agents can run continuously. They persist over time, i.e., the output of one stage affects the next stage.
- **Reactive:** Agents can perceive changes in the environment and adapt their behaviour in response to the changing environment.
- **Proactive:** Agents are goal-oriented and take proactive initiatives toward fulfilling the goals set for them.
- **Personalized :** Agents learn over time and can also be taught what to do in a particular situation.

- **Social Behavior:** Agents can interact and collaborate with other users to help them to achieve their goals.

Nowadays, conversational agents are increasing their appearance to perform different tasks and help people in various domains and utilities. A conversational agent is a dialogue system that conducts natural language processing and responds automatically using human language. These agents represent the practical application of computational linguistics, usually employed as chatbots over the internet or as portable device assistants. This interpretation/response interaction does not have to be conducted just with text [2].

2.3 Chatbot

The chatbot concept originated from the proposed program called Eliza in the 1960s, which intended to make natural language conversation with a computer possible [67]. One of the first objectives of this program was to pass the Turing Test, in which a human interrogator considers if a computer was intelligent enough to pass as a human [61]. Eliza received as inputs sentences that were analyzed and decomposed as rules matched to response, and an answer was generated.

A chatbot is a machine conversation system that interacts with human users via natural conversational language. Its technology integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language [55]. Usually, a chatbot works by a user asking a question or making a comment, with the chatbot answering the question, making a comment, or initiating a new topic [38]. Chatbots receive natural language input, sometimes interpreted through speech recognition software, and execute one or more related commands to engage in goal-directed behaviour (often on behalf of a human user). As intelligent agents, they are usually autonomous, reactive, proactive, and social. The most advanced systems employ machine learning (often Markov chains or deep neural networks) so that they may also adapt to new information or new requests [51].

According to Adamopoulou et al. [15], chatbots can be classified by the knowledge domain, service provided, goals, input processing, response generation method, human-aid method, and build method.

- Classification on knowledge domain: considers the knowledge a chatbot can access or the amount of data it is trained upon.
 - **Open domain chatbots** can talk about general topics.
 - **Closed domain chatbots** are focused on a particular knowledge domain.
- Classification on Goals: considers the primary goal chatbots aim to achieve.
 - **Informative chatbots** are designed to provide the user with information stored beforehand or available from a fixed source, like FAQ chatbots.
 - **Chat-based/Conversational chatbots** talk to the user like another human being, and their goal is to respond correctly to the sentence they have been given.

- **Task-based chatbots** perform a specific task such as booking a flight or helping somebody.
- Classification on Build Method:
 - **Rule-based model chatbots** are the type of architecture which most of the first chatbots have been built with, as numerous online chatbots. They choose the system response based on a fixed predefined set of rules based on recognizing the lexical form of the input text without creating any new text answers.
 - A **retrieval-based chatbot** retrieves some response candidates from an index before it applies the matching approach to the response selection. It offers more flexibility as it queries and analyzes available resources.
 - The **generative model** generates answers in a better way than the other three models, based on current and previous user messages. These chatbots are more human-like and use machine learning algorithms and deep learning techniques.
- Classification on Human Aid:
 - **Human-aided chatbots** utilize human computation in at least one element from the chatbot.
- Classification on Platform Development:
 - **Chatbot Platforms** provide an interface for working with intents, entities, and actions to facilitate the development and testing of the bot. Some examples are Microsoft Luis, IBM Watson, and Google DialogFlow.

As a rule, chatbot services are delivered by multi-turn Question Answering. In order to produce responses, chatbots require natural language processing techniques, dialogue management modules, and external knowledge bases (e.g., corpora of data). The natural language processing functions as the basic algorithm to parse the input of texts, and the dialogue management modules manipulate the conversational process [22].

2.4 Natural Language Processing

Natural Language Processing (NLP) is a prominent field of Artificial Intelligence. Natural Language Processing is the processing of natural language to derive meaning from it. It helps the computer to understand the text as humans do. Natural Language Processing has many useful applications in Machine Translation, Information retrieval, Question Answering, and a lot of other important fields [19].

NLP's foundations lie in several disciplines, namely, computer and information sciences, linguistics, mathematics, electrical and electronic engineering, artificial intelligence and robotics, and psychology. NLP applications include some fields of study, such as machine translation, natural language text processing, and summarization, user interfaces, multilingual and cross-language information retrieval (CLIR), speech recognition, and artificial

intelligence and expert systems [23]. Natural Language Processing (NLP) and Natural Language Understanding (NLU) attempt to solve the problem of chatbot development by parsing language into entities, intents, and a few other categories described by Ramesh et al. [52]:

- **Entities** are natural language variables associated with certain standard phrases used in daily life. Entities can either be system-defined or developer-defined.
- **Intents** correspond to what actions are to be invoked or triggered as a response to user input. Actions correspond to the steps the chatbot will take when specific intents are triggered by user inputs and may have parameters for specifying detailed information about it.
- **Contexts** are strings that store the context of the object the user is referring to/talking about.

Natural Language Processing has a list of techniques that impact real applications and others that work as a subtask to reach a specific goal. The following Syntactic and Semantic Analysis techniques described by Garbade [12] are the main techniques for completing such tasks:

- Syntactic Analysis: Syntax refers to the arrangement of words in a sentence such that they make grammatical sense. In NLP, it is used to assess how natural language aligns with grammatical rules. Computer algorithms are used to apply grammatical rules to groups of words and derive significant meaning from them. Some techniques:
 - **Lemmatization:** Minimize several forms of a word into a single form for easy analysis.
 - **Morphological segmentation:** Distribution of words into individual units called morphemes.
 - **Word segmentation:** Division of a large piece of continuous text into distinct units.
 - **Part-of-speech tagging:** Identification of the part of speech for every word.
 - **Parsing:** Perform grammatical analysis for the provided sentence.
 - **Sentence breaking:** Place sentence boundaries on a large piece of text.
 - **Stemming:** Cut inflected words to their root form.
- Semantic Analysis: Semantic refers to the meaning produced by a text. It involves computer algorithms that understand the meaning and interpretation of words and how sentences are structured. Some techniques:
 - **Named entity recognition (NER):** Determination of the parts of a text that can be identified and categorized into preset groups.
 - **Word sense disambiguation:** Determine the meaning of a word based on the context.

- **Natural language generation:** Involves using databases to derive semantic intentions and convert them into human language.

Natural Language Processing implies directly in real-world applications that help people in common daily tasks and improve their processes and results. Examples of applications mentioned by Zhang et al. [58]:

- The use of machine translation, text processing, and language generation (e.g., Google Translate).
- Speech to text and text to speech.
- Word Processors like Grammarly employs NLP to check the grammatical accuracy of the texts.
- In call centers, the Interactive Voice Response (IVR) application is used to respond to the user's request.
- Personal assistant applications (e.g., Siri, Alexa, and Cortana).

2.5 Ontology

Ontology, also used as a synonym for metaphysics, is the science of what is, of the kinds and structures of objects, properties, events, processes, and relations of reality. It seeks to provide a definitive and exhaustive classification of entities in all spheres of being [56].

Ontology is an explicit specification of conceptualization. It is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents [35].

An ontology typically provides a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary. Depending on this specification's precision, the notion of ontology encompasses several data or conceptual models, e.g., classifications, database schemas, fully axiomatized theories [30].

Different components define an ontology, and the name of these components differs between ontologies depending on the ontology language, the philosophical persuasion, or the author's background. The core components are shared between the ontologies; they are classified between the components that describe the Entities of the domain (concepts, individuals, and relationships) and the components that describe themselves [43].

Lord [43] describes that a computational ontology consists of several different components:

- **Ontological Aspects**
 - **Concepts:** Also called classes, types, or universal, are a core component in an ontology. It represents a group of different individuals that share characteristics. A subconcept, also known as a subclass, is defined as a class that consumes another class.

- **Individuals:** Also known as instances are the base unit of an ontology, they are the things that ontology describes and may model concrete and abstract objects.
- **Relationships:** Concepts share relationships with other concepts, and it describes the way individuals of one concept relate to the individuals of another.
- Non-Ontological Aspects:
 - **Documentation:** Formal definitions that are provided for each concept, relation, and individual.
 - **Ontology metadata:** Provides documentation for the ontology describing the purpose and scope of the ontology, the release date or version number, and the authorship of the ontology.
 - **Imports:** Other ontologies from which entities have been used and which are required to have a full understanding of the domain.

Roussey et al. [53] stated that ontologies could be classified depending on the dimension in which the ontology is focused. The dimensions are the expressivity and formality of the language used, and the other is based on the object's scope described by the ontology.

- Classification Based on Language Expressivity and Formality
 - **Information ontologies:** are composed of diagrams and sketches used to clarify and organize the ideas of collaborators in the development of a project.
 - **Linguistic ontologies:** can be glossaries, dictionaries, controlled vocabularies, taxonomies, folksonomies, thesauri, or lexical databases
 - **Software ontologies:** provide conceptual schemata whose main focus is generally on data storage and data manipulation and are used for software development activities to guarantee data consistency.
 - **Formal ontologies:** require clear semantics for the language used to define the concept, clear motivations for the adopted distinctions between concepts as well as strict rules about how to define concepts and relationships.
- Classification Based on the Scope or Domain of the Ontology
 - **Local ontologies:** are specializations of domain ontologies where there could be no consensus or knowledge sharing.
 - **Domain ontology:** is only applicable to a domain with a specific viewpoint.
 - **Core Reference:** This type of ontology is linked to a domain, but it integrates different viewpoints related to a specific group of users.
 - **General ontologies:** are not dedicated to a specific domain or field. They contain general knowledge of a vast area.
 - **Foundational ontologies:** are generic ontologies applicable to various domains. They define basic notions like objects, relations, events, and processes.

All the mentioned ontologies are created using a specific language. Two of the most common languages are:

- **RDF:** it is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed [53]. RDF is composed of triples that include the subject, property or predicate, and object [11].
- **OWL:** is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language. Knowledge expressed in OWL can be exploited by computer programs, e.g., to verify that knowledge's consistency or make implicit knowledge explicit. OWL documents, known as ontologies, can be published on the World Wide Web and may refer to or be referred from other OWL ontologies. OWL is part of the W3C's Semantic Web technology stack, including RDF, RDFS, SPARQL, etc. [10].

There are different approaches in terms of designing an ontology. Roussey et al. [53] suggested that all these approaches and methodologies share common and core activities to construct an ontology. These core activities are ontology specification, knowledge acquisition, conceptualization, formalization, implementation, evaluation, maintenance, and documentation.

2.6 SDMX

SDMX, which stands for Statistical Data and Metadata eXchange, aims to standardize and modernize ("industrializing") the mechanisms and processes to exchange statistical data and metadata among international organizations and their member countries. It is sponsored by seven international organizations, including the Bank for International Settlements (BIS), the European Central Bank (ECB), Eurostat (Statistical Office of the European Union), the International Monetary Fund (IMF), the Organization for Economic Cooperation and Development (OECD), the United Nations Statistical Division (UNSD), and the World Bank [14].

These organizations are the main players at the world, and regional levels in the collection of official statistics in a large variety of domains (agriculture statistics, economic and financial statistics, social statistics, environment statistics, etc.) [14].

Cyganiak et al. [27] described that SDMX covers everything from how to represent statistical data in flat files and as XML, to the definitions of the dimensions and attributes of observations, to how to discover statistical dataset flows through a central registry. In their work, they also described how SDMX is structured. The model is based on an earlier standard, GESMES/TS, which used the UN/EDIFACT flat-file syntax. SDMX, in its current version, has expanded this model to include a view of the entire process of statistical production.

2.7 Linked Data Platforms

A Linked Data Platform (LDP) provides a set of integration patterns for building RESTful HTTP services capable of reading and writing RDF data. Under this definition, we can find applications like VIVO and Neo4j that, to some extent, enable the visualization of information stored in RDF format.

VIVO is an open-source linked data platform that supports recording, editing, searching, browsing, and visualizing scholarly activity. It encourages research discovery, expert finding, network analysis, and assessment of research impact [25]. The VIVO ontology contains the schemas required for representing this information.

Neo4j is a native graph data store built from the ground up to leverage data and data relationships. It connects data as it is stored, enabling queries at high-speed [9]. Neo4J uses native graph storage, which provides the freedom to manage and store data highly disciplined manner. It is considered the most popular and used graph database worldwide, used in areas such as health, government, automotive production, and military area, among others [31].

Stothers defines some advantages of using Neo4j graph database as follows [57]:

- Well suited to storing information structures that are not well suited to relational databases, such as ontologies or networks.
- Operational simplicity, especially in using relationships to avoid joining tables.
- Ability to include properties in relationships and nodes.
- Neo4j query language can present query results in multiple formats allowing creative insights into data interpretation.
- Efficient queries and an attractive interface lead to ease of use and an intuitive user experience.

In order to load RDF triplets to the graph database, the plugin called n10semantics from the Neo4j labs needs to be installed. This plugin enables RDF and its associated vocabularies for data interchange (OWL, RDFS, SKOS, and others). This plugin is also used to build integration with RDF generation and consuming components [41]. Some functionalities that are included by installing this plugin are the following:

- Import and Export RDF in multiple formats (Turtle, N-Triples, JSON, etc.)
- Model mapping on import and export
- Import and Export Ontologies in different vocabularies
- Graph validation
- Basic Inference

Cypher is the graph-optimized query language incorporated in Neo4j. It understands and takes advantage of connections (relationships) between data. It is inspired by SQL, with the addition of pattern matching borrowed from SPARQL. It uses simple ASCII symbols to represent nodes and relationships, making queries easy to read and understand [46].

2.8 Scientometric Indicators

The chatbot domain relies specifically on scientometric indicators that belong to the field of Scientometrics. It is of study which concerns itself with measuring and analyzing scientific literature [59]. The use of scientometric indicators in research evaluation emerged in the 1960s and 1970s, first in the United States and then also in various European countries [42]. Scientometric indicators can be defined as metrics founded on bibliographic facts and figures employed to quantify and evaluate the scientific, scholarly output of an individual, institution, nation, and so on [69]. Some examples of scientometric indicators are the following: Publications, Cites, Cites per Document, Author, Co-author, Patents, PosDocs, etc.

2.9 Chatbot Evaluation

The use of intelligent conversational agents, such as a chatbot, has increased during the past years. An important way to analyze if the chatbot is accomplishing the task is by performing several evaluations on it. The work by Peras [49] presented a review on evaluation metrics for measuring success in a chatbot, and some of the several evaluations proposed perspectives along their metrics are the following:

- User Experience: Usability, Performance, Affect and Satisfaction
- Information Retrieval: Accuracy, Accessibility and Efficiency
- Linguistic: Quality, Quantity, Manner, Grammatical Accuracy
- Technology: Humanity
- Business Perspective: Business value

Another type of metric that belongs to the information retrieval perspective is the Goal Completion Rate (GCR). It calculates the percentage of users who finish a particular business goal or series of goals [6]. We apply this concept to evaluate how many business goals (chatbot processes) can be performed in this situation.

2.10 Related Work

We revised previous works on which indicators are semantically modeled and enquired. Moreover, we analyzed ontologies used for representing science or scientometric data. We also introduce previous chatbots' work with different goals and the scientific community.

Semantic indicator modeling

Fox proposed modeling city indicators with a semantic approach in 2018 [33]. His approach includes key aspects such as membership extent, temporal extent, spatial extent, and measurement of populations. Fox uses the RDF Data Cube Vocabulary to specify city population

dimensions, but the SDMX standard was not incorporated as part of his solution. The ontology evaluation was divided into the population's representation as indicators definition, consistency of indicator definitions against the interpretation of a city, and how it can be used to support data collection of a city.

A semantic approach can be implemented in several contexts; one of the most common scenarios is in statistical databases. The work by Thiry et al. [60] presented an interactive tool for a question answering system that accesses statistical databases, and it follows the SDMX standard. In this research work, they look at understanding general dimensions from user questions and found that time and location were dimensions for this kind of data. This system was evaluated by testing queries and measuring the accuracy of the result in terms of detecting dimensions. Queries were tested using only one dimension. Many institutions use SDMX, and this research work represents a good approach for answering questions about the selected data.

Scientometric Ontologies

Hu et al. [37] converted data, originally stored in a relational database, collected from the Semantic Web Journal to RDF and published them as linked data. This data contains an entire timeline for each paper and metadata from the Semantic Web Journal (SWJ) unique open and transparent review process. This linked data gives insights into scientific networks and new trends. The Bibliographic Ontology (BIBO) ontology was extended to capture information about the paper's timeline. BIBO provides main concepts and properties for describing citations and bibliographic references (i.e., quotes, books, articles, etc.) on the Semantic Web [47].

Osborne et al. [48] presented a novel approach for clustering authors according to their citation distribution. This work introduced the Bibliometric Data Ontology (BiDo), which allows an accurate representation of such clusters. BiDO is a modular ontology encoded using Ontology Web Language (OWL) 2, that allows the description of bibliometric data of people, articles, journals, and other entities described by Semantic Publishing and Referencing (SPAR) Ontologies in RDF [50]. BiDO has kinds of bibliometric data: numeric and categorical. Some measures such as citation count, e-index, and journal impact factor are available through BiDO's numeric property. Categorical data is for specifying categories describing the research career of authors.

Chatbot

Baby [19] presented a chatbot using Natural Language Processing techniques to understand and extract key information on user requests. The chatbot was connected to a web application for home automation and performed several fan or light controlling tasks and other electrical appliances. In our research work, a chatbot approach is to fully understand the request and create a context to have better insight for answers.

Munir [44] discussed several approaches in the creation of ontology models for retrieving information. There is an increase in search requests for retrieving information. The structural complexity of a database and semantic relationship in data is compromised because the volume gets bigger each day. The research outcome compares ontology-based information

retrieval, database-to-ontology transformations, and ontology-to-database mappings in terms of capabilities.

Huang [38] worked in a chatbot for extracting Knowledge from Online Discussion Forums. Extraction was made using an SVM classifier to obtain a pair of <thread-title, reply> based on correlations of structure and content, and the pairs were ranked based on the quality of the content. It was demonstrated as an effective method for obtaining chatbot knowledge.

Chen [22] created a chatbot for attacking a frequently presented problem in the integration of migrants in a host country. Experimentation shows that effectiveness may improve by involving users in designing expected answers for a chatbot's common tasks.

2.11 Summary

In the literature review presented above, we found several methodologies that can be applied for representing and enquiring scientometric indicators, but that must be integrated into a single solution. First, we must select a semantic representation of statistical data. Some authors used BIDO to represent numerical data about citations and other specific indicators. We found an area of opportunity because it is a problem to talk about numerical data in one way and when looking at other works, they handle it differently. The SDMX will allow us to have a standardized way to represent any indicator. On the other hand, SDMX can be extended in terms of dimension, attributes and values appropriate for describing publications, citations, researchers, etc. This extension solves the problem of having data with a certain level of information, such as authors' properties. In this way, we have the flexibility for defining dimensions proper of scientometric indicators. After revising state of the art, research works of chatbots using an ontology as a knowledge base were not found.

Finally, a semantic approach must be evaluated to demonstrate its advantages over traditional approaches. SDMX provides a consistent representation of multidimensional data, but it must permit to capture particular differences between indicator definitions (e.g., annual versus quinquennial periods). Besides, we must assure that any person familiar with SDMX is capable of discovering, enquiring, and composing scientometric indicators encoded with this data model. We also must provide high-performance on query answering, so we decided to use the Neo4j platform for this purpose.

Chapter 3

Chatbot Design

The research work's scope is to design and implement an ontology model in a chatbot for a subset of scientometric indicators defined by the office. The goal is to construct the solution to scale to other indicators and other areas of the institution. The main goal of our research is to build a model of scientometric indicators. The model will utilize RDF to describe the resources by extending SDMX with a vocabulary appropriate for representing dimensions, attributes, and values found in scientometric indicators (e.g., schools, cites, papers). We will extract a sample of scientometric indicators used in the Research Office from Tecnológico of Monterrey to evaluate our approach. Data will be transformed manually to RDF, and a tool will be constructed for automation. Another goal is to deploy this model in a graph database to evaluate queries and visualizations. Moreover, this ontology model will be integrated as a knowledge base for a chatbot in order to answer questions about the scientometric indicators domain.

3.1 Proposal Solution

The proposed framework is shown in Figure 3.1 allows us to have a complete path in which we collect and process data for the ontology model and understand natural language questions about scientometric indicators to be able to predict the intention of the query and retrieve the correct answer from the ontology model.

The components of this framework are described next.

1. The original data source is an excel file that contains information about scientometric indicators. We extract a sample of them and classify them by their dimensions. We input the samples in a python script that automates the conversion from tabular way to RDF (triplets) and SDMX (dimensions and attributes). The output is a series of RDF files ready to be used. (1 RDF file = 1 scientometric indicator).
2. We are using a graph database in Neo4j. In order to be able to upload the RDF files in our graph, we needed to add a plugin from neosemantics that allow the use of RDF format. We uploaded each RDF file, and our graph is ready to be enquired. SDMX allows having multi-dimension through nodes and relationships in Neo4j. The language query in Neo4j is Cypher.

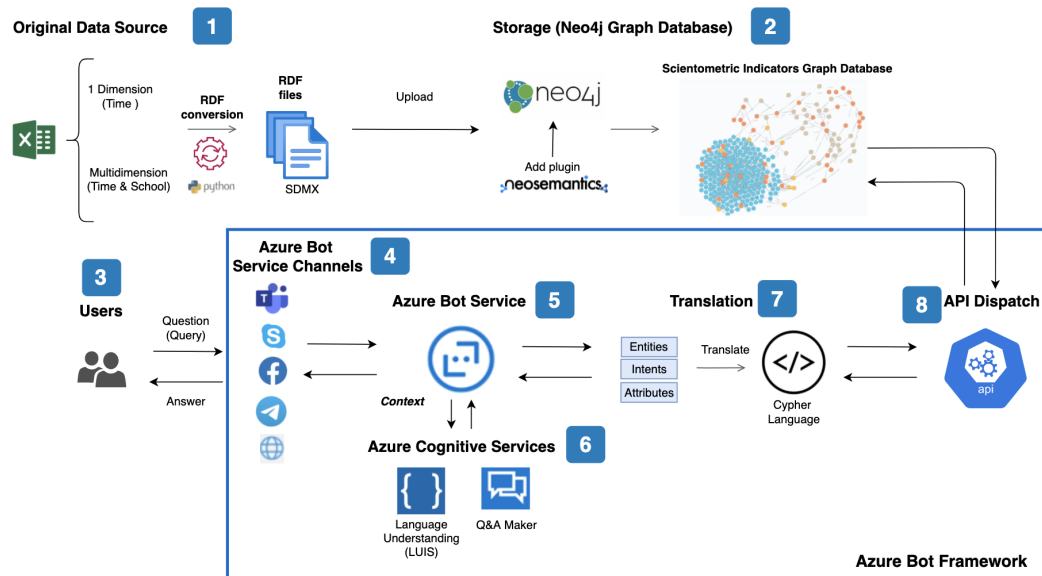


Figure 3.1: General Framework - Diagram

3. Users from the Research Office from Tecnológico de Monterrey will ask questions about scientometric indicators.
4. Azure Bot Service Channels help deploy the chatbot on social media platforms such as Microsoft Teams, Skype, Facebook, Telegram, and a WebChat. This service is the tool for interaction between users and the chatbot.
5. Azure Bot Service is a framework for designing and developing artificial intelligence. Bot Framework Composer allows creating bots more quickly. This service is a very important component because we connect to chatbot platforms, cognitive services, and APIs.
6. Azure Cognitive Services is a component that enables artificial intelligence through an API call. Language Understanding provides machine learning algorithms to predict a user's natural language input and returns important details as output. Q&A provides a natural language processing service.
7. This process consists of the translation from a natural language query (scientometric indicator question) to Cypher query language. The use of cognitive services before this process enables the use of entities, intents, and attributes for constructing the query.
8. The API sends the query and is executed in our graph database. It should return an answer, and it needs to be interpreted in the Azure Bot Service. The answer to the query (question about scientometric indicators) should be shown to the user through the chatbot platform. If the question lacks elements for identifying the main question, context should be used.

3.2 Methodology

This section will explain the methodology of this research work based on CRISP-DM. We describe step by step each process needed to complete the objectives.

3.3 Data Collection

Data for this research work is taken from an official workbook of the Research Office that stores data tabularly. The file is frequently updated, and we took the last version in March 2021. Historical data stays the same unless an error is found in a past calculation. The list of scientometric indicators that appear on this worksheet, shown in A.1, has more than 100 indicators, and each one of them belongs to a category. Among these categories, we can find the following: Publications and Cites, Patents, Students, Researchers, and Rankings. In order to construct and build our model, we decided to take a representative sample of 10 indicators, including both uni-dimensional and multidimensional characteristics. In Table 3.1 we show the sample dataset that we will use for our research work, which includes data from 2017 to 2021.

3.3.1 Data Understanding

In order to build the ontology from a dataset in tabular form, we need to understand the data that we will model. In Table 3.2 we give a brief description of each scientometric indicator.

These indicators were selected to illustrate that both unidimensional and multidimensional indicators can be represented.

- Unidimensional. The first kind of indicator associates a value to a time period.
- Multidimensional. The second kind of indicator associates a value to a time period and another dimension(s) (e.g., schools).

3.3.2 Data Preparation

After selecting the sample of scientometric indicators for modeling, the next step is data preparation. This process is essential because our sampled dataset continues to be a raw source of information, and it needs cleaning and transformation. With this process, we will ensure consolidation and separation of characteristics due to the nature of our dataset also. Transformation is needed to achieve the correct format for our input for the data conversion. We will encourage preparing our data to be able to automate its transformation. The first step of this process is to separate our dataset. The dataset contains the ten scientometric indicators chosen, and we need to separate each indicator into a proper dataset. In Figure 3.2 we can observe that process receives as input the dataset. It employs a python program that completes the filtering and separation of each scientometric indicator into several datasets as output.

Also, in the data collection from the original dataset from the research office, data is stored in a stacked way in which the time dimension (years) is represented as columns. In order to transform the data into a tabular way in which the columns of time dimension pass

Table 3.1: Scientometric Indicators' Dataset.

Scientometric Indicator		Category	School	2021	2020	2019	2018	2017
Quinquennial Publications		Pubs and Cites	TEC	6510	5369	4518	3891	3334
Quinquennial Cites		Pubs and Cites	TEC	33941	22943	18393	12311	9759
Cites per Document		Pubs and Cites	TEC	5.213	4.273	4.071	3.163	2.927
Annual Publications Scopus-Tec		Pubs and Cites	TEC	128	1886	1501	1206	1090
Annual Publications per School	Pubs and Cites	EAAD	0	17	8	3	4	
		EHE	4	133	109	93	76	
		EN	7	110	115	99	108	
		ECSG	1	53	47	45	39	
		EIC	48	1094	1044	793	741	
		EMCS	8	372	319	263	187	
Quinquennial Publications per School	Pubs and Cites	EAAD	40	26	23	21	20	
		EHE	525	468	404	363	328	
		EN	505	450	384	337	287	
		ECSG	229	196	173	149	134	
		EIC	4191	3622	3044	2666	2296	
		EMCS	1267	1020	805	611	494	
Quinquennial Cites per School	Pubs and Cites	EAAD	42	15	59	38	23	
		EHE	1057	711	594	407	416	
		EN	2212	1332	878	623	613	
		ECSG	543	318	179	114	102	
		EIC	25265	17405	14319	9752	7408	
		EMCS	6415	5000	4219	3186	2461	
Cites per Document and School	Pubs and Cites	EAAD	1.05	0.576	2.565	1.809	1.15	
		EHE	2.013	1.519	1.47	1.12	1.26	
		EN	4.38	2.96	2.286	1.848	2.135	
		ECSG	2.371	1.622	1.034	0.765	0.761	
		EIC	6.028	4.805	4.704	3.657	3.226	
		EMCS	5.06	4.901	5.24	5.214	4.981	
Number of Researchers		Researchers	TEC	NA	803	714	665	563
Number of PosDocs		Researchers	TEC	NA	72	81	69	73

Table 3.2: Scientometric Indicators' Description.

Scientometric Indicator	Dimension	Description
Quinquennial Publications	Time	Measures publications in quinquennial time period
Quinquennial Cites	Time	Measures cites in quinquennial time period.
Cites per Document	Time	Measures Publications over Cites in a time period.
Annual Publications Scopus - Tec	Time	Measures annual publications at Tecnológico de Monterrey from Scopus database.
Annual Publications per School	Time,School	Measures annual publications grouped by schools at Tecnológico de Monterrey.
Quinquennial Publications per School	Time,School	Measures publications grouped by schools at Tecnológico de Monterrey in quinquennial time period.
Quinquennial Cites per School	Time,School	Measures cites grouped by schools at Tecnológico de Monterrey in quinquennial time period.
Cites per Document and School	Time,School	Measures publications over cites grouped by schools at Tecnológico de Monterrey in quinquennial time period.
Number of Researchers	Time	Measures number of researchers at Tecnológico de Monterrey in a time period.
Number of PosDocs	Time	Measures number of posdocs at Tecnológico de Monterrey in a time period.

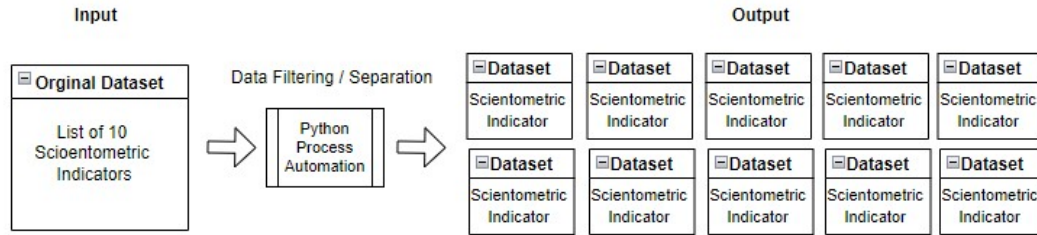


Figure 3.2: Scientometric Indicators filtering and separation process

to be the values of a new column called time, we will proceed to apply a melting process to each scientometric indicator dataset. The melting process reshapes the dataset by selecting a point of start, in this case, all the columns of the time dimension. From this point, the dataset is transformed by following the format from wide to long. As all the columns were variables, they proceeded to be treated as values. In Figure 3.3 we observe the example of the unidimensional scientometric indicator Quinquennial Publications. We observe how applying this transformation produces an unstacked and tabular way.

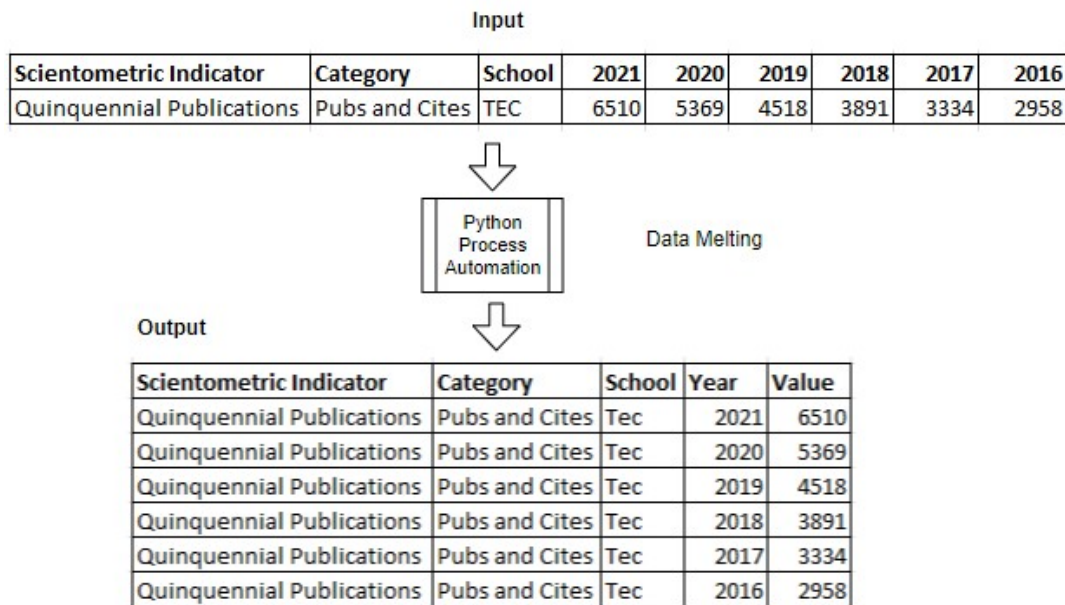


Figure 3.3: Scientometric Indicators melting process

3.4 RDF construction

The next step in the methodology consists of constructing the RDF files. One of our research work goals is to build a model of scientometric indicators. Our approach consists of a model

Table 3.3: Ontologies used in the model.

Prefix	Ontology URI
qb	http://purl.org/linked-data/cube#
sdmx-attribute	http://purl.org/linked-data/sdmx/2009/attribute#
sdmx-dimension	http://purl.org/linked-data/sdmx/2009/dimension#
vivo	http://vivoweb.org/ontology/core
interval	http://reference.data.gov.uk/def/intervals/
tec	https://www.tec.mx/ontos/indicators/

that can give descriptive information on the scientometric indicator and reuses the resources already existing on the web, such as datacubes, statistical data, metadata, and time intervals. We propose to model our scientometric indicators using the RDF standard because it features the merging of different schemas and supports the evolution of the model. The following step consists of building an RDF file for each scientometric indicator dataset. Because scientometric indicators are different in terms of dimensions, the head and observations of the RDF file will differ. The format of the RDF file will be Turtle, as it is one of the valid formats that the Neo4j graph database accepts. A Turtle file allows writing down an RDF graph in a compact textual form. The RDF model uses triples consisting of a subject, a predicate, and an object $\langle s, p, o \rangle$ to represent data [71]. We divided the RDF file format into Vocabularies and Namespace, Dataset, Data Structure Definition (DSD), Dimensions, Measures, School Concept Scheme, and Observations.

3.4.1 Vocabularies and Namespace

The first element of the RDF file is vocabulary and namespaces. It consists of a collection of Internationalized Resource Identifiers (IRIs). These are used to define and document RDF vocabularies in the model with an alias. Also, the referenced IRI is validated for existence for adopting existing concepts and resources at the moment of compiling. Additionally to standard vocabularies for representing RDF/XML (rdf, rdfs, xsd) and ontologies (owl, skos), we incorporated the ontologies shown in Table 3.3. The SDMX ontology is used to model our data and metadata attributes and dimensions. VIVO ontology is included because its research domain will allow us to establish our approach for our model. Interval ontology plays an important role in defining our time intervals, such as annual and quinquennial. The last ontology (Tec) was defined for publishing our definitions. Our objective is to create an easy and flexible model to scale in case of adding new scientometric indicators.

3.4.2 Dataset

In this section of the RDF file, we define the dataset of each scientometric indicator. We created a Universal Resource Identifier (URI) for each scientometric indicator and assigned the `rdf:property` to this URI. It states that all instances of one class are instances of another, and the property dataset of the data cube ontology for defining that this URI represents a collection of observations. With these two properties extended, we define the indicator as an instance of the class `Dataset`. URI can also be identified by a literal for the representation of simple values such as strings or numbers [64]. In order to define a relationship between the scientometric indicator dataset and its observations, we extend the `qb:structure` that indicates the structure to which this dataset conforms. This description of the dataset structure is called `Data Structure Definition` and will be explained in the next section. At the end of the definition of the `Dataset` in the RDF file, we extend the `rdf:label` and `rdf:comment` that provides a human-readable description of the resource. These last properties change depending on the indicator's dimension and will work as a description, allowing us to use it for querying. The `rdf:label` property stores representative entities of the scientometric indicator, while the `rdf:comment` property stores the descriptive name.

An example of a unidimensional scientometric indicator is the `Quinquennial Publications`. In Figure 3.4, we can observe the indicator URI of the dataset and the relation within the indicator structure. In the `rdf:label` property, the following entities: `publication`, `quinquennial`, and `school` represent a tokenized description of itself. For easy identification, the label includes the indicator measure, time interval, and the school dimension if it applies. The comment property simply stores the name of the dataset.



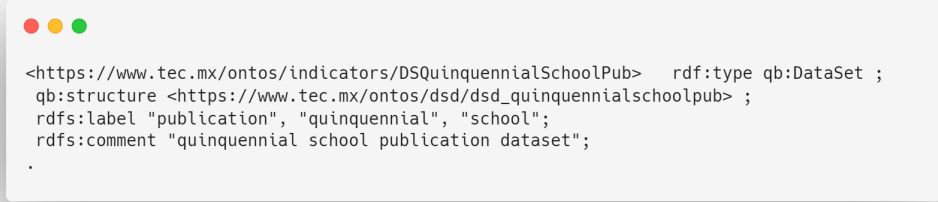
```
<https://www.tec.mx/ontos/indicators/DSQuinquennialPublications>    rdf:type qb:DataSet ;
qb:structure <https://www.tec.mx/ontos/dsd/dsd_quinquennialpublications> ;
  rdfs:label "publication", "quinquennial";
  rdfs:comment "quinquennial publications dataset";
.
```

Figure 3.4: Example of unidimensional Scientometric Indicators Dataset definition

An example of a multidimensional scientometric indicator is the `Quinquennial School Publications`. In Figure 3.5, we can observe that the structure of the dataset definition is the same as in unidimensional indicators. The only difference appears in the `rdf:label`; it varies because it includes the school dimension because it applies in this case.

3.4.3 Data Structure Definitions (DSD)

The `Data Structure Definition` of a dataset describes the components such as dimensions, attributes, and measures [29]. We define the data structure by extending the `rdf:type` property



```


<https://www.tec.mx/ontos/indicators/DSQuinquennialSchoolPub>  rdf:type qb:DataSet ;
qb:structure <https://www.tec.mx/ontos/dsd/dsd_quinquennialschoolpub> ;
rdfs:label "publication", "quinquennial", "school";
rdfs:comment "quinquennial school publication dataset";
.

```

Figure 3.5: Example of multidimensional Scientometric Indicators Dataset definition

that produces an instance of the Data Structure Definition property from the data cube ontology. In the DSD, we extend the `qb:component` and `qb:ComponentSpecification` properties that allow us to define properties of a component from a data structure such as attributes, dimensions, and measures. Data cube ontology can deeply specify the property type using the attribute, dimension, and measure properties. For the attribute and dimension definition, we extend the SDMX ontology properties of `sdmx-attribute:unitMeasure`, which refers to the unit in which the data values are measured, and the `sdmx-dimension:refPeriod`, which refers to the time period of the measured observation. The tec ontology is extended for the measure property as it includes our definitions for our scientometric indicators. At the end of the DSD, we establish its label with the `rdfs:label` property for the descriptive name of the DSD.

In Figure 3.6 we can observe the DSD for the unidimensional Quinquennial Publications indicator. In this DSD, we have the attribute, dimension, and measure defined using the SDMX and Tec ontologies.



```

<https://www.tec.mx/ontos/dsd/dsd_quinquennialpublications>  rdf:type qb:DataStructureDefinition ;
qb:component [      rdf:type qb:ComponentSpecification ;
qb:attribute sdmx-attribute:unitMeasure ;      ] ;
qb:component [      rdf:type qb:ComponentSpecification ;
qb:dimension sdmx-dimension:refPeriod ;      ] ;
qb:component [      rdf:type qb:ComponentSpecification ;
qb:measure tec:NumberOfPublications ;      ] ;
rdfs:label "dsd for datacube quinquennial publication "@en ;
.

```

Figure 3.6: Example of uni dimensional Scientometric Indicators DSD

In Figure 3.7 we have the DSD for the multidimensional Quinquennial School Publications indicator. The definition varies because it also includes the School dimension URI using the Concept Scheme that will be explained further.

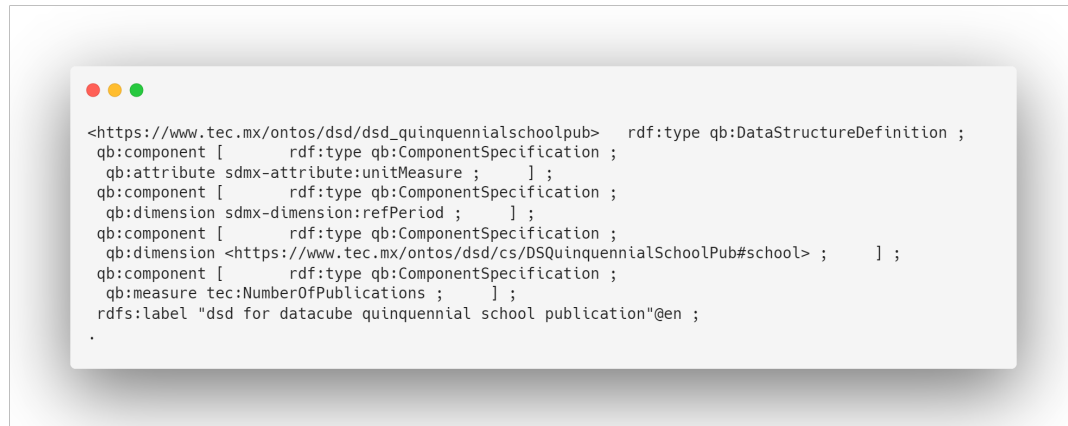


Figure 3.7: Example of multidimensional Scientometric Indicators DSD

3.4.4 Dimensions

We are leading to define the dimensions of our dataset. As we mentioned before, we have two kinds of Indicators, unidimensional (only time dimension) and multidimensional (time and other dimensions). The first step is to extend our tec ontology by defining the time dimension. Here, we define if the indicator's time period is quinquennial or annual. Extending the `rdf:property` and `qb:DimensionProperty`, we establish that this component represents a dimension of our dataset. We extend the `rdf:label` property for describing the time-period dimension. We also extended the property `sdmx-dimension:refPeriod`, used for representing temporal intervals. In our case, we defined an annual and a quinquennial (five-years) time interval. Time interval instances were borrowed from the United Kingdom's reference data server (<http://reference.data.gov.uk/>).

In Figure 3.8 we observe the dimension definition for the unidimensional Quinquennial Publications indicator. This definition allows us to establish and describe the dimensions of our dataset.

In Figure 3.9 For the multidimensional indicators, in this case, the Quinquennial School Publications indicator, we also extended the basic `sdmx-dimension` property. We constrained its value to instances of the School class provided by the VIVO ontology. School instances are available at the VIVO website of our institution (<https://research.tec.mx/>).

3.4.5 Measures

The next step is to define the measure of our dataset. For this definition, we extend our tec ontology for defining the kind of measure used in each dataset. The measures included in the Tec ontology are the following:

- Number of Publications.
- Number of Cites.
- Number of Cites per Document.

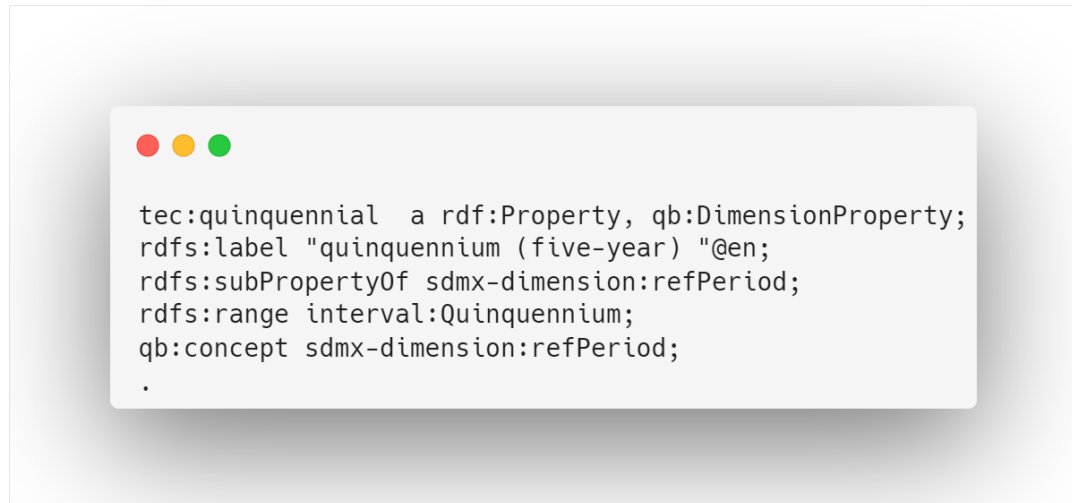


Figure 3.8: Example of unidimensional Scientometric Indicators Dimension Definition

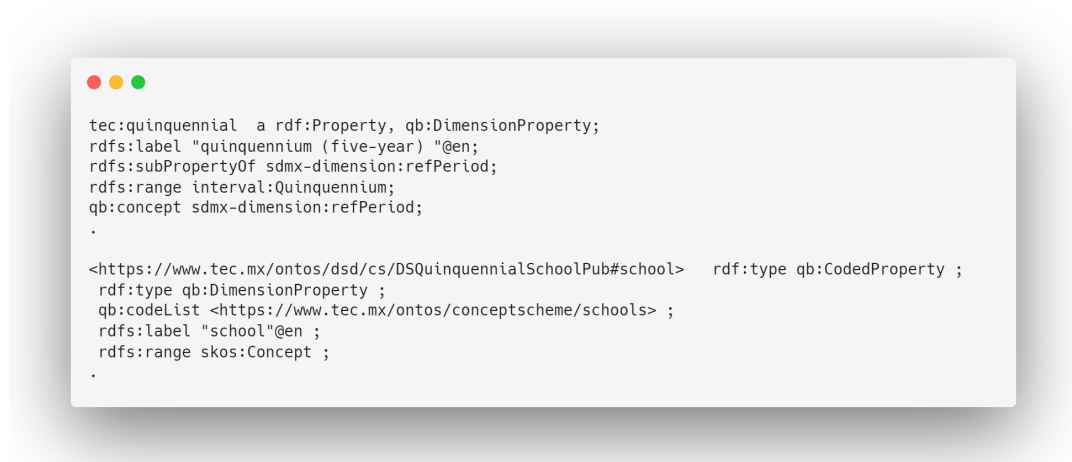


Figure 3.9: Example of multidimensional Scientometric Indicators Dimension Definition

- Number of Researchers.
- Number of Postdocs.

We use the `rdf:property` and the `qb:MeasureProperty` to represent the attribute of the observations in our dataset. We extended the property `sdmx-attribute:unitMeasure` to represent the unit in which data is measured. We added the `xsd:integer` property to specify numerical values. In Figure 3.10 we show a measure definition example for the unidimensional Quinquennial Publications indicator. There is no difference in the definition between unidimensional and multidimensional indicators.



```
tec:NumberOfPublications rdf:Property qb:MeasureProperty ;
rdfs:label "Number of publications"@en ;
rdfs:subPropertyOf sdmx-attribute:unitMeasure ;
rdfs:range xsd:integer ;
.
```

Figure 3.10: Example of multidimensional Scientometric Indicators Measure Definition

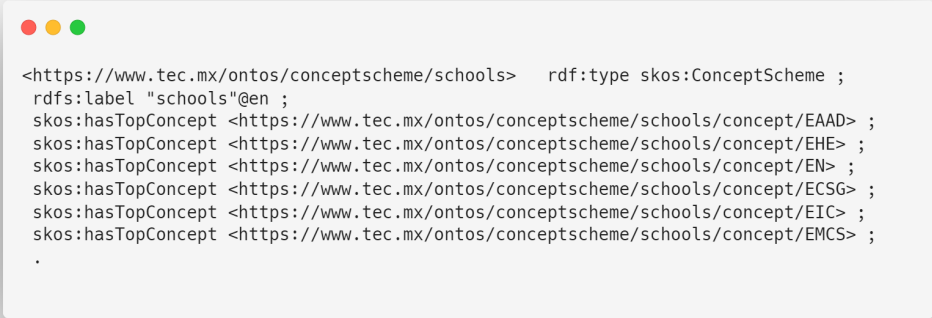
3.4.6 Concept Scheme

A concept scheme provides labels to concepts and realizes both hierarchical and associative links [34]. For this purpose, we used SKOS, a concept-centric data model based on RDF that identifies concepts using URIs to make already available knowledge organization systems public on the web in machine-readable formats [20]. SKOS is devoted to developing specifications and standards that support the use of knowledge organization systems (KOS) such as thesauri, classification schemes, subject heading systems, and taxonomies within the Semantic Web framework. It provides a standard way to represent knowledge organization systems using the Resource Description Framework (RDF) [65]. In our case, we reused common concept schemes such as the list of current Schools, an external dimensional in our dataset. The concept scheme only applies to multidimensional indicators because we use an external dimension besides the time dimension.

We define a unique URI for the concept scheme by extending the `rdf:Property` and the `skos:ConceptScheme` that represent a class viewed as an aggregation of one or more `skos:Concept`. Using the `skos:hasTopConcept` we linked the concepts into the Concept Scheme. We extended all the following list of Schools:

- EAAD: Escuela de Arquitectura, Arte y Diseño
- EHE: Escuela de Humanidades y Educación
- EN: Escuela de Negocios
- ECSG: Escuela de Ciencias Sociales y Gobierno
- EIC: Escuela de Ingeniería y Ciencias
- EMCS: Escuela Medicina y Ciencias Salud

Abbreviations of schools in the concept scheme are used for quick identification in the querying process. In Figure 3.11 we show how all the `skos:Concepts` of school are listed in the definition of the Concept Scheme.



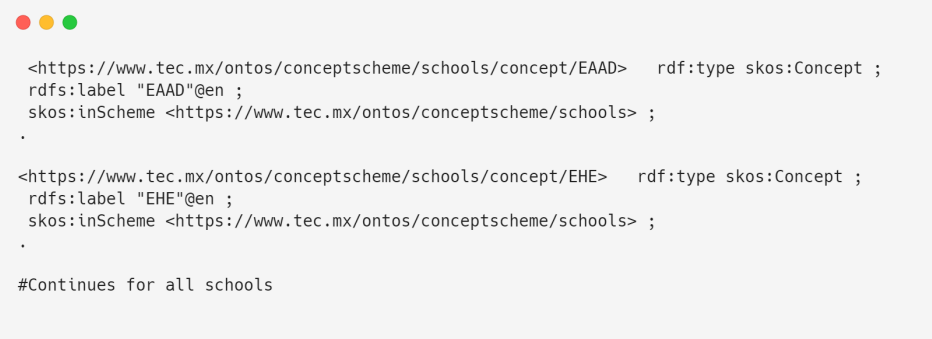
```

<https://www.tec.mx/ontos/conceptscheme/schools>  rdf:type skos:ConceptScheme ;
rdfs:label "schools"@en ;
skos:hasTopConcept <https://www.tec.mx/ontos/conceptscheme/schools/concept/EAAD> ;
skos:hasTopConcept <https://www.tec.mx/ontos/conceptscheme/schools/concept/EHE> ;
skos:hasTopConcept <https://www.tec.mx/ontos/conceptscheme/schools/concept/EN> ;
skos:hasTopConcept <https://www.tec.mx/ontos/conceptscheme/schools/concept/ECSG> ;
skos:hasTopConcept <https://www.tec.mx/ontos/conceptscheme/schools/concept/EIC> ;
skos:hasTopConcept <https://www.tec.mx/ontos/conceptscheme/schools/concept/EMCS> ;
.

```

Figure 3.11: Concept Scheme Definition

The next step is to define each school by establishing a unique URI extending the `rdf:property` and the `skos:Concept` that is represented as a unit of thought. We added the `rdfs:label` for proper identification for human-readable purposes. In the end, we extend the `skos:inScheme` to make the relationship that the concept defined belongs to the Concept Scheme previously defined. In Figure 3.12 we observe the definition of each `skos:Concept`; we define each concept for each school listed.



```

<https://www.tec.mx/ontos/conceptscheme/schools/concept/EAAD>  rdf:type skos:Concept ;
rdfs:label "EAAD"@en ;
skos:inScheme <https://www.tec.mx/ontos/conceptscheme/schools> ;
.

<https://www.tec.mx/ontos/conceptscheme/schools/concept/EHE>  rdf:type skos:Concept ;
rdfs:label "EHE"@en ;
skos:inScheme <https://www.tec.mx/ontos/conceptscheme/schools> ;
.

#Continues for all schools

```

Figure 3.12: Concepts Definition

3.4.7 Observations

In this section, we will define all the observations of our dataset. The first step is to define a unique URI where we describe the name of the dataset and the year or quinquennium of the observation. We extended the `rdf:type` and `qb:Observation` properties to represent a class of a single observation in our dataset that may have one or more associated measured values. Then we extend the `qb:Dataset` property to indicate that this observation belongs to the dataset defined earlier. The next definition comes using the `sdmx-attribute:unitMeasure` and `sdmx-dimension:refPeriod` for indicating the proper URI description. As our measure is numeric, our unit measure is described as a number. For the time dimension, the URI is established depending if it corresponds to the quinquennium or annual time period. For quinquennium, we use a range between the year established and the year established minus five years. For annual, we use the established year. As in other definitions, we extended the `rdf:label` for a human-readable description that will be used in the querying process. In the end, we use our `tec` ontology to extend the property `tec:NumberOfPublications` to indicate the type of measure and the value. In Figure 3.13 we show an example of the unidimensional Quinquennial Publications indicator.

```
<https://www.tec.mx/ontos/observation/DSQuinquennialPublications/2022/count>  rdf:type qb:Observation ;
qb:dataset <https://www.tec.mx/ontos/indicators/DSQuinquennialPublications> ;
sdmx-attribute:unitMeasure <http://qudt.org/vocab/unit#Number> ;
sdmx-dimension:refPeriod <http://reference.data.gov.uk/id/quinquennium/2017-2021> ;
rdfs:label "number of publications (quinquennial) made in 2022" ;
tec:NumberOfPublications 5811
.
```

Figure 3.13: Example of unidimensional Scientometric Indicator Observation Definition

There is a slight difference in the definition for multidimensional indicators where an additional statement is added. As shown in Figure 3.14, we extended the dimension of the school created in the Concept Scheme definition by reference to the proper Concept Scheme and the School concept defined.

3.5 Modeling

In this phase of the CRISP-DM methodology, we describe a load of these RDF files into the graph database Neo4j. As our datasets are modeled in RDF, we need to set them up in the Neo4j graph. We used a plugin from Neo4j Lab called `n10semantics`, which enables the use of RDF in Neo4j. It allows us to store RDF data in Neo4j without losing a single triple, mapping, and inferencing. We installed the plugin and proceeded to initialize the graph with settings such as handling `Vocab-Uris` as shorten, overwriting multi-values, and handling RDF types as labels. The remaining settings remained in their default value. After the graph initialization,

```

<https://www.tec.mx/ontos/observation/DSQuinquennialSchoolPub/2022/EAAD/count>    rdf:type
qb:Observation ;
qb:dataSet <https://www.tec.mx/ontos/indicators/DSQuinquennialSchoolPub> ;
sdmx-attribute:unitMeasure <http://qudt.org/vocab/unit#Number> ;
sdmx-dimension:refPeriod <http://reference.data.gov.uk/id/quinquennium/2017-2021> ;
rdfs:label "number of publications (quinquennial) made in EAAD in 2022" ;
<https://www.tec.mx/ontos/dsd/cs/DSQuinquennialSchoolPub#school>
<https://www.tec.mx/ontos/conceptscheme/schools/concept/EAAD> ;
tec:NumberOfPublications 32
.

```

Figure 3.14: Example of multidimensional Scientometric Indicator Observation Definition

we proceeded to use a store procedure from the n10s plugin for importing the first RDF file containing the scientometric indicator. This store procedure is called `import.fetch`, and we call it from the browser terminal of the graph database. The procedure receives the location of the RDF file and the RDF format, in our case, Turtle. An example of the load is shown in Figure 3.15.

```

$ CALL n10s.rdf.import.fetch("file:G:///Mi unidad\Maestria\Tesis\Datos\RDF\2nd Row of RDF\1-PubQuinquenio.ttl","Turtle"...

```

terminationStatus	triplesLoaded	triplesParsed	namespaces
"OK"	59	59	<pre> { "ns0": "http://purl.org/linked-data/cube#", "rdfs": "http://www.w3.org/2000/01/rdf-schema#", "ns2": "http://purl.org/linked- data/sdmx/2009/dimension#", "ns1": "http://www.tec.mx/ontos/dsd/cs/DSQuinquennialPublication: "ns3": "http://purl.org/linked- data/sdmx/2009/attribute#" } </pre>

Figure 3.15: Example of loading an unidimensional Scientometric Indicator in Neo4j

This procedure is repeated for all scientometric indicators. We did it manually to ensure all the triplets were loaded correctly, but this procedure can be automated. After completing the list of scientometric indicators, our graph database is ready to be evaluated with some queries. In Figure 3.16 we show all the nodes and relationships stored in our graph database of scientometric indicators. Blue nodes represent observation nodes, i.e., indicator data points, hence predominating in the graph.

3.5.1 Neo4j Aura

After having the scientometric indicators loaded, we passed the graph, including configurations, to Neo4j Aura. Neo4j Aura is the cloud database version of Neo4j. We chose this

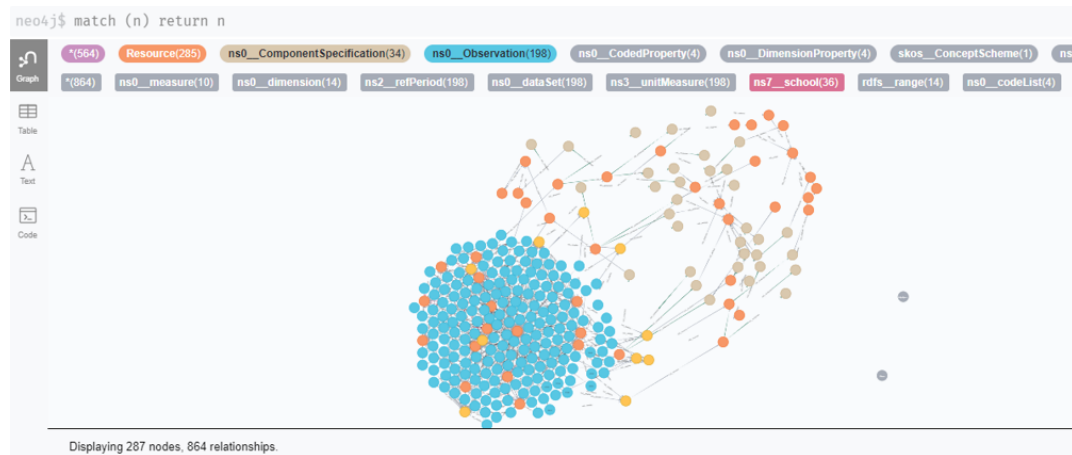


Figure 3.16: Scientometric Indicators loaded in Neo4j.

approach because it will allow us to better integrate with our chatbot in terms of connection, such as receiving query requests. This process is automated by Neo4j Desktop, where we have our current graph database. Information is integrated and secured for a proper load to Neo4j Aura.

3.6 Chatbot Construction and Integration

We built and integrated our scientometric indicator chatbot with our model in this section. We built a scientometric indicator API (SI API) that enables the communication between the Chatbot Framework in Azure and our model in Neo4j Aura. Technical details of the deployment are explained in a further section.

Our chatbot is classified in the closed domain of scientometric indicators. It is a task-based chatbot whose main task is to answer questions related to the mentioned domain. In order to answer the questions, the chatbot will follow a retrieval-based strategy that will use the ontology model to dig into data and return values in the form of answers.

3.6.1 Chatbot Self Knowledge

One of the advantages of the model that we just built is that it allows us to describe that model itself. For the chatbot, it is a convenient advantage because it can be used to inform the user with an insight into the information and prepare the user to ask a proper question about data. This section proposes a semantic methodology to demonstrate its advantages over traditional approaches. SDMX has demonstrated a consistent representation of multidimensional data, but it must permit to capture particular differences between indicator definitions (e.g., annual versus quinquennial time periods). Besides, we must assure that any person familiar with SDMX is capable of discovering, enquiring, and composing scientometric indicators encoded with this data model. We also must provide high-performance on query answering, so we used the Neo4j platform for this purpose. In the following list, we describe the processes of

this semantic approach:

- Describe from the ontology which scientometric indicators are modeled?
- What does the scientometric indicator measure?
- Which kind of time dimension has a certain scientometric indicator measure?
- How many dimension has every scientometric indicator?
- How to calculate a new scientometric indicator with existing data?

3.6.2 Natural Language Processing

This section of the methodology will analyze the natural language questions that the users may ask the chatbot. We decided to work with an expert from the Research Office at Tecnológico de Monterrey for this part of the process. The expert provided a list of frequently asked questions about scientometric indicators and the classification of each question into low, medium, and high complexity categories shown in Table 3.4.

The classification of the questions will allow us to detect the intention of the question and prepare the process to answer it. Depending on the complexity of the question, the categories are described as follows:

- Low Complexity: Simple questions with a direct answer.
- Medium Complexity: Questions that require calculations such as sum, average, difference, or further analysis.
- High Complexity: Questions that require machine learning models for prediction.

Intent identification

In this process of the methodology, we will identify the intention of the question to be able to answer it. Intent can be represented as a task or action the user wants to perform, and it can be considered as a purpose or goal. We choose to use an Artificial Intelligence service from Azure Cognitive Service called Language Understanding (Luis) for this step. This service applies custom machine learning intelligence to the user's natural language text to predict the meaning or pull out relevant information. One of the reasons for choosing this approach is the integrity of the chatbot framework of Azure.

The first step is to define a set of intents that will correspond to actions users want to perform in our chatbot. In Table 3.5 we show a couple of utterances (questions) with the intent classification defined for Luis.

We used the categorization done by the expert at the Research Office at Tecnológico de Monterrey by including them as intents. We also add the greeting and none intent. Greeting intent will be used to identify when the user wants to start a conversation in the chatbot, and the None intent is empty at purpose because we can identify questions out of the domain.

Table 3.4: Frequently asked Scientometric Indicators' questions.

Question	Complexity	Translated Question
Cuál es el Número de publicaciones en el quinquenio actual del Tec?	Low	Number of publications in the actual quinquennium at Tec?
Cuál es el Número de publicaciones en el quinquenio actual de la EIC?	Low	Number of publication in the actual quinquennium at EIC?
Cuál es el Número de publicaciones en el año actual del Tec?	Low	Number of publications of the actual year at Tec?
Cuál es el Número de publicaciones en 2020 de la EHE?	Low	Number of publication in 2020 at EHE?
Cuál es el avance de publicaciones 2021 para la EN?	Low	Number of publication in 2021 at EN?
Cuál es el Número de citas en el quinquenio pasado?	Low	Numer of Cites in the previous quinquennium?
Cuál es el avance de citas para este quinquenio?	Low	Number of cites in this quinquennium?
Cuales son las Citas por publicación de Tec?	Low	Number of cites per publication at Tec?
Cual es el número de citas por escuela?	Medium	Number of cites per school?
Cuantos SNIs en el Tec en 2021?	Low	How many researchers exist at Tec in 2021?
Cuantos SNIs hay en la escuela de Ingeniería (EIC)?	Low	How many researchers exist in the engineerng school (EIC)?
Cuantos Postdocs hay en el Tec en el último año?	Low	How many PosDocs existed at TEc in the last year?
Cuantos postdocs por escuela?	Medium	How many PosDocs exist per school?
Cuantos SNIs Nivel 1 hay en el Tec?	Low	How many researchers of level 1 exist at Tec?
En cuanto se ha incrementado la producción científica del Tec en el último año?	Medium	In what percentage has the scientific production at Tec increased in the last year?
En cuanto se ha incrementado las citas en el último quinquenio con respecto al anterior?	Medium	How was increased the number of cites comparing the actual quinquennium with the previous one?
Como ha crecido el número de SNIs por año?	Medium	How has increased the number of researches per year?
Que porcentaje de los SNIs son de Nivel2?	Medium	Percentage of level 2 researchers?
Cual es el pronostico de publicaciones en el 2022?	High	Forecast of number of publication in 2022?
Cuantas citas habra el proximo año en el Tec?	High	How many cites will Tec have the next year?

Table 3.5: Intent and Utterances Definition

Intent		Utterance examples	Utterance translation
Greeting		Hola	Hello
Greeting		Buen Día	Good Morning
Low	Com- plexity	Cuál es el numero de publicaciones quinquenales en 2020 de la EHE?	Number of quinquennial publication in 2020 at EHE?
Low	Com- plexity	Cuántos SNIS hay en el Tec en el 2019?	How many researchers existed at Tec in 2019?
Medium	Com- plexity	Cuál ha sido el año con el menor número de SNIS?	Which year had the lowest number of researchers?
Medium	Com- plexity	En cuánto se ha incrementado las publicaciones del Tec en el último año?	In what percentage has increased the number of publications in the last year?
High	Com- plexity	Cuál es el pronóstico de publicaciones en el 2025?	What is the forecast for number of publications in 2025?
High	Com- plexity	Cuántas citas habrá el próximo año en el Tec?	How many cites will Tec have the next year?
None		Empty	Empty

Entities extraction

This section of the methodology will extract information from the natural language question. Luis AI service will perform this process as it is a feature of this Artificial Intelligence service integrated into the Azure framework. Entities are created along intents used to extract relevant data from user utterances. An entity can be defined as words or phrases inside the utterance that describe important information of the intent and that are essential to perform the task of answering the question.

There are several types of entities in Luis. We choose the machine-learned entity type because it uses context to extract entities based on labeled examples. It is very useful for this case scenario as it performs well when entities can be found with variations in terms of format but still have the same meaning. In the following list, we show the created machine-learned entities:

- **Indicador:** Entity created for finding scientometric indicator names.
- **Lugar:** Entity created for finding a school, set of schools, or referring to Tecnológico de Monterrey that considers all the schools.
- **Objeto:** Entity created for obtaining the "thing" that we are looking for in medium complexity intentions.
- **Tiempo:** Entity created for finding years or time-related words.
- **Tipo Indicador:** Entity created for finding if the scientometric indicator refers to an annual or quinquennial type.

Table 3.6: Intent and Entities labeling

Intent	Utterance	ex-amples	Indicador	Lugar	Objeto	Tiempo	Tipo Indicador
Greeting	Hola		NA	NA	NA	NA	NA
Greeting	Buen Día		NA	NA	NA	NA	NA
Low Complexity	Cuál es el numero de publicaciones quinquenales en 2020 de la EHE?	publicaciones EHE			NA	2020	quinquenales
Low Complexity	Cuántos hay en el Tec en el 2019?	SNIS	SNIS	Tec	NA	2019	NA
Medium Complexity	Cuál ha sido el año con el menor número de SNIS?		SNIS	NA	Año	NA	NA
Medium Complexity	En cuánto se ha incrementado las publicaciones del Tec en el último año?	publicaciones Tec			incrementado	ultimo año	NA
High Complexity	Cuál es el pronostico de publicaciones en el 2025?	publicaciones		NA	pronóstico	2025	NA
High Complexity	Cuántas anuales habrá el próximo año?	citas	citas	NA	citas	próximo año	anuales
None	Empty		NA	NA	NA	NA	NA

The next step is to label entities in all the utterances in each intent type, as shown in Table 3.6. With the intent classification and entity labeling, we are ready to train the model.

Model training

The next step of the process is to train the model. Training is the process of teaching the model how to identify intents and extract entities from utterances. Training is performed in Luis AI service and is done iteratively. We start by randomly selecting five utterances with a different classification of intents and already having the entities labeling done in the previous step. We train the model and make some tests to observe the accuracy of correct identification and extraction. We continue doing this step iteratively until we reach the 50 utterances collected from the Research Office at Tecnológico de Monterrey, including several variations to the questions to achieve a better result.

Scientometric Indicator identification

In this methodology process, we will use the intent identification and entity extraction of the natural language question to identify the scientometric indicator that the user wants an answer about. The first step is to obtain descriptive information on the scientometric indicators, and we will obtain this by querying our ontology model as shown in Figure 3.17.



Figure 3.17: Query for obtaining Scientometric Indicators Dataset labels and description.

After matching all the nodes of the dataset type and returning all the `rdf:labels` and comments of the dataset, it will allow us to make a knowledge base of Indicators for the chatbot. The result of the query will be transformed from a JSON data type to a dictionary to use in a further process. The knowledge base of true scientometric indicators is shown in the following list, and it contains the description of the indicator and relevant tags extracted from the label node property.

- annual publications scopus-tec dataset: ['publication', 'annual']
- annual school publications dataset: ['school', 'publication', 'annual']
- quinquennial school publication dataset: ['quinquennial', 'school', 'publication']
- quinquennial school cites dataset: ['quinquennial', 'school', 'cite']
- annual school document cites dataset: ['cite', 'school', 'annual']
- researchers dataset: ['researcher', 'annual']
- posdocs dataset: ['posdoc', 'annual']
- quinquennial publications dataset: ['quinquennial', 'publication']
- quinquennial cites dataset: ['quinquennial', 'cite']
- document cites dataset: ['quinquennial', 'cite']

The next step works with a Language knowledge base that will allow us to transform the intention and entities extracted from the natural language question in Spanish to the English language to compare it against the true scientometric indicator knowledge base. This knowledge base has all the Spanish singular and plural variations of the indicator intention and school entity.

- cite: ['cita', 'citas', 'citaciones'],
- publication: ['publicacion', 'publicaciones'],
- researcher: ['investigador', 'investigadores', 'snis', 'sni'],
- posdoc: ['posdoc', 'posdocs', 'postdoc', 'postdocs'],
- quinquennial: ['quinquenio', 'quinquenios'],
- school: ['escuela', 'escuelas', 'eaad', 'ehe', 'en', 'ecsg', 'eic', 'emcs']

The following step is to send the natural language question to the Luis AI service to return the intent and entities of the natural question. For the intent, it will return a list with the best predictions, and we will take the one with the highest value. It returns all the entities extracted with their respective value for the entities. Intents and entities are stored in a dictionary data type, and they are lowered case for a complete comparison. If we had the following natural language question: *Cuántas publicaciones quinquenales se hicieron en el 2021?* (How many quinquennial publications were made in 2021?), by sending this question to Luis AI service, it will return the following response.

- topIntent: Low Complexity
- entities: indicador: 'publicaciones', tipo indicador: 'quinquenales', tiempo: '2021', Lugar: ''

The next step is to create a set by comparing the entity's response from Luis AI service with the Language knowledge base. This comparison analyzes if the value of the entities extracted by Luis AI, such as *Indicador*, *Lugar*, *Tiempo*, and *Tipo Indicador* exist in the knowledge base due to the language differences. It will assign a key with the English value in the new set called *Tags* if it exists.

At this moment, we have two sets. The first set is the true scientometric indicators, in which we query the data from the ontology model. This set contains the scientometric indicator description and the relevant labels. The second set is the *Tag Set*, in which we have all the processed data from entities extracted from the natural language question with the Luis AI service. We will use the *indicador*, *tipo indicador* and *school* entities values as labels from the tag set. In order to identify the correct scientometric indicator to which the natural language question is referring, we will work with both sets. Our approach for correct identification is to use the set theory.

- **Equal Sets:** When elements (labels) are the same members of True scientometric indicator and Tags Sets. Also called super sets.
- **Proper Subset:** When elements (labels) from Tag Set are included in the true scientometric indicator Set elements but still have other elements missing to be a Super Set.
- **None equal Sets:** Both sets have different elements.

For our approach, we will use the definition of the Equal set to establish that if the labels of both sets are the same, we will take the scientometric indicator from the true scientometric indicator, also called the Exact Indicator. It is important to say that it is possible to have only one match of the scientometric indicator in this specific scenario. The proper subset will be used for the contextual approach. We can obtain several proper subsets, called possible indicators, because they still have left some elements (labels) to be an exact indicator. The None equal sets are used when both sets have different labels and cannot be an option for possible indicators.

If we get an Exact Indicator, this will be used for querying the answer along with the entities extracted from the natural language question using the Luis AI service. If we don't obtain an exact indicator, we will use the list of possible indicators and ask the user if he is referring to one in the list and, if needed, ask for the pending entities. Suppose we don't get an exact indicator nor a list of possible indicators. In that case, we will respond to the user that the indicator he is asking in the question does not exist in our ontology domain.

Continuing with the example of the question: *Cuántas publicaciones quinquenales se hicieron en el 2021?* (How many quinquennial publications were made in 2021?). After passing on the comparison with both knowledge bases, we got the following Tag Set: indicador: 'publications', tipo indicador: 'quinquennial'. We proceed to apply the set theory, and we can state that this label value corresponds exactly equal to the scientometric indicator Quinquennial Publications as shown in Figure 3.18. The matched tags are highlighted in green, and the possible indicators with missing tags are highlighted in yellow and red.

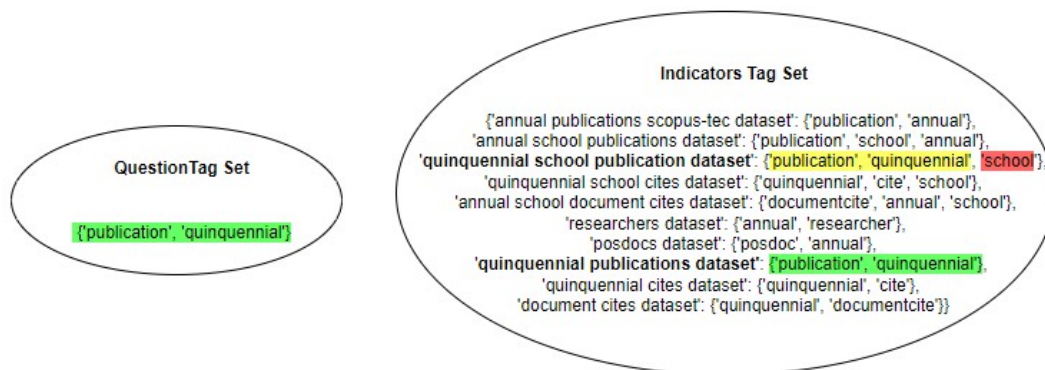


Figure 3.18: Example of matching Tag Set with Indicators' Set.

3.6.3 Natural Language transformation into Cypher Query

In this last step of the methodology, we will use the intent identification provided by Luis AI service. Depending on the complexity intention, a structured query is ready with some parameters to fulfill the entities extracted from the natural language question. Our approach for this chatbot will only focus on the low complexity intention.

The first step is to work with the time entity. We created a Time knowledge base that checks if time is given in a certain year or relative time. If the year is given numerically, the year number is used in the conditions of our query. If the year is given in relative time, we

will proceed to use the Time knowledge base that has several cases for identifying to which year does it relate as the following examples:

- Actual: Actual Year using get date function.
- Pasado, Ultimo, 1 año atrás]: Actual year - 1 year
- Antepasado, 2 años atrás: Actual year - 2 years

There are several variations included in this Time Knowledge base that will help us query the correct time of the indicator. The next step is to define a Measure knowledge base that extracts the name of the measures from our ontology model and is stated as follows:

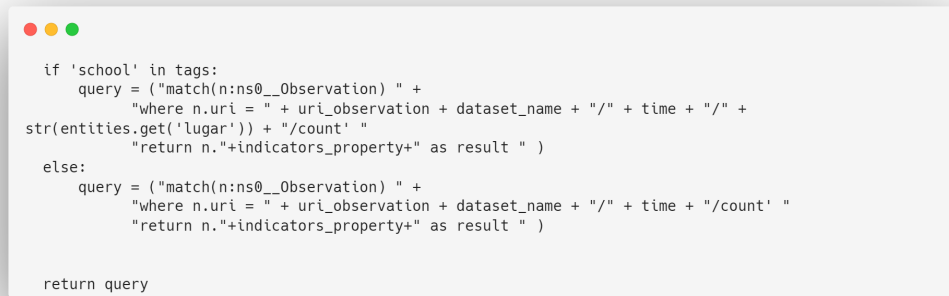
- cite: 'ns1__NumberOfCites',
- publication: 'ns1__NumberOfPublications',
- researcher: 'ns1__NumberOfResearchers',
- posdoc: 'ns1__NumberOfPosdocs'

This Measure knowledge base will allow us to query the correct measure from each indicator. It is important to mention that we just needed to query our model to obtain the correct description of the measures without any other previous knowledge.

The next step is to define the following parameters needed to run the query in our ontology model.

- uri_observation: Uses the following URI (<https://www.tec.mx/ontos/observation/>) defined in our ontology model for querying scientometric indicator observations.
- dataset_name: This value is obtained from the true scientometric indicator Set in which the descriptive value is selected by matching with the Exact Indicator key.
- time: Numerical value obtained after comparing with Time Knowledge-base, if the indicator is of type annual, we will only use a year. However, if it is quinquennial, we will use the range period of a year-(year-5).
- school: If the label school is in our tags, the structure changes, and we add the condition by obtaining the value of the entity Lugar.
- indicators_property: This value is obtained from the Measure knowledge base by matching the Exact Indicator key.

After filling up all the values of our parameters, we run the following query in Figure 3.19. This query matches all the nodes of type observation, and it filters for a specific unique URI that all the nodes have. This URI contains relevant information from which dataset, time, school, and measure are we filtering in order to answer the question. The query is run in our ontology model, and the result is displayed in the chatbot. If the ontology returns a numerical value, it answers the question, but if there is an empty response, some conditions are not met. We have to ask the user to establish the entities in the knowledge domain of our chatbot with the help of previous knowledge known as context.



```

if 'school' in tags:
    query = ("match(n:ns0__Observation) " +
            "where n.uri = " + uri_observation + dataset_name + "/" + time + "/" +
            str(entities.get('lugar')) + "/count' " +
            "return n."+indicators_property+" as result " )
else:
    query = ("match(n:ns0__Observation) " +
            "where n.uri = " + uri_observation + dataset_name + "/" + time + "/count' " +
            "return n."+indicators_property+" as result " )

return query

```

Figure 3.19: Natural Language Question into Cypher Query

3.7 Data Flow between Chatbot and Ontology

The proposed methodology aims to integrate the chatbot and the ontology model to complete the task of answering scientometric indicators. In Figure 3.20, we can observe the complete process in which a simple observation of an Indicator Dataset is transformed into RDF triples and then uploaded into the Neo4j graph database. The chatbot asks a question and is passed through natural language processes to identify the correct scientometric indicator and provide relevant information to answer the question. The natural language question is transformed into a query to the model. The ontology model returns a JSON response with important information, including the answer. Finally, it is processed by the chatbot to answer the question.

3.8 Deployment

This section explains deployment in detail by describing the important components and their integration to work as a system. The ontology modeling scientometric indicators in Neo4j Aura, the chatbot code, and the scientometric indicator API are stored in Azure Repositories at Tecnológico de Monterrey.

3.8.1 Chatbot Framework

In the previous steps, we set up the back-end of the proposed solution. The next step is to build the chatbot in order to be able to take the benefits of the ontology generated. A chatbot is a conversational agent that allows the user access to information and services through natural language dialogue, including text and voice [32]. The chatbot will provide natural language processing to obtain the question's relevant information, such as the intent and its entities. The chatbot must be capable of identifying the indicator been asked for, extracting the parameters of the query (dimensions), and building the Cypher query.

Several chatbot frameworks are available to work with, such as Google Dialogflow, IBM Watson, and Azure Bot Service.

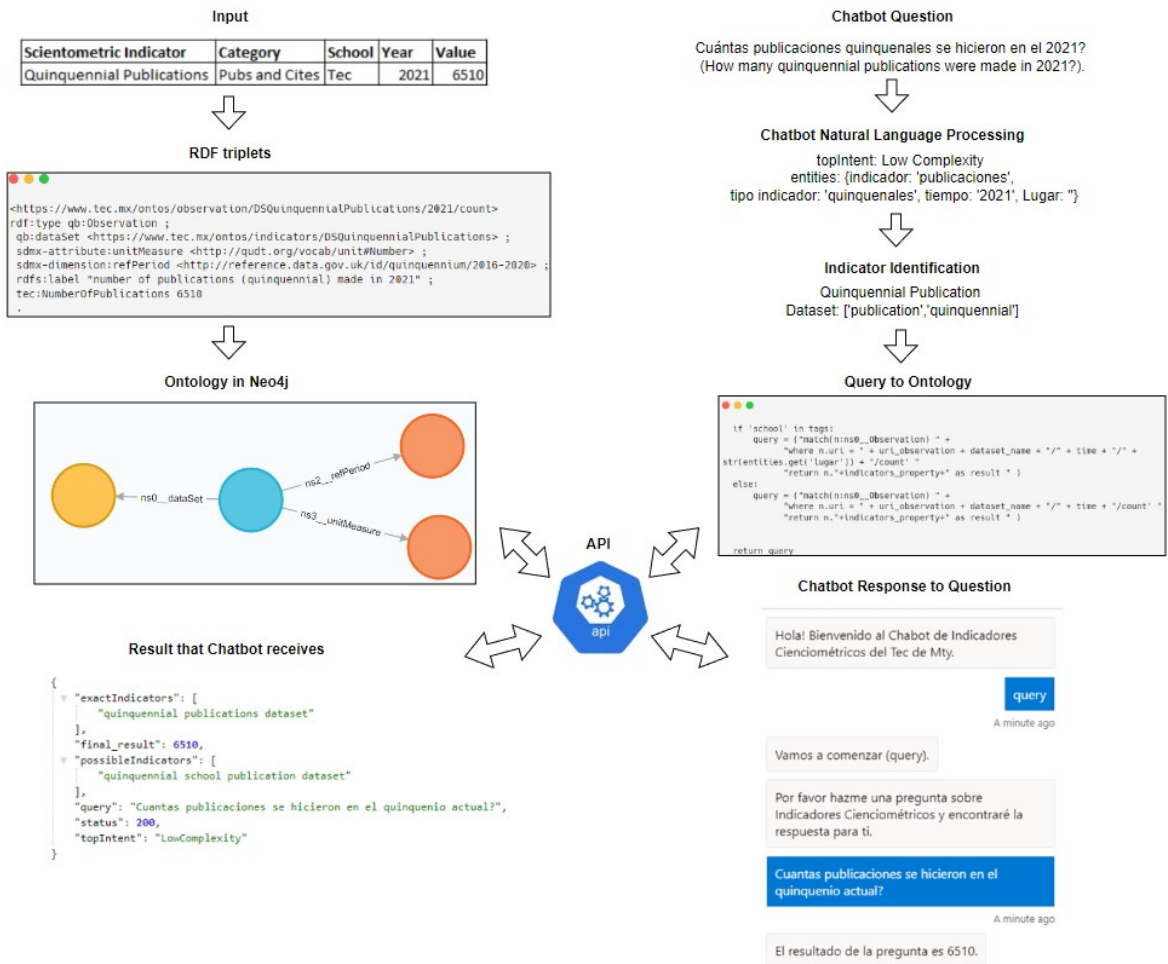


Figure 3.20: Data Flow from Input to Chatbot response

- Dialogflow is a natural language understanding platform that allows to design and integrate a conversational user interface [3].
- IBM Watson is a data analytics processor that uses natural language processing, a technology that analyzes human speech for meaning and syntax [7].
- Azure Bot Service provides an integrated development environment for bot building. It allows you to create bots using a low code graphical environment providing telemetry [1].

After comparing the mentioned frameworks, we chose to work with Azure Bot Service. It provides a scalable, integrated bot development and hosting environment for intelligent bots. It is an open-source Software Development Kit (SDK) with different tools that help us build, test, and publish our bot. It improves interaction with users by integrating Cognitive Services and allows us to deploy our bot to popular channels. Cognitive Services is a comprehensive family of AI services and cognitive APIs to build intelligent applications. Another important factor for our decision is that at Tecnológico de Monterrey. The complete Microsoft suite is

used in this institution, where the study case is presented. Working with this framework will add the benefits of data and process integration, user's relation with Azure, and cost.

3.8.2 Bot Framework Composer

The first step to start building the chatbot for scientometric indicators is to use a tool provided by the Azure Bot Service called Bot Framework Composer. It is an open-source IDE for developers to author, test, provision, and manage conversational experiences. It provides a powerful visual authoring canvas enabling dialogs, language-understanding models, QnA-Maker knowledge bases, and language generation responses [8].

The chatbot runs through a flow diagram in which two main concepts are Dialog and Triggers. Dialogs are a central concept in the SDK, providing ways to manage a long-running conversation with the user. It performs a task that can represent part of or a complete conversational thread, and it can span just one turn or many and span a short, or long time-period [4]. Dialog triggers handle dialog-specific events that are related to the life-cycle of the dialog [5].

Main Dialog

The main dialog is the first process that the chatbot runs. It is composed of a task in which the chatbot greets and asks the user their email, then it stores the email for log purposes. The next step is that it displays the Options Dialog. In Figure 3.21 we observe the chatbot view along its logic process, which produces the task of the chatbot greeting the user.

Options Dialog

The Options Dialog sends a response to the user as a thumbnail card. The thumbnail card displays a thumbnail image with several buttons needed for the conversation flow. We choose to implement the procedural flow in which the chatbot is focused on a task to make the bot achieve its main goal: answering questions about scientometric indicators. We define several options to have a centralized logic and guide the user to ask questions.

After prompting the user's email, the chatbot will display a menu with the following options: Conocer Indicadores, Conocer Métricas, Preguntar, and Salir.

In Figure 3.22, we can observe that the displayed menu shows additional information such as the date on which the ontology was loaded and the objective and logo of the chatbot. The following step is that the user selects an option. The options in the menu are listed in a way in which the user follows an order in which he can gather relevant information before making the questions, such as the indicators and metrics known by the chatbot.

Informative Dialogs

When the displayed menu is shown in the chatbot conversation, see Figure 3.23, and the user clicks on the Conocer Indicadores or Conocer Métricas Options, an automatic response is sent to the conversation, and a trigger is activated. The trigger recognizes the response and displays the dialog. The goal of these dialogs is to introduce to the user the indicators and metrics known by the chatbot with the loaded ontology. In 3.23 we observe the complete

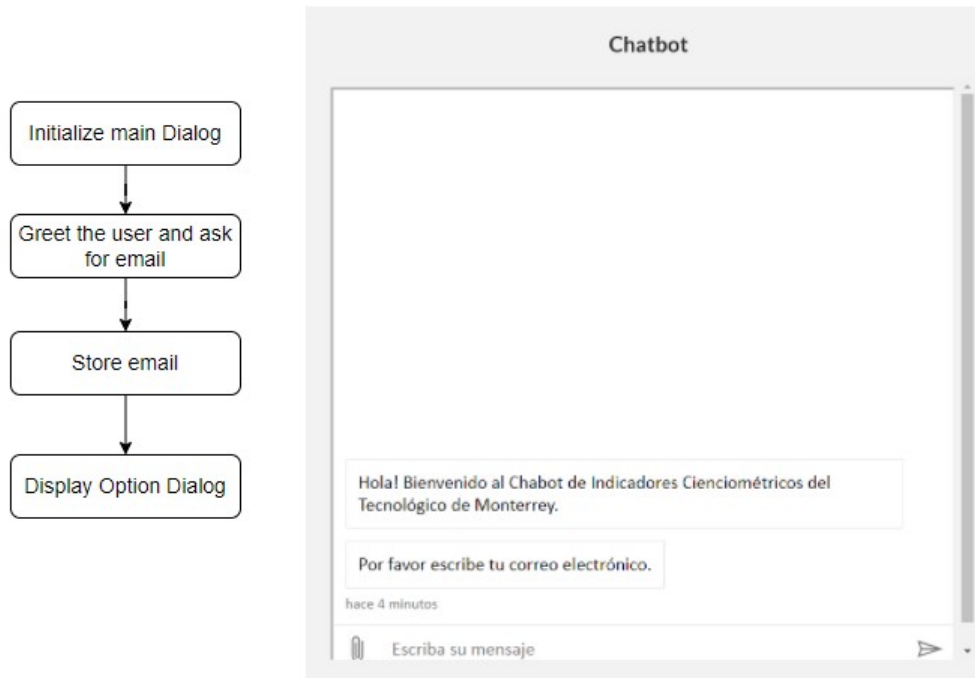


Figure 3.21: Greeting Dialog Flowchart and Chatbot View

flowchart of these dialogs in which several components such as Chatbot Framework, SI API, and Neo4j work together to achieve the goal.

In the first step, the chatbot sends an HTTP get request method to the scientometric indicator Application (SI API). We will explain the SI API in more detail in the following section. The SI API receives the petition and connects to the Neo4j database, where the ontology is loaded. The SI API performs a query, depending on the selected option, to the ontology shown in Figure 3.24 where we look for all the nodes of type Dataset and return their description in the form of list structure to extract the scientometric indicators. For the metrics retrieval, we query all the nodes of type Measure and return their label for extracting the metrics of the scientometric indicators.

The data contained in the list is processed and sent back in a JSON structure. The JSON is received by the task of the dialog along with a status code. The next step is to validate the status code, and if it is correct, the JSON content is stored in a variable inside the dialog. The chatbot extracts the information obtained from the ontology and displays a list of the known indicators. This dialog aims to introduce the user to the knowledge the chatbot knows in terms of indicators and metrics.

In Figure 3.25 we can observe the Chabot views by clicking on the menu options of Conocer Indicadores and Conocer Métricas. We return a list of indicators or metrics in both views, depending on the selection. This retrieval of information is important for the user to know the chatbot's knowledge to ask proper questions about scientometric indicators. After sending the response, the bot displays the menu again to continue with the conversation. At this step, the user is ready to ask a question.

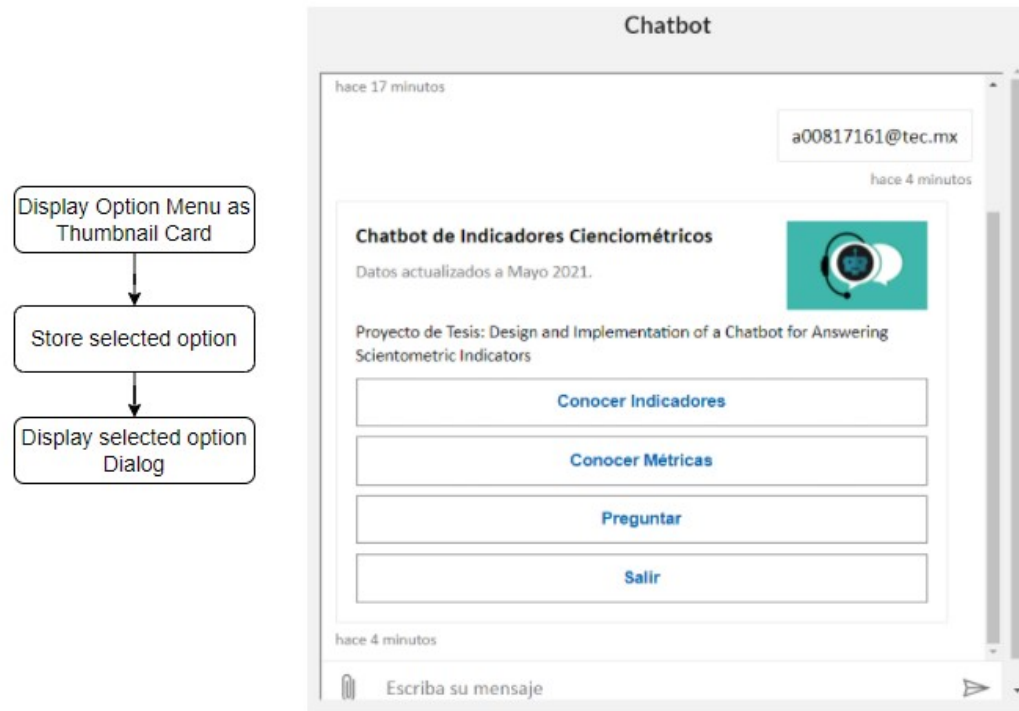


Figure 3.22: Option Dialog Flowchart and Chatbot View

Ask Question Dialog

When the displayed menu is shown in the chatbot conversation, and the user clicks on the Preguntar Option, an automatic response is sent to the conversation, and a trigger is activated. The trigger recognizes the response and displays the dialog. The goal of this dialog is to have a conversation with the user to provide the service of answering questions. The first step in this process is that the chatbot asks the question to the user and after the user prompts the question in a natural language way, the chatbot stores the question. Then it connects with the scientometric indicator API sending the question and user as parameters. The next step is to connect to the Neo4j graph database, where the ontology model is stored. We also connect to Luis AI service by sending the natural language question and having the intent and entities from the question in return. Then we query the ontology model to have the available indicators with their respective labeling tags stored with the RDF label property. We proceed to compare the entities using the set theory and identify the scientometric indicator of the question along with the entities extracted from the context of the question. All the data is stored to convert the natural language question into a Cypher query to execute it in the ontology model. Finally, a result is returned in JSON from the Bot Framework Composer and is shown to the user in the conversation. The chatbot asks the user what the next task is by prompting the display menu. In Figure 3.26 we can observe the complete flowchart of the dialog in which all the components such as Chatbot framework, API, and Neo4j work along to achieve the goal of the process that is answering questions.

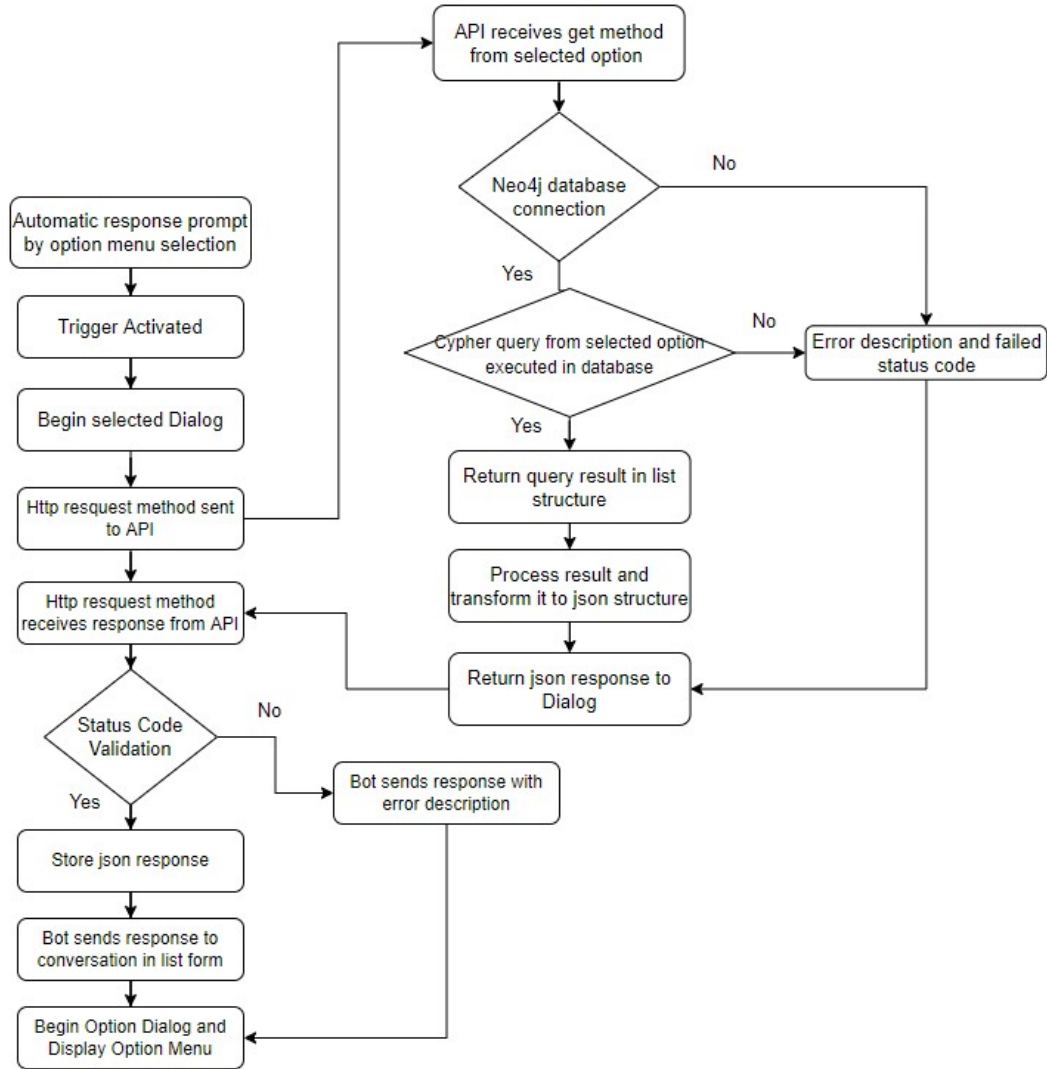


Figure 3.23: Informative Dialog Flowchart

In Figure 3.27 we observe an example of a question asked about scientometric indicators to the chatbot. The user proceeded to click on the Preguntar Option from the displayed menu, and the chatbot prompted a response to mention to the user to prompt the question. After the question is written, the chatbot takes the question and follows the methodology process to answer the question. The result value is given to the chatbot framework, and the response is shown to the user.

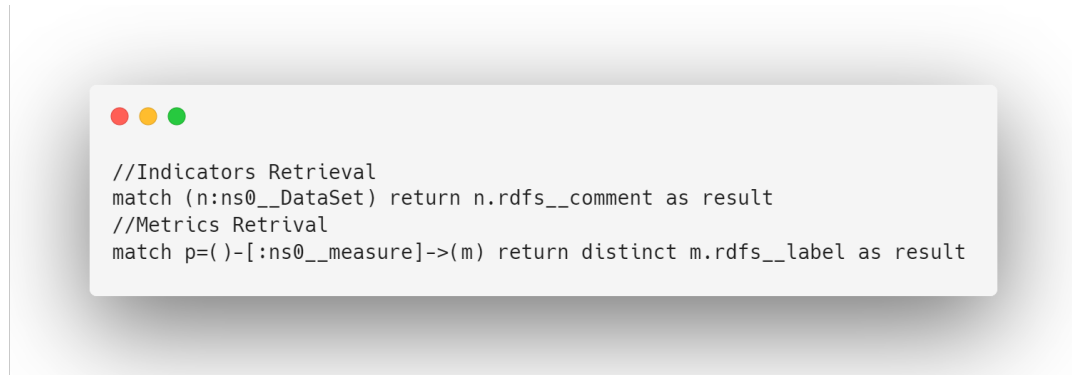


Figure 3.24: Informative Dialog Flowchart Query

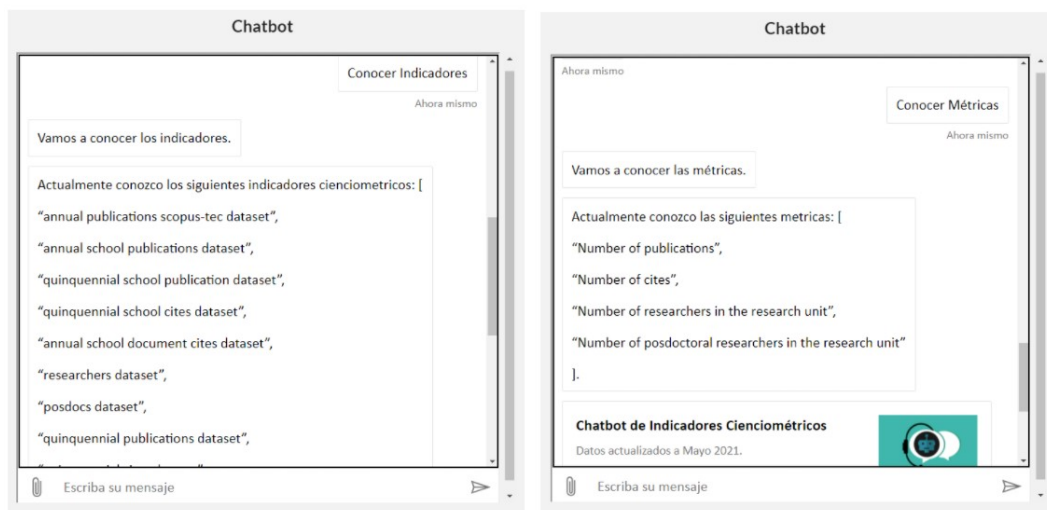


Figure 3.25: Informative Dialog Chatbot View

3.8.3 Scientometric Indicator API

The scientometric indicator API was created in python language using the Flask framework for faster development. The goal of the API is to connect several components such as the ontology model in Neo4j, the chatbot framework, and the Luis AI service for intent identification and entity extraction.

3.8.4 Chatbot Hosting

The process of hosting the chatbot for the testing period was achieved using Google Sites for sharing the chatbot with the users from the Research Office at Tecnológico de Monterrey.

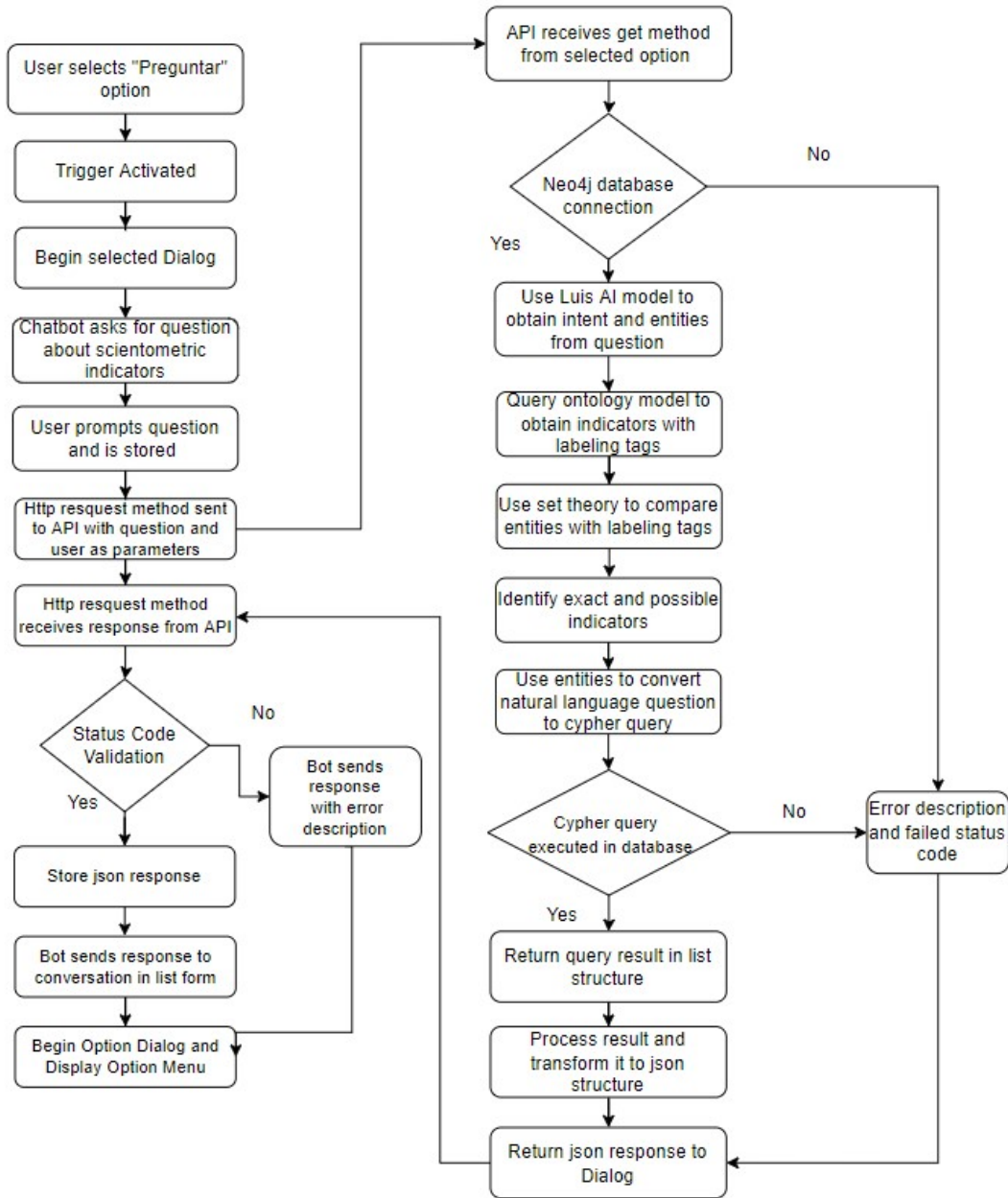


Figure 3.26: Asking-Question Dialog Flowchart

3.9 Evaluation

The evaluation procedure is going to be divided into four main stages.

3.9.1 Ontology Self-Descriptive Knowledge Evaluation

The first stage consists of the ontology of self-descriptive knowledge. In this stage, we want to evaluate our stage by demonstrating that including SDMX in our data model is enough for

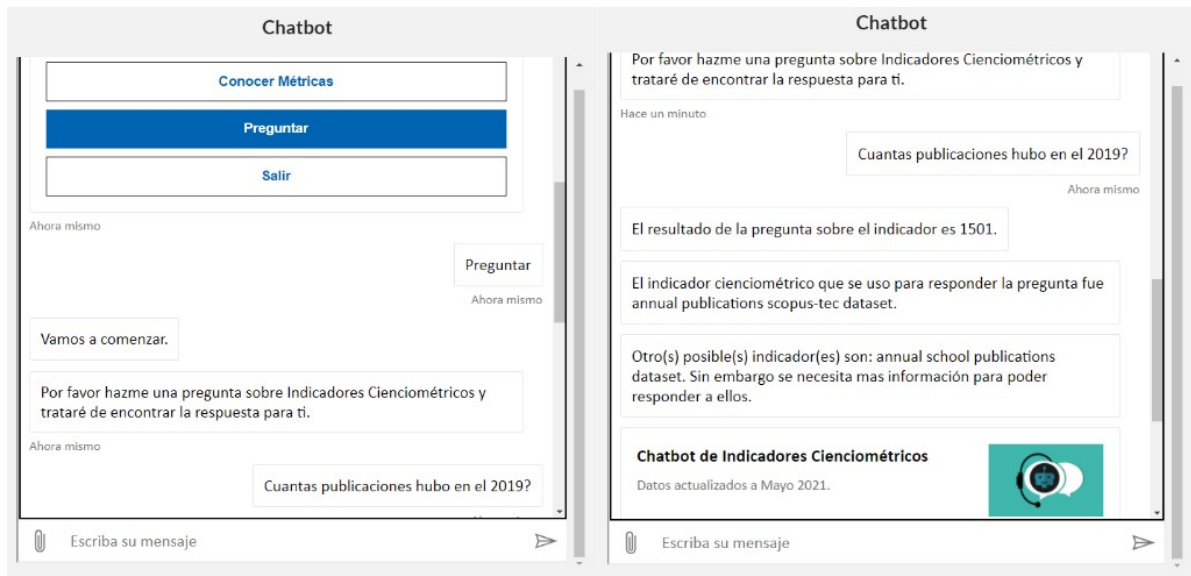


Figure 3.27: Asking-Question Dialog Chatbot View

enabling indicator discovery, enquiring, retrieval and composition. To do so, we define four queries using Cypher language. This evaluation aims to state that our model can provide self-descriptive knowledge about scientometric indicator structure, definition, and description.

3.9.2 Query Complexity Evaluation

The second evaluation will demonstrate that querying from low to high complexity levels can be performed in our model. We will define four queries that will increase their level of complexity. We will evaluate with the following approaches: unidimensional indicator single step calculation, the use of several dimensions for multidimensional indicators, calculation of scientometric indicator using averages, sum, min, or max, among other aggregation functions, and finally, the creation of new scientometric indicators using existing data.

3.9.3 Ontology Storage Optimization in Neo4j

In the third evaluation, we will compare the number of nodes and relationships in our Neo4j graph database against the number of triplets uploaded to a graph database using RDF files in an Apache Jena Fuseki server to analyze the complexity of the data structure.

3.9.4 Chatbot Test Evaluation with Users

The last evaluation will consist of a test evaluation with users from the Research Office at Tecnológico de Monterrey. We host the scientometric indicator chatbot on the web for this evaluation and make it available for the users for testing. The users will have three weeks to test several natural language questions on the chatbot and, in the end, evaluate it with a survey that will contain the following questions with answers from 1 to 5. 1 meaning

Minimal Knowledge or Deficient and five meaning Expert or Excellent depending on the type of question.

- user's email
- User's knowledge about scientometric indicators
- Usability: Ease of use and time required for the chatbot to answer the questions.
- Strictness: Ability of the chatbot to understand language variations.
- Comprehension: Ability of the chatbot to understand the question and answer due to the relevant information extracted.
- Correlation: Relevance of the questions according to the context of the question.
- Satisfaction: User's feeling with the chatbot and its future.
- Comments or Feedback

We will log all the users' interactions with the chatbot along with this survey. With these logs, we will evaluate the following metrics:

- Structure of the Conversation: Number of users and Total Conversations.
- Goal Completion Rate: Number of times chatbot answered correctly, number of correct intention detected, correct indicator detected, and correct entity extraction.
- Bot Response Time: Comparison between the time taken to answer a question of the chatbot and the actual process.

Chapter 4

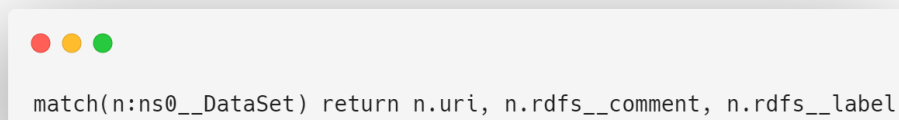
Results

4.1 Ontology Self-Knowledge

This section will evaluate our approach by demonstrating that knowing the SDMX data model is enough for enabling indicator discovery, enquiring, and composition. To do so, we define and test five queries using the Cypher language.

4.1.1 Indicator Retrieval

The first evaluation answers the following question: Which scientometric indicators are available in our ontology model? This Cypher query can also be named Indicator Retrieval. Using the concept of Dataset, we will query the nodes of dataset type and query its properties shown in Figure 4.1. In the properties, we can find the URI, a unique identifier of the node, the description or name of the dataset, and the labels that are relevant tags for proper identification in other processes. The result of this query, shown in Table 4.1, returns the ten scientometric indicators loaded in the ontology with their respective description. Due to the prefixes concept defined in our ontology, we can present a better readable result because it automatically substitutes the URI (<https://www.tec.mx/ontos/indicators/>) for tec.



```
match(n:ns0__DataSet) return n.uri, n.rdfs__comment, n.rdfs__label
```

Figure 4.1: Indicator Retrieval Query

Table 4.1: Result for query: Which scientometric indicators are in the model?

URI	rdfs:comment	rdfs:label
tec:DSAnnualPubScopusTec	annual publications scopus-tec dataset	[publication,annual]
tec:DSSchoolPub	annual school publica- tions dataset	[school,publication,annual]
tec:DSQuinquennialSchoolPub	quinquennial school publication dataset	[quinquennial,school,publication]
tec:DSQuinquennialSchoolCites	quinquennial school cites dataset	[quinquennial,school,cite]
tec:DSDocumentSchoolCites	annual school docu- ment cites dataset	[documentcite,school,annual]
tec:DSResearchers	researchers dataset	[researcher,annual]
tec:DSPosDocs	posdocs dataset	[posdoc,annual]
tec:DSQuinquennialPublications	quinquennial publica- tions dataset	[quinquennial,publication]
tec:DSQuinquennialCites	quinquennial dataset	[quinquennial,cite]
tec:DSDocumentCites	document cites dataset	[quinquennial,documentcite]

4.1.2 Measure Discovery

By knowing that indicators are encoded as SDMX datasets, we can ask for what is measured, as shown in the Cypher query in Figure 4.2. This query inspects the structure of all the indicators by matching the dataset with the data structure definition. We dig into the components to list the available metrics (measures). The result of this query, shown in Table 4.2, has the four measures available in the indicator sample: counts of publications, citations, researchers, and postdoc researchers.

```

match (ind:ns0__DataSet)-[:ns0__structure]->(struct:ns0__DataStructureDefinition)
match (struct)-[:ns0__component]->(comp)
match (comp)-[:ns0__measure]->(m)
return distinct m.uri

```

Figure 4.2: Measure Discovery Query

Table 4.2: Result for query: Which measures (metrics) are available in the model?

	Measure URI
1	tec:NumberOfPublications
2	tec:NumberOfCites
3	tec:NumberOfResearchers
4	tec:NumberOfPosdocs

4.1.3 Dimension Arrangement Query

Next, we asked for the number of dimensions used for describing each indicator in Figure 4.3. This query inspects the dataset's structure with the data structure definition. Afterward, we dig into the components and get the number dimensions. In this way, we can distinguish between the indicator's dimensions in the result of the query shown in Table 4.3.

```

match (ind:ns0__DataSet)-[:ns0__structure]->(struct:ns0__DataStructureDefinition)
match (struct)-[:ns0__component]->(comp)-[:ns0__measure]->(m)
match (struct)-[:ns0__component]->(comp_d)-[:ns0__dimension]->(dim)
return ind.uri, count(dim) as dimension
return distinct m.uri

```

Figure 4.3: Dimension Arrangement Query

4.1.4 Indicator Composition

In this evaluation, we will select the measure of the number of publications to know the indicators' composition. Once we select the measure, we can investigate how it is reported, i.e., broken down. For this, we need to investigate which dimensions have reported the measure. The Cypher query shown in Figure 4.4 extends the previous query by selecting only those indicators that report publication counts and adding the dimensions associated with it. The results in Table 4.4 show that publications are reported in annual or quinquennial periods and at an institutional level or broken down by school.

4.2 Query Complexity Evaluation

This section will proceed to evaluate several queries about scientometric indicators. These queries were chosen by simulating frequently asked questions performed in the Research Office at Tecnológico de Monterrey.

Table 4.3: Result for query: How many dimensions has every indicator?

	Dataset URI	Dimension Count
1	tec:DSSchoolPub	2
2	tec:DSQuinquennialSchoolPub	2
3	tec:DSQuinquennialSchoolCites	2
4	tec:DSDocumentSchoolCites	2
5	tec:DSResearchers	1
6	tec:DSPosDocs	1
7	tec:DSQuinquennialPublications	1
8	tec:DSQuinquennialCites	1
9	tec:DSDocumentCites	1
10	tec:DSAnnualPubScopusTec	1

Table 4.4: Result for query: How is measured the number of publications?

	Indicator URI	Dimension URI
1	tec:DSAnnualPubScopusTec	tec:Annual
2	tec:DSQuinquennialPublications	tec:Quinquennial
3	tec:DSQuinquennialSchoolPub	tec:Quinquennial
4	tec:DSQuinquennialSchoolPub	tec:School
5	tec:DSSchoolPub	tec:Annual
6	tec:DSSchoolPub	tec:School

```

match (ind:ns0__DataSet)-[:ns0__structure]->(struct:ns0__DataStructureDefinition)
match (struct)-[:ns0__component]->(comp)-[:ns0__measure]->(m)
match (struct)-[:ns0__component]->(comp_d)-[:ns0__dimension]->(dim)
where (m.uri = 'https://www.tec.mx/ontos/indicators/NumberOfPublications')
return ind.uri, dim.uri
order by ind.uri, dim.uri

```

Figure 4.4: Indicator Composition from measure drill through Query

Table 4.5: Result for query: How many quinquennial cites where produced in 2021 in the EHE school?

	Interval URI	Number of Cites
1	interval:quinquennium/2016-2020	1057

4.2.1 Scientometric Indicator retrieval in a specific year period

The first evaluation consists in answering the following question: How many quinquennial cites were produced in 2021 in the EHE school? In this cypher query shown in Figure 4.5, we use 2 types of nodes: Observation and Resource. In the node of type Observation, we will find the values of measures in the properties of each node. We also query a relationship between these nodes (Observation and Resource) called refPeriod, which indicates that these nodes are related to the time dimension we are querying. We filter the year using the property URI of 2021 in the node of type resource. In this filter, we use the quinquennial intervals defined in the ontology model using (<http://reference.data.gov.uk/>). After running the query, we can state the result shown in Table 4.5 that 1057 cites were produced in 2021 using the quinquennium interval from 2016 to 2020.

```

match (obs:ns0__Observation)-[:ns2__refPeriod]->(period:Resource)
where obs.uri = 'https://www.tec.mx/ontos/observation/DSQuinquennialSchoolCites/2021/EHE/count'
and period.uri = 'http://reference.data.gov.uk/id/quinquennium/2016-2020'
return period.uri, obs.ns1__NumberOfCites

```

Figure 4.5: Scientometric Indicator retrieval in a certain year period Query

Table 4.6: Results for query: Show the number of publications made by the institution, reported in quinquennial periods.

	Interval URI	Number of Publications
1	interval:quinquennium/2011-2015	2958
2	interval:quinquennium/2012-2016	3334
3	interval:quinquennium/2013-2017	3891
4	interval:quinquennium/2014-2018	4518
5	interval:quinquennium/2015-2019	5369
6	interval:quinquennium/2016-2020	6510
7	interval:quinquennium/2017-2021	5811

4.2.2 Scientometric Indicator retrieved from a time interval range

The second evaluation is about retrieving the values stored in a specific indicator. We want to answer the following question: Show the number of publications made by the institution, reported in quinquennial periods. Figure 4.6 shows the Cypher query in which we use the Observation and Resource type. These nodes have a relationship of type `refPeriod`, which allows us to filter through the quinquennium interval. We also use the dataset relationship between the observations to specify that we filter the Quinquennial Publications Dataset. The results, shown in Table 4.6, include the quinquennial periods and the corresponding number of publications. As in the previous evaluation, intervals are retrieved from the vocabulary defined in the ontology model.



Figure 4.6: Scientometric Indicator retrieved from a time interval range Query

4.2.3 Aggregation functions of the Number of Publication of all Schools in a quinquennium interval

In the third evaluation, we calculate the number of schools and aggregation functions such as sum, average, standard deviation, min, and max number of quinquennial publications of

Table 4.7: Results for query: Calculate the number of schools and aggregation functions such as sum, average, standard deviation, min and max number of quinquennial publications of all the schools from Tecnológico de Monterrey in 2020.

	School Count	Sum	Average	Standard Deviation	Min	Max
1	6	5782	963.66	1345.21	26	3622

all the schools from Tecnológico de Monterrey in 2020. Using the same strategy as in previous Cypher queries, we match a node of type observation to a node of type resource (year dimension) by the relationship called refPeriod. We filter the year using the property URI with the quinquennium interval from 2020 in the period node of the type resource. Instead of returning a set of nodes and relationships, we return the calculation of the aggregation functions of the measures in the properties of the nodes of type observation shown in Figure 4.7. The results from this query, shown in Table 4.7, state that we have six schools currently at this evaluation period, the number of quinquennial publications is 5782, and the average is 963.66, the standard deviation is 1345.21, while the minimum and maximum values are 26 and 3622 respectively.

```

match (obs:ns0__Observation)-[:ns0__dataSet]->(ind)
match (obs)-[:ns2__refPeriod]->(period:Resource)
where ind.uri = 'https://www.tec.mx/ontos/indicators/DS0quinquennialSchoolPub'
and period.uri = 'http://reference.data.gov.uk/id/quinquennium/2015-2019'
return count (obs.ns1__NumberOfPublications[0]),sum(obs.ns1__NumberOfPublications[0]),
avg(obs.ns1__NumberOfPublications[0]),stDev(obs.ns1__NumberOfPublications[0]),
min(obs.ns1__NumberOfPublications[0]), max(obs.ns1__NumberOfPublications[0])

```

Figure 4.7: Aggregation functions of the Number of Publication of all Schools in a quinquennium interval Query

4.2.4 Calculation of a new Scientometric Indicator: Cites per Document

The last evaluation is a Cypher query that calculates a new scientometric indicator with information available in our graph database. As we already have the scientometric indicators of Quinquennial Publications and Quinquennial Cites, we can calculate a new scientometric indicator called Cites per Document by dividing the number of cites by the number of publications in Neo4j. We evaluated the capability of our approach for calculating a new indicator from those currently stored. Figure 4.8 shows how both quinquennial publications and quinquennial citations are retrieved for calculating a scientific impact indicator: citations per publication. The results in Table 4.8 show the time interval, the number of publications (docs),

Table 4.8: Results Cypher query: Calculate cites per publication?

	Interval URI	Docs	Citations	cites_per_doc
1	interval:quinquennium/2011-2015	2958	6682	2.2580
2	interval:quinquennium/2012-2016	3334	9759	2.9271
3	interval:quinquennium/2013-2017	3891	12311	3.1639
4	interval:quinquennium/2014-2018	4518	18393	4.0710
5	interval:quinquennium/2015-2019	5369	22943	4.2732
6	interval:quinquennium/2016-2020	6510	33941	5.2136
7	interval:quinquennium/2017-2021	5811	25034	4.3080

the number of cites (citations), and the number of citations per publication (cites_per_doc). We had to cast both publications and citations to float to make a correct calculation.

```

match (doc:ns0__Observation)-[:ns0__dataSet]->(ind_doc)
match (doc)-[:ns2__refPeriod]->(period)
match (cites:ns0__Observation)-[:ns0__dataSet]->(ind_cites)
match (cites)-[:ns2__refPeriod]->(period)
where ind_doc.uri = 'https://www.tec.mx/ontos/indicators/DSQuinquennialPublications'
and ind_cites.uri = 'https://www.tec.mx/ontos/indicators/DSQuinquennialCites'
with doc.ns1__NumberOfPublications[0] as docs, cites.ns1__NumberOfCites[0] as citations,
period.uri as quinq
return quinq, docs, citations, toFloat(citations)/toFloat(docs) as cites_per_doc
order by quinq

```

Figure 4.8: Calculation of a new Scientometric Indicator: Cites per Document Query

4.3 Ontology storage optimization in Neo4j

This section compares the number of nodes and relationships in our Neo4j graph database against the number of triplets uploaded to a graph database using RDF files in an Apache Jena Fuseki server to analyze the data structure complexity. In order to observe the behavior of our graph structure in Neo4j, we decided to make a comparison of the number of nodes and relationships in Neo4j against the number of triplets in a default graph database created in Apache Jena Fuseki server with all the scientometric indicator RDF files created for this work. Both graphs received as input the 10 RDF files of scientometric indicators. The Neo4j graph database produced 287 nodes and 864 relationships, while the RDF graph has 1578 triplets.

Table 4.9: Data Evaluation.

	Intent	Questions	Correct Intention	Correct Entities	Correct Indicator	Available Data	Correct Answer
1	LowComplexity	12	Yes	Yes	Yes	Yes	Yes
2	Greeting	4	No	Yes	No	No	No
3	LowComplexity	4	Yes	No	No	No	No
4	LowComplexity	3	Yes	No	Yes	Yes	Yes
5	Greeting	2	No	No	No	No	No
6	MediumComplexity	2	Yes	Yes	No	No	No
7	Greeting	1	No	Yes	Yes	Yes	Yes
8	LowComplexity	1	Yes	Yes	No	No	No
9	LowComplexity	1	Yes	Yes	No	Yes	Yes
10	MediumComplexity	1	No	No	No	No	No
11	MediumComplexity	1	Yes	No	No	No	No
12	MediumComplexity	1	No	Yes	No	No	No
13	MediumComplexity	1	No	Yes	Yes	No	No
14	None	1	Yes	No	No	No	No
15	Total	35	71.42%	65.71%	48.57%	48.57%	48.57%

4.4 Chatbot Test Evaluation with Users

4.4.1 Log Evaluation

We tested the chatbot with users from the Research Office at Tecnológico de Monterrey for the Log Evaluation. A total of 11 users participated in this test in which they made 35 different questions about scientometric indicators to the chatbot. Section 3.9 in Chapter 3 details the implementation made for testing the chatbot with users. Whenever the users asked the chatbot, relevant information such as the date-time, user, top intent, exact indicator, comments, answer, and list of possible indicators were retrieved and stored in our ontology model. This section will evaluate the Goal Completion Rate (GCR) in answers, intent detection, entity extraction, and scientometric indicators identification. The following data are shown in Table 4.9 which contains information extracted from the log that was manually classified according to the correct solution that the chatbot should have provided will allow us to evaluate the GCR. It is crucial to notice that each column represents a process that should have a positive outcome to answer the question correctly, measured in the last column.

Goal Completion Rate

The first evaluation is the GCR of the correct intention detection. Intent detection is a great resource for correctly retrieving the scientometric indicator value. In Table 4.10 we can observe that 21 questions were identified as Low Complexity intention. Users made six questions of Medium complexity, and finally, eight questions were identified with the None and Greeting

intentions. The GCR evaluates how many intentions were correctly identified by the chatbot. In Figure 4.9 we can observe that 25 intentions, that represent 71.42% from the total questions, were identified correctly.

Table 4.10: Intent Identification

	Intent	Number of Questions	Correct Intent
1	Low Complexity	21	21
2	Medium Complexity	6	3
3	Greeting	7	0
4	None	1	1

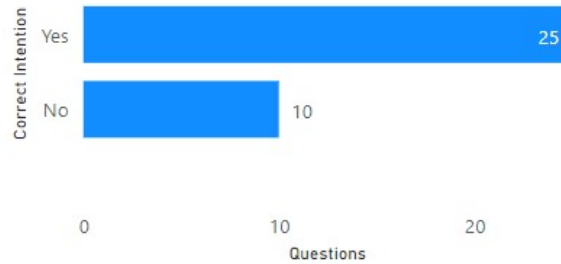


Figure 4.9: Goal Completion Rate: Correct Intention

The second evaluation assesses the GCR of the correct identification of scientometric indicators. In Figure 4.10 we can observe that the chatbot identified the scientometric indicator correctly in 17 questions made by the users. This outcome represents 48.6% of the total questions.

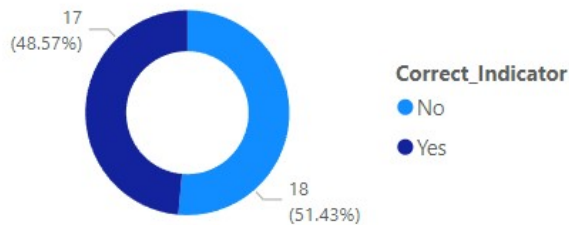


Figure 4.10: Goal Completion Rate: Correct Indicator

The next evaluation is about the GCR from entity extraction. Entity extraction allows us to find relevant data for retrieving the scientometric indicator value more precisely. In Table 4.11 we can observe that 23 questions entities were extracted correctly and it represents 65.71% from all the question.

Table 4.11: Entity Extraction

	Number of Questions	Correct Entity
1	23	Yes
2	12	No

In order to evaluate the GCR for correct answers from the chatbot to the questions, we need to establish the following categories.

- 1: Answered correctly with the required value
- 2: Answered incorrectly with a value or could not understand the question due to indicator matching, intent identification, or entity extraction.
- 3: The chatbot understood the question correctly but could not answer the question because data was not available in the ontology. In other words, the value for that time (year or quinquennium) is not stored in the ontology.

In Table 4.12 we can observe that the chatbot answered 14.3% of the questions correctly with a proper value from the ontology. In the other case, the chatbot could not answer 51.4% of the questions because it could not understand the question properly due to indicator matching or entity extraction. In the last case, the chatbot answered 34.3% correctly because it understood the question, but unfortunately, the value in the period was not uploaded for this test.

Table 4.12: Goal Completion Rate: Correct Answer

ID	Number of Questions	Percentage
1	5	14.3%
2	18	51.4%
3	12	34.3%

With the previous results, we decided to group categories 1 and 3 because the chatbot understood the question and answered correctly according to the values stored in our ontology due to the problem nature. The results are shown in Figure 4.11 and state that the chatbot answered correctly 48.6% of the questions from the test evaluation while 51.4% were answered incorrectly.

Bot Response Time

We extracted the chatbot's time to answer the questions and calculated the average from the testing log data. The chatbot took approximately 1.5 seconds to answer a question of scientific indicators, and the actual process takes approximately 10 seconds to retrieve the answer.

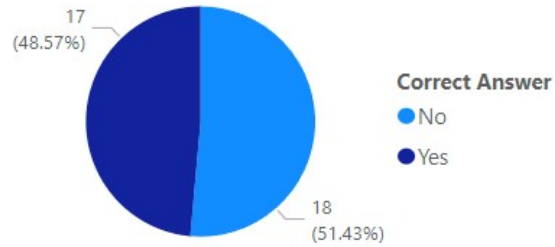


Figure 4.11: Goal Completion Rate: Correct Answer

4.4.2 Survey Evaluation

This evaluation consists of a survey in which users submitted their responses after testing the chatbot. The survey contained questions about metrics such as user confidence in their knowledge about scientometric indicators at Tecnológico de Monterrey, chatbot usability, strictness in language variations, chatbot comprehension, correlated replies from the chatbot, and users' satisfaction over the chatbot. The response was available from 1 to 5, with 1 meaning low and 5 referring to a high level. We had 11 users from the Research Office at Tecnológico de Monterrey who participated in the survey, and the replies are shown in Table 4.13.

Table 4.13: Survey Data

User	User Confidence	Usability	Strictness	Comprehension	Correlation	Satisfaction
1	4	4	1	1	1	1
2	5	3	1	1	1	1
3	4	4	3	3	3	4
4	4	4	1	1	1	1
5	4	1	1	1	1	3
6	5	4	1	1	1	1
7	3	4	3	3	3	3
8	4	4	3	4	4	4
Avg	4.2	3.5	1.75	1.87	1.87	2.25

From the survey data, we can observe that the overall users' knowledge about scientometric indicators at Tecnológico de Monterrey is 4.2, meaning that their confidence in the topic is high and questions were truly made as in a real-world scenario. The usability metric was evaluated with an average of 3.5. This evaluation refers to the chatbot's ease of use in accessibility, conversation, and how long it takes to answer. Strictness in terms of language variations by changing the format of the questions was scored with 1.75. The chatbot's comprehension of the questions about scientometric indicators was rated at 1.87, including how the chatbot understood the question and how it works internally to obtain the value from the

ontology. After the messages interchanged between users and the chatbot, users evaluated the correlation of the chatbot replies with a score of 1.87. This metric considers that the chatbot uses the context of the conversation and does not lose focus on answering the question. Finally, users' feelings towards the chatbot, including their experience and future work, were 2.25 out of 5.

Chapter 5

Discussion

In this chapter, we discuss the work done in this research, including five main processes: Data Preparation and Model construction, Ontology model evaluation, Ontology storage, Indicator querying, and finally, the implementation and evaluation of the chatbot along with the ontology model.

5.1 Data Preparation and Model Construction

Data Preparation represented a challenging part of constructing the ontology model. Since we collected an original dataset from scientometric indicators from Tecnológico de Monterrey, we found that the current format did not allow us to work straight with the data. We took advantage of choosing ten frequently used scientometric indicators to extract the data in a table-based format considering all the relational constraints that existed in the original dataset. This process allowed us to understand better the way we needed to process the data for the RDF file construction. After we understood how the format of the scientometric indicators' datasets should be, we remarked that doing a melting process to the datasets made the data more accessible to understand in terms of dimensions of time and school used in a further process.

Model Construction is one of the essential parts of the process of this research work. We constructed an ontology model with solid bases to be updatable and quick to access, allowing us to have scalable research work. Previous ontologies such as VIVO and BIBO provide semantic definitions for researchers, institutions, schools, and publications, but they do not accurately represent scientometric indicators. Whereas BiDo defines some scientometric indicators, using a vocabulary for statistical data such as SDMX allows to describe each indicator and compose new ones using their description. By extending SDMX with definitions borrowed from the VIVO ontology, we provide means for representing scientometric indicators. In this way, the detailed information used for calculating these indicators could be accessed and verified in the institutional VIVO instance.

If we have to deploy this model to other institutions using the same domain, we must understand how their data is structured. After understanding their data, it is important to prepare it for a proper input of this model and add non-existent dimensions to the model. Extending vocabularies allows us to generalize the concepts in the domain, and existent resources can

be reused. In the case of deploying this model in a different domain, model construction will change and need to focus on accomplishing the necessities required for solving the problem. Tasks such as data understanding and model conceptualization are needed to produce a model of the new domain. It is possible to reuse some ontologies if it converges with the new domain.

The model is prepared to add new scientometric indicators, metrics, or dimensions. In the case of adding new data, the data structure needs to be validated to have the dimensions that the new indicator would need. Updating information is easy to perform. In the scenario where a simple observation value needs to be updated, that observation value contains a URI, a unique identifier. Through this URI, the update can be done without the requirement of other steps.

The ontology modeled the scientometric indicators in English. It represents an issue when querying this ontology using Spanish concepts. One way to tackle this issue is to follow a multilingual ontology approach which links the original ontology with multiple languages in the same system. This approach can apply semantic tagging, which adds labels to the ontology elements in other languages.

Unlike Fox's approach [33], we used the well-known SDMX data model, which would facilitate the adoption of our approach. On the other hand, we extended the work of Thyri et al. [60] by evaluating the representation of multidimensional indicators modeled with SDMX. We took a similar strategy as [37] by converting data from a relational database into RDF. Nevertheless, their approach included the BIBO ontology to capture information about the paper's timeline, while our approach extended SDMX to describe the data and metadata of the scientometric indicators. In our deployment to Neo4j, we only imported the basic definitions for time intervals and VIVO instances (e.g., Schools). However, it is possible to import additional information available in the corresponding linked data platforms. This information includes sequence order between time intervals and the list of schools' faculty. The former can calculate the increment from one year to another in a given indicator (e.g., publications). In contrast, the latter can be used for counting the number of faculty members and building a new indicator.

5.2 Model Evaluation

The model evaluation approach stressed the ontology's capability in terms of self-knowledge, such as retrieving available indicators, measuring discovery, dimensions arrangement, and indicator composition. The second evaluation retrieves the corresponding values depending on the query complexity and calculates new indicators using the existing ones. In contrast to a relational data model, a user only needs to know the basic structure of a SDMX model to discover which metrics are defined and how they are broken down. Cypher is the query language for our ontology model stored in Neo4j, and it is very intuitive and graph-optimized.

The first approach is about ontology self-knowledge. The first evaluation was to retrieve which scientometric indicators were available in the ontology model. Self-knowledge plays a significant role for the chatbot in terms of user interaction. Knowing the exact knowledge domain of the topic will allow the chatbot to set a boundary for answering questions. Furthermore, this process also sets knowledge in the chatbot by having labeling tags for scientometric indicators recognition. The following evaluation was to discover the different measures in the

ontology model. SDMX helped define the measures in our ontology and establish the relation with the scientometric indicators. This query evaluation also is fitted into the chatbot knowledge to let the users know which scientometric indicators are measured. Another evaluation was to identify the dimension arrangement of each scientometric indicator. This evaluation allows us to structure proper knowledge for the chatbot for analyzing missing elements such as dimensions for the scientometric indicator identification process. The last evaluation consists of the composition of a scientometric indicator using a drill-through method from the measure. SDMX allows us to have the proper relationship between nodes in our ontology to break down the information from the indicator using the measure as the pivot.

The second approach consists of performing queries that users might ask the chatbot. In this evaluation, we increased the complexity of the questions in each step to evaluate the chatbot's performance with frequently asked questions about scientometric indicators. We started with a simple question by retrieving a value from a particular year period. This evaluation uses time dimension interval to identify if the indicator is annual or quinquennium time-based. The result returns the indicator's type of measure to allow the chatbot to answer with consistency. The second evaluation is to retrieve a value using a time interval range. The complexity is found in determining how to tackle the time interval problem. SDMX allows us to filter by the time dimension with intervals of type quinquennium, and the result returns a table with all the values along with the time interval. This feature will allow the users to optimize the questions in the chatbot. Instead of doing ten questions for each year, we use the time interval to retrieve the values in one question. The following evaluation consists in adding more complexity by using aggregation functions. This evaluation allows us to produce many options for the users by obtaining average, count, min, max, variance, and standard deviation, among other aggregation functions from the data. The last evaluation demonstrates that we can produce new scientometric indicators using data already stored in our graph database. This new calculation is essential because it opens new possibilities for visualizing and analyzing data.

5.3 Ontology Storage

In this evaluation, we compare the storage of 10 scientometric indicators in our ontology model in Neo4j against the same 10 scientometric indicators uploaded in a RDF graph. The Neo4j graph contains 287 nodes and 856 relationships, while the RDF graph has 1578 triplets. The Neo4j graph database achieves this quantity of nodes and relationships because all the sdmx-dimensions (time and school) are indeed reused nodes with a relationship with all the scientometric indicator (observation) node types. The Neo4j is 45% smaller than the original RDF graph in terms of relationships/triplets. The reduction in the number of relationships in Neo4j is due to the absorption of RDF literal values into nodes.

5.4 Indicator Querying

One of the objectives of the semantic representation of scientometric indicators in our research work is to allow a connection between the ontology model and the chatbot. The semantic representation lets us have an ontology model that describes itself and provides the chatbot with features to inform the user about available indicators and measures. Semantic representation

allows the ontology model to reuse and extend different vocabularies such as time intervals and statistical data in dimensions. The relationship of descriptive and numerical data in the model helps the chatbot retrieve values with queries that do not require much complexity to build them—the chatbot requires this process to identify the scientometric indicator in a natural language question. The semantic representation allows the model to describe the indicators with labeling tags to match them with tags obtained from the question to identify the indicator and reduce complexity in matching. Entities extracted from the question were used to condition the query in the ontology model by the relationship of descriptive data.

5.5 Chatbot Evaluation

The chatbot evaluation consisted of two-week testing with real users from the Research Office at Tecnológico de Monterrey. They used the chatbot and asked questions about scientometric indicators. This strategy leads us to evaluate the chatbot using two different approaches. The first approach is to evaluate the log of the chatbot from the testing period and use the Goal Completion Rate (GCR) of several features. The second approach is to decompose a survey that each user answered at the end of the testing period of the chatbot.

In the first approach, we evaluated the Goal Completion Rate of several features extracted from the log in the testing period of the chatbot. The first feature to evaluate was if the chatbot correctly answered the questions about scientometric indicators. We followed a rule which determines if the chatbot answered correctly or not. If the chatbot returned a value and is correct according to the context of the question, it is considered correct. If the chatbot could not identify the scientometric indicator or did not extract correctly the entities (time and school dimension if it applies), it is considered incorrectly. If the chatbot understood the question and extracted all the components correctly, but the time dimension value does not exist in the domain, the chatbot answered that the information is not available and is considered correctly answered. After analyzing all these rules, we can state that the chatbot answered 48.57% of the questions correctly. This result could not get near 100% because the chatbot only had one training step with questions about the scientometric indicators. This process led the chatbot to generate learning with those kinds of questions. In the testing evaluation, when the users asked questions with different variations in the way of asking by changing verbs or changing the position of nouns and predicates, the chatbot could not understand the questions.

The second evaluation was to identify how many times the chatbot correctly identified the intention of the question. The chatbot was trained with Luis AI service and learned about five intentions: Low, Medium, and High Complexity Questions and the Greeting and None intentions. After performing analysis in the log, the chatbot identified the intention correctly in 71.42% of the questions. This result means that the training helped a lot in terms of variances in the questions due to complexity. Obtaining this result allowed the chatbot to understand what process to follow to answer the question correctly.

The third evaluation consisted in evaluating the entity extraction of the question. Entities such as time, indicator, type of indicator, and school were extracted in each question. We can state that the chatbot correctly identifies these entities in 65.71% of the questions due to differences in asking questions and missing required data.

The last evaluation from the log approach was to evaluate the correct identification of

scientometric indicators in the questions. Our methodology strategy used the set theory with labeling tags to identify the indicator. The chatbot could identify the correct indicator in 48.6% of the questions. From this evaluation approach, we can state that every time the chatbot correctly identified the scientometric indicator, it answered the question correctly. It is essential to state that the chatbot only had one round of training and obtained reasonable goal completion rates.

From the log evaluation, we can state that the chatbot did not obtain the expected results in the interpretation of the question. Understanding the question requires success in a series of processes such as identifying the intention, entity extraction, indicator identification, and data availability. Our research highlights the approach of defining three types of complexities for intention identification by obtaining a good result in the evaluation. However, this only represents a part of the process of interpreting the question correctly. There can be several issues like not extracting the correct entities and not filtering the desired value or not having the value stored in the model. In order to improve the Goal Completion Rate, several strategies can be performed. Adding more valuable natural language processing such as lemmatization or stemming may improve the training datasets. It is also important to add more training steps to the model. Several variations, including synonyms or formatting questions in a different order, can improve the training model. By improving these processes, the chatbot could increase its evaluation in terms of GCR.

In the second approach, a survey is used for evaluation. Overall, all the users who tested the chatbot felt confident in their knowledge of the topic because the questions made in the testing period were from the Research Office at Tecnológico de Monterrey. The first feature to evaluate in the survey was the usability of the chatbot. The users considered an overall of 3.5 because it offered a good interaction and was intuitive to use. The second feature was language strictness which was evaluated with 1.75 overall. This result was expected because the chatbot was trained with only a language format and the questions made during the testing period had different language variations in the questions. This result goes along with the following feature called chatbot comprehension that obtained an overall of 1.87, as in most of the questions existed variations from the training data, the chatbot could not understand the questions. The consequence was that it answered incorrectly or expected more context in the questions. The next feature is the correlation in answers from the chatbot with an overall of 1.87. Due to the mentioned problem, when the chatbot got confused, it asked for information that existed in the question or could not identify intentions or entities correctly. Finally, we evaluated the users' feelings towards the chatbot, and it obtained 2.25.

There are several reasons for the obtention of these results from the survey. We decided only to briefly explain the chatbot and not provide sample questions not to produce bias in the users. Users tested the chatbot without any experience, which provoked several misunderstandings in asking questions. One example that can be considered an opportunity area is that the descriptions of the indicators and measures were in English. The users tried to ask questions in this language when the chatbot was only prepared to receive Spanish questions. We had questions without any stop words or verbs. Another reason was that when users asked a question and data was not available in the model, and they thought the chatbot did not understand the question. Chatbot usability can improve by applying reference guide cards to help them build default questions for the chatbot. This feature includes selecting from a list of available indicators and filtering through time and school dimensions. Yet, this is only to help

the users learn how to ask, but their real questions help the chatbot improve its knowledge by using them as training datasets. The goal is that the chatbot learns from real questions and not impose the user to ask in a certain way. Finally, we also consider that these results would have improved if more training steps were added to the model to increase the chatbot's capacity to understand questions. We found much feedback from this evaluation approach that will allow us to improve the chatbot.

Despite following the same goal of extracting information for a chatbot, Huang [38] used a SVM classifier to obtain data from Forums. In contrast, we used semantic representation by extracting data from a relational database and modeling it in an ontology using SDMX to extend dimensions and measures. The semantic representation allows the model to describe itself, improving the chatbot's knowledge. We extend the research work from Munir [44] in retrieving information from a database to build an ontology and translate ontological knowledge into search requests. With our approach of implementing the ontology in Neo4j, we facilitate querying despite the complexity of the request.

We consider successful implementing the scientometric indicators' ontology model to the chatbot because it could identify the intention in 71.42% of the questions. Almost 50% of the testing questions could be answered correctly by following the methodology of identifying indicators and value retrievals from the ontology model with different levels of complexity.

Chapter 6

Conclusion

The hypothesis stated in this research was proven correct. This research demonstrated that a chatbot could be used along with a statistical ontology model that extends SDMX to correctly answer any given questions about scientometric indicators. We proposed a framework that includes the methodology to model the scientometric indicators and develop the chatbot knowledge. This ontology model is scalable and with quick access due to its construction and definition, including the extension of SDMX properties in dimensions and measures definitions allowing quick access to the value retrieval. Using an intelligent agent like the chatbot with an ontology model provides a strong foundation for developing AI services for research and academic purposes.

The research's general objective of this thesis was to develop a chatbot using an ontology model and indicators modeled with an SDMX extension that understands natural language questions related to scientometric indicators and answers these questions correctly. We can state that the specific objectives have been met, and the research questions were answered during the development of this research work. We cover the proposed framework methodology with the essential processes in the coming paragraphs and answer the established research questions.

The data collection and data understanding steps were successful with the cooperation of the Research Office at Tecnológico de Monterrey. We obtained a dataset that contained more than 100 scientometric indicators, and we chose a sample that contained diversity in the dimensional aspect. This process gave us a solid base for understanding the research question on how to implement the SDMX extension for modeling the indicators. The dataset given had a relational aspect that let us analyze and propose a way to transform and process the data to use it in an ontology model with the SDMX extension.

The data processing step was very challenging because after understanding the dataset, we needed to transform it to reduce the complexity of model construction. We first extracted ten scientometric indicators in different relational tables and then performed a melting process. Each scientometric indicator dataset was reshaped from all the columns of the time dimension, and data were transformed from wide to long. Columns were variables, and they were treated as values after the melting process. This process was necessary to consolidate the characteristics of the dataset. This step allows us to have excellent foundations to achieve an updatable and provide a quick access model in a further process. This process was automated to receive as input all the scientometric indicator datasets and return as outcome all the

transformations needed to be ready for the model construction.

The RDF construction step was semi-automated for modeling the scientometric indicators in the ontology. This approach lets us have descriptive information of the indicators and reuses existing resources such as data cubes, statistical data and metadata, and time intervals. The RDF standard allowed us to merge different schemas and support evolution. This step defined essential concepts, such as Vocabularies and Namespaces, Dataset, Data Structure Definition, Dimensions, Measures, Concept Scheme, and observations. Every concept was influential in the RDF construction because they play an essential role in relationships between them for the model. We focus on the dimension and measure concepts to answer the research question on how the SDMX extension will consist of the indicators modeling. We extended the SDMX property `refPeriod` to represent time intervals in the dimensions' definition. In the measures' definition, we extended the SDMX property attribute representing the unit in which data is measured. These SDMX extensions allow us to describe the data statistically and make querying easier and be human-understandable.

The modeling and evaluation steps allowed us to generate an ontology model whose characteristics allow the model to be updatable and with quick access. The modeling process consisted in uploading each RDF file that contained a scientometric indicator modeled and defined into the graph database in Neo4j. Suppose in the future we want to add another sample of scientometric indicators. It is easy to do because we need to process the data with the automated scripts, construct the RDF file definition, and upload it to the model to increase the number of available scientometric indicators. It is quick access because querying in Neo4j is easier than querying RDF graphs or relational databases. It takes advantage of nodes and relationships and their properties to make the syntax intuitive. The evaluation process allowed us to meet the objective of evaluating the use of SDMX for modeling scientometric indicators. In this step, we used an approach of ontology self-knowledge that evaluates if the ontology can describe itself in terms of scientometric indicator retrieval, dimensions and measures discovery and indicators' composition. SDMX was important in modeling, making the query for the mentioned approach easier by using the node properties. We also evaluated different types of query complexity, and SDMX made querying human-readable even do in high complexity queries. We can state that the semantic representation of scientometric indicators served the purpose of answering questions through a chatbot.

The chatbot development step was also a vital process of this research work. The use of Luis AI service to train data received from a collaborator of the Research Office at Tecnológico de Monterrey and manually classified their intentions and entities allowed us to answer the research question on how to fully understand users' intents related to the topic of scientometric indicators. Our approach defined intents to have a complexity (Low, Medium, and High). Low complexity tackles simple questions that retrieve a value of the model. Medium complexity focuses on aggregation functions of several model values, and High Complexity is about predictions and data forecasting. This classification definition is a relevant contribution to this research work because it allows us to receive a different kind of input and be capable of answering a question without changing the internal processes of the chatbot, producing a scalable solution. The next step was classifying training data. Each question of the training data was classified, and when the chatbot received a question, we used this model to identify the intention of the question. We also trained the model to extract entities. Entities are keywords in questions that help the chatbot answer the question correctly. After identifying the

scientometric indicator asked in the question, we extract the entities with the Luis AI model. If the entities extracted do not match the description of labeling tags in the ontology model, the chatbot asks for the missing values to answer the question correctly. This process allowed us to answer the research question of how the chatbot talks with the user to create a context for gathering data when there is not enough information to answer the question. The chatbot needs to identify the question's intention and extract the entities needed to understand the question and provide a correct response. It allows us to answer the research question on which essential keywords the chatbot must recognize to perform the task.

The chatbot testing step consisted of a three-week testing period with users with high confidence in the knowledge of the scientometric indicator domain that asked several questions to the chatbot. The chatbot answered correctly in almost 50% of the questions made to it and clearly described itself with available scientometric indicators and measures. We can state that we met the objective of applying an ontology model that allowed the chatbot to understand and classify the user's input to extract the correct information of the question and provide a correct answer. The chatbot development knowledge was designed to create context during the conversation where information was needed to understand the user's question. It allows us to meet the objective of proposing and designing a scalable chatbot that creates a context for missing information for answering a question that can be used in other academic areas or industries.

6.1 Future Work

Many different features and developments, also considered opportunity areas, have been left for the future. This research work can be considered the initial step for using the statistical ontology model and the chatbot in the Research Office at Tecnológico de Monterrey. Future work will concern with the following aspects:

- Improve the training in Luis AI service to achieve higher results in Goal Completion Rate in answering the questions correctly, correctly identifying scientometric indicators, and correct entity extraction. The test we performed provided questions with language variances from experts in the domain that are already labeled and classified and can work as training data to improve the chatbot's knowledge.
- The chatbot was evaluated with a query complexity approach in which the result was positive. It will be interesting to add medium and high complexity knowledge to the chatbot in the future. Having a feature of using aggregation functions in time intervals and using stored information to forecast the number of cites in a particular year using machine learning models can lead to a very intelligent chatbot.
- Provide the Scientometric model with higher-granularity information such as knowing the authors of the number of publications indicator or storing the active researchers and postdocs to give another type of analysis.
- Implementing VIVO integration will allow us to perform a different analysis. It can be valuable for the Research Office at Tecnológico de Monterrey because they are currently using this ontology.

- In the chatbot deployment, we can improve the usability by using thumbnail cards during the conversation to guide the user in building the question according to the knowledge, context, and stored data in the ontology model. This feature will help in having fewer questions with missing entities.
- Despite SDMX only supporting the representation of statistical metrics calculated over the entire population, it could be extended to support approximate calculations made on Big Data. Approaches such as Gapprox [17] make use of clustering and sampling techniques for obtaining statistical metrics with 95% of confidence. The remaining 5% could be annotated as an uncertainty attribute in the indicators built with this method.
- Some processes were automated, but still, others are performed manually, such as data extraction from the raw dataset or uploading the RDF files to Neo4j. Complete automation is an important feature that will be made in the future to reduce time costs and make the research work with better quality.

Appendix A

Appendix

The following table shows the complete list of Scientometric Indicators found in the Original Dataset provided by the Research Office at Tecnológico de Monterrey. The selected indicators for our research work are bolted in the table.

Table A.1: Complete list of Scientometric Indicators from Research Office

ID	Indicator
1	Publicaciones Quinquenio
2	Citas Quinquenio
3	Citas por Documento
4	Publicaciones Año Scopus - Tec
5	Publicaciones Anuales por Escuela
6	Publicaciones Quinquenales por Escuela
7	Citas Quinquenales por Escuela
8	Citas por Documento por Escuela
9	Publicaciones promedio por Investigador
10	Coautoría Quinquenal a Nivel Tec (entre escuelas)
11	Coautoría Anual a Nivel Tec (entre escuelas)
12	Coautoría Quinquenal a Nivel Tec (intra escuelas)
13	Coautoría quinquenal a Nivel Escuela (intrerescuela)*
14	Numero de Autores TEC por Publicacion Anual
15	Vitalidad Intelectual
16	Mapeo a Identificador
17	% Mapeo
18	Numero de Autores Distintos
19	% Vitalidad Intelectual del Total de Publicaciones (Algoritmo)
20	Numero de Patentes

21	Patentes Otorgadas
22	Licenciadas
23	Alumnos de educacion superior
24	Alumnos de posgrado a Nivel Nacional
25	Alumnos de posgrado
26	Alumnos haciendo Investigacion
27	Alumnos haciendo Investigacion Doctorado
28	Alumnos haciendo Investigacion Maestría Científica
29	Alumnos haciendo Investigacion Maestría Profesionalizante
30	Alumnos haciendo Investigacion Profesional
31	EIC - Profesional
32	EIC - Posgrado
33	EMCS - Profesional
34	EMCS - Posgrado
35	EHE - Profesional
36	EHE - Posgrado
37	ECSG - Profesional
38	ECSG - Posgrado
39	EN - Profesional
40	EN - Posgrado
41	EAAD - Profesional
42	EAAD - Posgrado
43	EAAD - EIC - Profesional
44	EHE - EAAD - Profesional
45	Otros (PI-AE)
46	Sin Escuela - Profesional
47	Sin Escuela - Posgrado
48	Total Escuelas
49	SNIS (Mexico)
50	SNIS (Datos Enero)
51	% de SNIs Tec con respecto a México
52	SNIs (Actualizado con corte mensual)
53	Num. Investigadores
54	En Modelo
55	Lider
56	Miembro
57	Adscrito-A
58	Adscrito-B
59	Num. De Posdocs
60	Numero de Stars
61	GIEES

62	Por Escuela-Por tipo personal
63	Monto
64	Num Ganadores
65	QS WUR
66	QS LATAM
67	QS GER
68	QS by Faculty - Arts and Humanities
69	QS by Faculty - Social Science and Management
70	QS by Faculty - Engineering and Technology
71	Art and Design
72	Business and Management
73	Modern Languages
74	Computer Science & Information Systems
75	Engineering - Mechanical, Aeronautical & Manufacturing
76	Accounting & Finance
77	Law
78	Engineering - Electrical & Electronic
79	Engineering – Chemical
80	Economics & Econometrics
81	Medicine
82	Agriculture and Forestry
83	Chemistry
84	Physics & Astronomy
85	THE WUR
86	THE LATAM
87	THE Emerging Economies
88	THE Global Employability University Ranking
89	THE Impact Ranking
90	Webometrics Mundial
91	Webometrics LATAM
92	Webometrics Mexico
93	Webometrics Repositorio
94	Artículos de Conferencia
95	Artículos de Revista
96	Casos
97	Libros
98	Libro Traducido
99	Nota Periodística
100	Reportes Técnicos
101	Reseña / Ensayo
102	Resúmenes de Patentes

Bibliography

- [1] Azure bot service. <https://azure.microsoft.com/en-us/services/bot-services/>. Accessed on 27 Mar 2020.
- [2] Conversational agent definition — deepai. <https://deepai.org/machine-learning-glossary-and-terms/conversational-agent>. Accessed on 11 Mar 2022.
- [3] Dialogflow. <https://cloud.google.com/dialogflow/docs>. Accessed on 27 Mar 2020.
- [4] Dialogs library. <https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-concept-dialog?view=azure-bot-service-4.0>. Accessed on 27 Mar 2020.
- [5] Events and triggers in adaptive dialogs - reference guide. <https://docs.microsoft.com/en-us/azure/bot-service/adaptive-dialog/adaptive-dialog-prebuilt-triggers?view=azure-bot-service-4.0>. Accessed on 27 Mar 2020.
- [6] Goal completion rate. <https://growthvirality.com/digital-marketing-kpis/goal-completion-rate/>. Accessed on 07 June 2022.
- [7] Ibm watson: A cheat sheet. <https://www.techrepublic.com/article/ibm-watson-the-smart-persons-guide/>. Accessed on 27 Mar 2020.
- [8] Introduction to bot framework composer. <https://docs.microsoft.com/en-us/composer/introduction?tabs=v2x>. Accessed on 27 Mar 2020.
- [9] Neo4j graph database. <https://neo4j.com/product/graph-database>. Accessed on 20 Jun 2020.
- [10] Owl - semantic web standards. <https://www.w3.org/OWL>. Accessed on 15 Nov 2020.
- [11] Rdf - semantic web standards. <https://www.w3.org/RDF/>. Accessed on 15 Nov 2020.
- [12] A simple introduction to natural language processing. <https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32>. Accessed on 14 Nov 2020.
- [13] What is artificial intelligence (ai)? — ibm. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>. Accessed on 14 Nov 2020.
- [14] What is sdmx? — sdmx statistical data and metadata exchange. https://sdmx.org/?page_id=3425. Accessed on 11 Mar 2022.

- [15] ADAMOPOULOU, E., AND MOUSSIADES, L. An overview of chatbot technology. pp. 373–383.
- [16] AGARWAL, R., DEO, A., AND DAS, S. Intelligent agents in e-learning. *SIGSOFT Softw. Eng. Notes* 29, 2 (Mar. 2004), 1.
- [17] AHMADVAND, H., GOUDARZI, M., AND FOROUTAN, F. Gapprox: using gallup approach for approximation in big data processing. *Journal of Big Data* 6 (2019), 1–24.
- [18] ALBAYRAK, N., ÖZDEMİR, A., AND ZEYDAN, E. An overview of artificial intelligence based chatbots and an example chatbot application. In *2018 26th Signal processing and communications applications conference (SIU)* (2018), IEEE, pp. 1–4.
- [19] BABY, C. J., KHAN, F. A., AND SWATHI, J. Home automation using iot and a chatbot using natural language processing. In *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)* (2017), IEEE, pp. 1–6.
- [20] BIAGETTI, M. T. Ontologies (as knowledge organization systems), 2020.
- [21] CAPADISLI, S., AUER, S., AND RIEDL, R. Towards linked statistical data analysis. In *1st International Workshop on Semantic Statistics (SemStats 2013)* (2013), pp. 61–72.
- [22] CHEN, Z., LU, Y., NIEMINEN, M. P., AND LUCERO, A. Creating a chatbot for and with migrants: Chatbot personality drives co-design activities. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (New York, NY, USA, 2020), DIS '20, Association for Computing Machinery, p. 219–230.
- [23] (CHOWDHURY, G. Natural language processing. *Annual review of information science and technology* 22 (2005), 79–108.
- [24] COLLEDGE, L. *Snowball metrics recipe book*. Elsevier, Montreal, 2017.
- [25] CONLON, M., WOODS, A., TRIGGS, G., O'FLINN, R., JAVED, M., BLAKE, J., GROSS, B., AHMAD, Q. A. I., ALI, S., BARBER, M., ET AL. Vivo: a system for research discovery. *Journal of Open Source Software* 4 (2019), 1182.
- [26] COPELAND, B. J. Artificial intelligence. <https://www.britannica.com/technology/artificial-intelligence>, 08 2020. Accessed on 14 Nov 2020.
- [27] CYGANIAK, R., FIELD, S., GREGORY, A., HALB, W., AND TENNISON, J. Semantic statistics: Bringing together sdmx and scovo.
- [28] ENGELMAN, A., ENKVIST, C., AND PETTERSSON, K. A fair archive based on the cerif model. *Procedia Computer Science* 146 (2019), 190–200.
- [29] ESCOBAR, P., CANDELA, G., TRUJILLO, J., MARCO-SUCH, M., AND PERAL, J. Adding value to linked open data using a multidimensional model approach based on the rdf data cube vocabulary. *Computer Standards & Interfaces* 68 (2020), 103378.
- [30] EUZENAT, J., SHVAIKO, P., ET AL. *Ontology matching*, vol. 18. Springer, 2007.

- [31] FERNANDES, D., AND BERNARDINO, J. Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In *Proceedings of the 7th International Conference on Data Science, Technology and Applications* (2018), p. 373–380.
- [32] FØLSTAD, A., ARAUJO, T., PAPADOPOULOS, S., LAW, E. L.-C., GRANMO, O.-C., LUGER, E., AND BRANDTZAEG, P. B. *Chatbot research and design*. Springer, Amsterdam, 2020.
- [33] FOX, M. S. The semantics of populations: a city indicator perspective. *Journal of Web Semantics* 48 (2018), 48–65.
- [34] GRÉVISSE, C., AND ROTHKUGEL, S. An skos-based vocabulary on the swift programming language. In *International Semantic Web Conference* (2020), Springer, Ed., pp. 244–258.
- [35] GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge acquisition* 5, 2 (1993), 199–220.
- [36] HOOD, W. W., AND WILSON, C. S. The literature of bibliometrics, scientometrics, and informetrics. *Scientometrics* 52, 2 (2001), 291.
- [37] HU, Y., JANOWICZ, K., MCKENZIE, G., SENGUPTA, K., AND HITZLER, P. A linked-data-driven and semantically-enabled journal portal for scientometrics. In *The Semantic Web – ISWC 2013* (2013), pp. 114–129.
- [38] HUANG, J., ZHOU, M., AND YANG, D. Extracting chatbot knowledge from online discussion forums. In *IJCAI* (2007), vol. 7, pp. 423–428.
- [39] JOSHI, A., AND MISHRA, G. Artificial intelligence. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology* (New York, NY, USA, 2010), ICWET '10, Association for Computing Machinery, p. 1023.
- [40] KRISHNAN, V. Research data analysis with power bi.
- [41] LABS, N. Neosemantics (n10s): neo4j rdf & semantics toolkit. <https://neo4j.com/labs/neosemantics/>, 2021. Accessed 20 Jun 2021.
- [42] LEYDESDORFF, L. The evaluation of research and the evolution of science indicators. *Studies in Science of Science* 22, 3 (2004), 225–32.
- [43] LORD, P. Components of an ontology. *Ontogenesis* (2010).
- [44] MUNIR, K., AND SHERAZ ANJUM, M. The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics* 14, 2 (2018), 116 – 126.
- [45] NEGASH, S., AND GRAY, P. Business intelligence. In *Handbook on decision support systems 2*. Springer, 2008, pp. 175–193.

- [46] NEO4J. Cypher query language. <https://neo4j.com/product/#cypher>, 2021. Accessed 20 Jun 2021.
- [47] ONTOLOGY, T. B. Bibliographic ontology specification. <http://bibliontology.com/>, 2021. Accessed 7 Nov 2021.
- [48] OSBORNE, F., PERONI, S., AND MOTTA, E. Clustering citation distributions for semantic categorization and citation prediction. In *Proceedings of the 4th International Conference on Linked Science - Volume 1282* (2014), CEUR-WS.org, p. 24–35.
- [49] PERAS, D. Chatbot evaluation metrics. *Economic and Social Development: Book of Proceedings* (2018), 89–97.
- [50] PERONI, S., AND SHOTTON, D. The spar ontologies. In *The Semantic Web – ISWC 2018* (2018), Springer International Publishing, pp. 119–136.
- [51] RADZIWILL, N. M., AND BENTON, M. C. Evaluating quality of chatbots and intelligent conversational agents. *arXiv preprint arXiv:1704.04579* (2017).
- [52] RAMESH, K., RAVISHANKARAN, S., JOSHI, A., AND CHANDRASEKARAN, K. A survey of design techniques for conversational agents. In *International Conference on Information, Communication and Computing Technology* (2017), Springer, pp. 336–350.
- [53] ROUSSEY, C., PINET, F., KANG, M. A., AND CORCHO, O. An Introduction to Ontologies and Ontology Engineering. In *Ontologies in Urban Development Projects*, vol. 1. Springer London, London, 2011, pp. 9–38.
- [54] SHAPIRO, S. C. *Artificial Intelligence (AI)*. John Wiley and Sons Ltd., GBR, 2003, p. 89–93.
- [55] SHAWAR, B. A., AND ATWELL, E. S. Using corpora in machine-learning chatbot systems. *International journal of corpus linguistics* 10, 4 (2005), 489–516.
- [56] SMITH, B. Ontology. In *The furniture of the world*. Brill Rodopi, 2012, pp. 47–68.
- [57] STOTHERS, J. A., AND NGUYEN, A. Can neo4j replace postgresql in healthcare? *AMIA Summits on Translational Science Proceedings 2020* (2020), 646–653.
- [58] SUN, M., WANG, J., ZHANG, D., ET AL. The progress that natural language processing has made towards human-level ai. *Journal of Artificial Intelligence Practice* 3, 1 (2020), 38–47.
- [59] SURESH, N., AND THANUSKODI, S. Research output on maize (zea mays): A scientometric study. In *Handbook of Research on Digital Content Management and Development in Modern Libraries*. IGI Global, 2020, pp. 225–242.
- [60] THIRY, G., MANOLESCU, I., AND LIBERTI, L. A question answering system for interacting with sdmx databases. In *The 6 Natural Language Interfaces for the Web of Data (NLIWOD) Workshop (in conjunction with ISWC)* (2020), HAL.

- [61] TURING, A. M. Computing machinery and intelligence. In *Parsing the turing test*. Springer, 2009, pp. 23–65.
- [62] VINKLER, P. *The evaluation of research by scientometric indicators*. Elsevier, Abington, 2010.
- [63] VITANOV, N. *Science dynamics and research production. Indicators, indexes, statistical laws and mathematical models*. Springer, Bulgaria, 2016.
- [64] VLACHOU, A., DOULKERIDIS, C., GLENIS, A., SANTIPANTAKIS, G. M., AND VOURO, G. A. Efficient spatio-temporal rdf query processing in large dynamic knowledge bases. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (2019), A. for Computing Machinery, Ed., pp. 439–447.
- [65] W3C. Introduction to skos. <https://www.w3.org/2004/02/skos/intro>, 2021. Accessed 7 Nov 2021.
- [66] WANG, X., HUANG, L., XU, X., ZHANG, Y., AND CHEN, J.-Q. A solution for data inconsistency in data integration. *J. Inf. Sci. Eng.* 27, 2 (2011), 681–695.
- [67] WEIZENBAUM, J. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9, 1 (1966), 36–45.
- [68] WEST, D. M. *The Future of Work: Robots, AI, and Automation*. Brookings Institution Press, 2018.
- [69] ZAINAB, T., AND WANI, Z. A. Advancement and application of scientometric indicators for evaluation of research content. In *Advanced Methodologies and Technologies in Library Science, Information Management, and Scholarly Inquiry*. IGI Global, 2019, pp. 532–542.
- [70] ZAKKA, W. P., ABDUL SHUKOR LIM, N. H., AND CHAU KHUN, M. A scientometric review of geopolymer concrete. *Journal of Cleaner Production* 280 (2021), 124353.
- [71] ZOUAGHI, I., MESMOUDI, A., GALICIA, J., BELLATRECHE, L., AND AGUILI, T. Query optimization for large scale clustered rdf data. In *DOLAP* (2020), pp. 56–65.

Curriculum Vitae

Víctor Iván López Rodríguez was born in Monterrey, México, on July 26, 1997. He earned the B.S. degree in Computer Science from Universidad Autónoma de Nuevo León in June 2019. He was accepted in the M.S. degree in Computer Science at Tecnológico de Monterrey in July 2020.

He did an internship in Germany in his B.S. degree program. He studied one semester in Technische Universität Dresden in 2017. In 2018, He worked as a Software Developer intern at IBM in Stuttgart, Germany.

His current research interest relies on Data Analysis and Machine Learning topics applied to real-world problems.

This document was typed in using L^AT_EX 2_ε^a by Víctor Iván López Rodríguez.

^aThe style file `phdThesisFormat.sty` used to set up this thesis was prepared by the Center of Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus